МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ,
СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Санкт-Петербургский государственный университет телекоммуникаций
им. проф. М.А. Бонч-Бруевича»

На правах рукописи

Помогалова Альбина Владимировна

РАЗРАБОТКА МОДЕЛИ И МЕТОДИКИ ОЦЕНКИ ЭФФЕКТИВНОСТИ АДАПТИВНОГО ВЫБОРА БЛОКЧЕЙН-СИСТЕМ С УЧЕТОМ ХАРАКТЕРИСТИК ТРАФИКА В СЕТЯХ СВЯЗИ

2.2.15. Системы, сети и устройства телекоммуникаций

Диссертация на соискание ученой степени
кандидата технических наук

Научный руководитель
кандидат технических наук , доцент
Елагин Василий Сергеевич

Санкт-Петербург – 2024

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ 4
1 РОЛЬ БЛОКЧЕЙН ТРАФИКА В СЕТЯХ СВЯЗИ И ЕГО АНАЛИЗ 12
1.1 Технология блокчейн 12
1.2 Применимость технологии в мире и Российской Федерации 15
1.3 Структура и архитектура блокчейна 18
1.4 Общий принцип работы блокчейна 20
1.5 Структура сетевого пакета 29
1.6 Выводы 43
ГЛАВА 2 ВЛИЯНИЕ АЛГОРИТМОВ КОНСЕНСУСА НА СЕТИ СВЯЗИ 44
2.1 Понятие алгоритма консенсуса 44
2.2 Семейства алгоритмов консенсуса 45
2.3 Нагрузка на аппаратное обеспечение 49
2.4 Исследование применимости блокчейн-сетей для записи данных в контексте ITS 63
2.5 Сетевые граничные значения переключения алгоритмов консенсуса 72
2.6 Выводы 74
ГЛАВА 3 МОДЕЛЬ МОДУЛЯ ПРИНЯТИЯ РЕШЕНИЯ ПО ВЫБОРУ БЛОКЧЕЙН-СИСТЕМ 75
3.1 Принцип работы модуля принятия решения 75
3.2 Аналитическая модель сети связи с блокчейном 81
3.3 Имитационная модель сети с модулем принятия решения адаптивного выбора алгоритма консенсуса блокчейн-сети 84
3.4 Результаты имитационного моделирования 90
3.5 Выводы 93
ГЛАВА 4 МОДЕЛЬ И МЕТОДИКА ОЦЕНКИ ЭФФЕКТИВНОСТИ АДАПТИВНОГО ВЫБОРА БЛОКЧЕЙН-СИСТЕМ 94
4.1 Оценка времени синхронизации узлов сети на примере ITS 94
4.2 Расчет временных характеристик распространения информации по сети связи 99

4.3 Модель оценки эффективности разработанного модуля адаптивного выбора консенсуса 101
4.4 Апробация математической модели оценки эффективности модуля принятия решения 105
4.5 Методика интеграции модуля принятия решения 107
4.6 Методика оценки эффективности адаптивного выбора блокчейн-систем с учетом характеристик трафика в сетях связи 110
4.7 Выводы 112
ЗАКЛЮЧЕНИЕ 113
СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ 115
СПИСОК ЛИТЕРАТУРЫ 117
ПРИЛОЖЕНИЕ А 130
ПРИЛОЖЕНИЕ Б 139

ВВЕДЕНИЕ

Актуальность темы исследования. В условиях непрерывного развития информационных технологий, способствующих созданию сложных вычислительных систем, решающих разнородные задачи, ключевую роль для обеспечения отказоустойчивости и стабильности функционирования играют сети связи передачи данных для обмена информацией. Многие компоненты современных информационных систем генерируют собственный поток информации, требующий обработки и распространения среди других устройств по сети. Выделение приоритетного трафика, балансировка и учет промежутков пиковой нагрузки диктуют определенные требования к используемым технологиям на основе затрачиваемых вычислительных ресурсов, памяти, объема и требований к скорости обработки и передачи генерируемого ими сетевого трафика. Наряду с этим большое внимание уделяется обеспечению безопасности и прозрачности ряда действий в информационных системах, с целью контроля инцидентов, потенциальных угроз злоумышленников, а также гарантий сохранности данных в неизменяемом виде (или фиксирования даты и объема этих изменений).

Технология блокчейн позволяет удовлетворить многие из требований к безопасным и устойчивым системам работы с данными, однако, в подавляющем ряде случаев требует излишних вычислительных ресурсов, обеспечение которых является критическим при использовании вариаций с наиболее популярными алгоритмами консенсуса. Однако существуют более лояльные с вычислительной точки зрения консенсусы, использование которых на постоянной основе также вносит ряд ограничений: требования к созданию доверенных участков сети, задержки на участках связи или обеспечение полносвязной системы, где каждый узел сети напрямую связан со всеми другими узлами. Это провоцирует создание множества алгоритмов консенсуса со своими особенностями для решения специфических задач, что обусловлено различными целями и сценариями использования [1]. Например, финансовые приложения могут требовать высокой

скорости транзакций, тогда как системы для хранения данных могут акцентировать внимание на безопасности и надежности. Это требует разработки специализированных алгоритмов консенсуса, адаптированных к конкретным потребностям [2].

С учетом особенностей современных гетерогенных сетей связи отсутствие гибкости при передаче блокчейн-трафика является одной из ключевых причин отказа от ее внедрения. Так, система регуляции объемов и скорости трафика блокчейн в течение дня позволила бы обеспечить требуемую гибкость для всех случаев – чаще, когда трафик блокчейн является не приоритетным, реже, когда является приоритетным, что зависит от конечных целей и реализации.

Помимо регуляции объемов трафика необходимо отметить и требования к вычислительным ресурсам. Современные сети связи состоят из компонентов разной вычислительной мощности, приоритетной задачей которых является обработка данных и их передача. В текущих условиях установка клиента блокчейн на компоненты сети связи будет оказывать ощутимое влияние на оборудование и скорость обработки данных, особенно в часы пиковой нагрузки. Так как алгоритмы консенсуса во многом определяют, насколько ресурсоемкими будут вычисления для обработки транзакций и формирования блоков, их регуляция в процессе обработки данных с учетом определенных сетевых и вычислительных параметров позволит гибко принимать решение о возможности использования наиболее ресурсоемкого алгоритма и переключаться на менее ресурсоемкий в моменты пиковых нагрузок.

Степень разработанности темы. Область применения и интеграции технологии блокчейн, как частного случая реализации технологии распределенного реестра, ее влияние на сетевых характеристики систем, вопросы применимости рассматриваются в работах отечественных и зарубежных ученых К.Е. Самуйлова, Р.В. Киричка, Б.С. Гольдштейна, А.Е. Кучерявого, А.Г. Владыко, Е.А. Кучерявого, В.Л. Достова, В.С. Елагина, В.В. Корхова, V. Buterin, S. Kasahara, Q. Xia, Y. Sun, L. Coccо, и др.

Многие работы посвящены исследованию вопросов распространения трафика по сети, влияние на сетевые характеристики [3], рассмотрению вопросов технической зрелости подхода для интеграции в существующие системы [4], а также аспекты безопасности технологии. Представлен ряд научных работ, посвященных оптимизации на основе исследований алгоритмов консенсуса [5], однако, проблема адаптации этих алгоритмов к условиям телекоммуникационных сетей остается недостаточно изученной, в особенности с точки зрения представления адаптивного алгоритма выбора консенсуса блокчейна на сетях связи [6]. Ключевыми результатами текущих научных исследований в этой области включают улучшение алгоритмов консенсуса и разработку энергоэффективных методов обработки транзакций, но требуется дальнейшее исследование для создания комплексных решений, учитывающих особенности сетевой инфраструктуры, а также специфики телекоммуникационных сетей.

Приведенные выше авторы и исследования внесли весомый вклад в отношении вопросов телекоммуникационных сетей и влияния на них технологии блокчейн. Представленная диссертационная работа дополняет приведенные выше исследования авторов, раскрывая ранее не рассматриваемые с достаточной степенью детализации вопросы интеграции адаптивного алгоритма выбора консенсуса блокчейн на сетях связи, позволяющего корректно учитывать сетевые особенности разных участков сетей, включая аппаратный уровень, для корректной интеграции технологии блокчейн [7]. При этом решается **научная задача** по разработке модели и методики оценки эффективности адаптивного выбора блокчейн-систем для снижения потерь блоков транзакций на сетях связи и обеспечения достаточного уровня гибкости управления.

Работа фокусируется на концепции мультиконсенсусности при интеграции технологии блокчейн, как основе адаптивности при выборе алгоритма консенсуса, влиянии концепции на сетевые характеристики, разработке модели и методики интеграции модуля принятия решения адаптивного выбора блокчейн-систем. Имитационное моделирование, проведенное в работе, позволяет получить более

точные результаты при проектировании систем и дальнейшей интеграции технологии [8].

Полученные результаты могут служить основой при формировании систем динамического подбора алгоритма консенсуса участка сети в зависимости от приоритезации трафика в некоторый момент времени, а также других характеристик сетевого участка, внося значительный вклад в область исследования вопросов интеграции блокчейн в современные системы связи.

Объектом исследования являются телекоммуникационные сети и системы связи, использующие технологию блокчейн. **Предметом исследования** являются временные характеристики телекоммуникационных систем и сетей связи в условиях применения на них блокчейн-систем.

Цель диссертационного исследования заключается в снижении коэффициента потери блоков транзакций на сетях связи за счет применения нового алгоритма адаптации консенсуса и частоты генерации блоков транзакций к сетевым характеристикам на сетях связи.

Для достижения этой цели поставлены следующие **задачи**:

– анализ структуры сетевых пакетов блокчейн-трафика и оценка его влияния на сети связи,

– анализ влияния блокчейн-систем на аппаратные компоненты оборудования,

– анализ существующих блокчейн алгоритмов консенсуса, включая их преимущества и недостатки, с целью выявления ограничений при применении в телекоммуникационных сетях,

– выявление граничных значений сетевых и аппаратных характеристик наибольшей эффективности рассматриваемых алгоритмов консенсуса,

– разработка модели модуля принятия решения выбора блокчейн-систем на сетях связи, учитывая специфику телекоммуникационных сетей, включая их топологию, пропускную способность, задержки и энергопотребление, расположение модуля в их архитектуре,

– разработка модели оценки эффективности модуля принятия решения по адаптивному выбору блокчейн-консенсуса с учетом сетевых характеристик,

– проведение аналитических расчетов и имитационного моделирования оценки процента расхождения,

– разработка адаптивного алгоритма выбора консенсуса блокчейн-сети с учетом значений сетевых характеристик, как ключевого компонента модуля принятия решения,

– анализ и сравнение результатов аналитических расчетов, имитационного и экспериментального моделирования для тестирования разработанного модуля принятия решения в реальных условиях. Оценка его производительности, надежности и безопасности в различных сценариях использования,

– разработка методики интеграции разработанного модуля принятия решения в различные телекоммуникационные системы и сети. Описание возможных сценариев применения и их преимуществ для операторов и разработчиков телекоммуникационного оборудования,

– разработка методики оценки эффективности разработанного модуля принятия решения с учетом параметров и конфигурации сети связи.

Научная новизна

1) Впервые предложены изменяемые компоненты технологии блокчейн и представлены результаты влияния на сеть связи изменения предлагаемых параметров. Выделены и определены пороговые значения сетевых характеристик наибольшей эффективности алгоритмов консенсуса.

2) Впервые предложен подход адаптивного выбора алгоритма консенсуса технологии блокчейн, в отличие от применяемого в современных исследованиях концепта разработки универсального алгоритма консенсуса, как основа модуля принятия решения. Предложенное математическое описание системы массового обслуживания с модулем принятия решения и смены консенсуса является впервые представленным математическим описанием концепции изменения правил блокчейна в зависимости от сетевых требований в момент времени.

3) Проведена апробация адаптивного выбора алгоритма консенсуса технологии блокчейн в рамках имитационного моделирования и исследование параметров потерь блоков данных при его использовании в отличие от неизменяемого алгоритма консенсуса с изменяемым параметром сложности системы.

Теоретическая значимость исследования заключается в анализе влияния сетевого трафика блокчейн-систем на современные сети связи и модель его распространения. Ценность представлена результатами исследования адаптации участка сети связи под условия более эффективного алгоритма консенсуса. Разработанная модель модуля принятия решения по выбору блокчейн-систем позволяет качественно управлять блокчейн-трафиком и оперировать сетевыми характеристиками для достижения лучшей эффективности сети связи. Предложенный модуль предлагает подходы по повышению уровня целостности сетевого трафика в сетях связи при изменении уровня нагрузки на канал связи. Также сформирована модель массового обслуживания данной системы для аналитического представления вероятностных исходов состояний сети под влиянием адаптивного выбора алгоритма консенсуса блокчейн-сети.

Практическая значимость диссертационной работы состоит в разработке методик оценки эффективности и интеграции модуля принятия решения адаптивного выбора блокчейн-систем для управления информационными потоками блокчейн на сетях связи, возможности применения разработанных моделей в телекоммуникационных системах. Это позволит эффективнее регулировать информационные потоки блокчейн при передаче данных, оптимизировать использование сетевых ресурсов и снизить энергопотребление. Рекомендации по оценке эффективности и интеграции модуля принятия решения могут быть полезны для разработчиков и операторов телекоммуникационных сетей, а также для специалистов, занимающихся интеграцией блокчейн-технологий в различные системы.

Методология и методы исследования. В диссертационном исследовании использовались методы теории телетрафика и теории массового обслуживания,

теории вероятностей, математической статистики, а также методах аналитического и имитационного моделирования событийных систем. Имитационное моделирование разработанного адаптивного алгоритма выполнено с использованием программного обеспечения Python, Kotlin, Anylogic.

Положения, выносимые на защиту:

1) Число сетевых характеристик для эффективного выбора блокчейн-систем в сетях связи не превышает десяти штук с учетом реальных диапазонов их пороговых значений.

2) Модель модуля принятия решения по выбору блокчейн-систем для снижения числа потерянных блоков транзакций показывает точность расчета с погрешностью не более 5% в сравнении с результатами имитационного моделирования.

3) Методика интеграции модуля принятия решения для адаптивного выбора блокчейн-систем с учетом сетевых характеристик позволяет уменьшить потери блоков не менее 10% относительно существующих блокчейн-систем.

Степень достоверности. Достоверность основных результатов диссертации подтверждается корректным применением математического аппарата, результатами аналитического и имитационного моделирования, обсуждением основных полученных результатов в рамках выступлений как на российских и международных конференциях.

Апробация результатов диссертационного исследования. Положения, выносимые на защиту, были представлены и обсуждались на научных конференциях и семинарах: 24-й международной конференции «Distributed Computer and Communication Networks: Control, Computation, Communications» (Москва, 20 - 24 сентября 2021), международной научной конференции «2022 Systems of Signals Generating and Processing in the Field of on Board Communications» (Москва, 15 - 17 марта 2022), 4-й международной научно-технической конференции «Современные сетевые технологии (MoNeTec-2022)» (Москва, 27 - 29 октября 2022), международной научной конференции «2022 Intelligent Technologies and Electronic Devices in Vehicle and Road Transport Complex» (Москва,

10 - 11 ноября 2022), 2-я международная конференция «International Conference on Advanced Computing & Next-Generation Communication» (Санкт-Петербург, 12 - 13 октября 2023), международной конференции «2024 Systems of Signal Synchronization, Generating and Processing in Telecommunications» (Выборг, 1 - 3 июля, 2024), международной научной конференции «Актуальные проблемы инфотелекоммуникаций в науке и образовании» (АПИНО). (Санкт-Петербург 2021 - 2024), семинарах кафедры инфокоммуникационных систем СПбГУТ.

Публикации по теме диссертации. Материалы работы изложены в полной мере в 17 публикациях, из них: 4 статьи в изданиях, включенных в перечень рецензируемых научных изданиях (перечень ВАК при Минобрнауки России), 4 статьи в изданиях, включенных в международные базы цитирования; 3 результата интеллектуальной деятельности; 2 отчёта о НИР; 4 статьи в других изданиях и материалах конференций.

Личный вклад автора. Все основные результаты диссертационной работы получены автором самостоятельно. Экспериментальные исследования проведены при его непосредственном участии и под научным руководством.

Соответствие паспорту специальности. Диссертационная работа выполнена по специальности 2.2.15 – Системы, сети и устройства телекоммуникаций и соответствует следующим пунктам паспорта специальности: 1, 4, 13.

1 РОЛЬ БЛОКЧЕЙН ТРАФИКА В СЕТЯХ СВЯЗИ И ЕГО АНАЛИЗ

Современные сети связи сталкиваются с рядом вызовов, связанных с увеличением объема передаваемых данных, необходимостью обеспечения безопасности и надежности передачи информации, а также требованием к эффективному управлению трафиком. В этом контексте технология блокчейн приобретает все большее значение, как частный случай семейства технологии распределенного реестра.

Технология распределенного реестра является общим термином, описывающим системы, которые разделяют, записывают и синхронизируют данные через несколько узлов. Основная идея ТРР заключается в том, что участники сети имеют копии одинаковых данных, что обеспечивает безопасность и прозрачность их хранения. Блокчейн представляет из себя один из видов технологии распределенного реестра, который представляет собой цепочку блоков данных, где каждый блок содержит информацию, а также фиксирует хэш предыдущего блока и временную метку. Эти блоки связаны друг с другом и формируют непрерывную цепь, что делает данные неизменными и подтвержденными.

1.1 Технология блокчейн

В общем виде блокчейн представляет из себя базу данных, которая одинаковой копией хранится на множестве устройств в одноранговой пиринговой сети. Устройства должны постоянно синхронизироваться между друг другом, чтобы передавать информацию о любых изменениях, валидировать и добавлять новые данные. При этом способ организации блокчейн-сети допускает отсутствие связи типа «каждый с каждым», что позволяет масштабировать сеть [9].

Первым характерным отличием блокчейна от других баз данных и решений сходного направления можно выделить отсутствие возможности удаления данных

или их изменения, единственное доступное изменение – добавление новой информации.

Блокчейн-сеть является полностью децентрализованной, и все участники системы в большинстве случаев обладают одинаковыми правами и возможностями, за исключением частных реализаций блокчейн-сетей и их интеграции в корпоративном секторе, где архитектура и некоторые функции могут отличаться в зависимости от запроса компании (приватные блокчейн-сети).

Ключевой особенностью такой системы является отсутствие единого управляющего центра, что обеспечивает высокую устойчивость к внешним воздействиям и возможность функционирования даже в условиях отказа отдельных узлов. Технология блокчейн была разработана для решения задач, связанных с необходимостью обеспечения доверия между участниками, которые не имеют прямого взаимодействия или не располагают доверенными посредниками.

Ключевыми характеристиками блокчейна являются:

– децентрализация: управление и хранение данных распределено между всеми узлами сети, что исключает необходимость в едином управляющем сервере,

– неподдельность данных: записи в блокчейне защищены от несанкционированных изменений благодаря криптографическим методам и множеству копий информации, где необходимо было бы воспроизвести аналогичные изменения,

– прозрачность: все участники сети имеют доступ к данным, что делает все процессы и данные открытыми для просмотра,

– устойчивость к ошибкам: избыточность копий данных позволяет системе работать корректно даже в условиях выхода из строя части узлов.

Начало развития технология получила в 2008 году, когда была представлена децентрализованная финансовая площадка Биткоин. Первая реализация представила в себе все ключевые принципы, давая основу для дальнейшей трансформации и увеличении поддерживаемых функций.

Основное отличие блокчейна Биткоин от технологий, существовавших в 2008 году, заключается в его децентрализованной природе. До появления Биткоин

существовали централизованные системы, такие как банки и платежные системы, которые контролировали финансовые транзакции. Биткоин, основанный на технологии блокчейн, позволил устранить необходимость в центральном посреднике, что сделало систему более устойчивой к манипуляциям и мошенничеству.

Однако полноценным прорывом технологии следует считать блокчейн-сеть Эфириум, как платформу, которая позволяет разработчикам создавать децентрализованные приложения (dApps) и смарт-контракты. Это расширяет возможности блокчейна за пределы простых финансовых транзакций, предоставляя инструменты для автоматизации и реализации сложных логических операций, а также изоляции программной логики в децентрализованной среде. Программные алгоритмы, написанные и установленные в блокчейн-сети (смарт-контракты) становятся неуязвимыми с точки зрения остановки их исполнения, так как программный код будет исполняться и доступен до тех пор, пока в рабочем состоянии находится хотя бы одно устройство сети.

В отличие от блокчейна Биткоин [10], который в первую очередь предназначен для передачи исключительно информации о переводах между участниками сети, Эфириум фокусируется на функциональности и гибкости [11], предлагая платформу для создания различных приложений, что делает его более универсальным инструментом для разработчиков.

Так, в 2014 году были представлены следующие принципиально новые функции:

– смарт-контракты или программируемые контракты, которые автоматически исполняют условия соглашения при выполнении определенных условий,

– децентрализованные приложения, обеспечивающие возможность создания и развертывания разноплановых интерфейсов к смарт-контрактам, которые работают на децентрализованной сети без централизованного контроля,

– токены и стандарты, что позволяет легко разрабатывать и управлять цифровыми активами,

– возможности интеграции и взаимодействие с другими блокчейнами для создания кроссплатформенных решений.

Подобное расширение функциональных возможностей, а также открытый исходный код значительно расширили интерес к данной технологии наряду с вариантами ее применения. Так разработчикам стали доступны разные конфигурации – от небольших смарт-контрактов и интерфейса для них, где разработчику не требуется решать вопросы сетевого и инфраструктурного характера до собственных доработок и альтернативных версий кода блокчейн-сети, их развертывания в общедоступной среде или в контексте закрытого контура корпоративных решений.

Как итог, эволюция блокчейн-решений предлагает совершенно разные по своим возможностям и назначению решения: кросс-чейн протоколы, блокчейн-подсети внутри единой блокчейн-сети (аналогично сценарию использования множества локальных сетей и масок подсети), множество консенсусов, разные по скорости обработки информации сети, а также решения без хранения данных как таковых. Во всех решениях, за исключением последнего, просматривается несколько ключевых требований: вычислительные ресурсы, стабильные объемы сетевого трафика.

1.2 Применимость технологии в мире и Российской Федерации

К настоящему моменту общее количество представленных блокчейн-сетей насчитывает более тысячи единиц, появившихся с течением времени с 2008-2009 года по текущий момент времени. Многие из этих сетей малоактивны и обладают небольшим количеством поддерживающих их пользователей. Из этого количества можно выделить порядка 60 блокчейнов, которые являются крупнейшими на международном рынке и обладают наибольшим количеством пользователей [12].

На Рисунке 1 представлен график появления крупнейших блокчейн-сетей, которые являются лидирующими по объемам пользователей и генерируемых блоков и транзакций.

Рисунок 1 – Количество крупнейших блокчейн-сетей, появившихся в период с 2009 по 2024 год

По прогнозам аналитика Bloomberg Джейми Куттса [13], в течение ближайших пяти лет количество активных пользователей блокчейн-сетей превысит 100 миллионов. В третьем квартале 2023 года число пользователей блокчейн-сервисов, ежедневно участвующих в транзакциях, оказалось выше 5 миллионов – это на 14% больше, чем в 2022 году. Это число характеризует именно пользователей блокчейн-сетей. Помимо самих пользователей в сети представлены узлы, который отвечают за создание блоков, валидацию транзакций и постоянно участвуют в процессе синхронизации с другими узлами сети. Например, в сети Эфириум количество активных узлов-валидаторов превышает 1 миллион устройств [14].

Безусловно, определенная доля пользователей мотивирована финансовыми операциями и торговлей валютами. Но следует отметить и мировую практику внедрения технологии в многие отрасли. Так, в Голландии и Эстонии активно развивается использование блокчейна в медицинской сфере, в Швеции и Грузии развиваются инициативы по разработке приложений для ведения земельного учета и кадастровой регистрации [15]. Мировая практика интеграции технологии

достаточно обширная и включает в себя множество отраслей, включая логистику, финансовый сектор, здравоохранение, учет и др.

В России технология блокчейн представлена и на государственном уровне. Ярким примером может служить цифровой рубль [16], финансовые цифровые активы, отечественная платформа для обмена и хранения финансовой информации. Также разработан и издан Федеральный закон «О цифровых финансовых активах, цифровой валюте и о внесении изменений в отдельные законодательные акты Российской Федерации» от 31.07.2020 N 259-ФЗ.

Сотовые операторы активно занимаются внедрением технологии, рассматривая сценарии регистрации сим-карт в блокчейне [17].

У ПАО СберБанк открыто подразделение «Лаборатория блокчейн», посвященное изучению и разработке финансовых решений с применением технологии [18].

Специалисты Центра компетенций НТИ «Центр технологий распределенных реестров СПбГУ» разработали платформу «Цифровой метр», которая открывает возможность вместо покупки квартиры за рубли приобретать цифровые квадратные метры за токены, которые и будут приносить доход инвестору. Об этом Санкт-Петербургский государственный университет (СПбГУ) сообщил 7 мая 2024 года [19].

В конце сентября 2024 года Минфин РФ представил проект цифровых обществ с ограниченной ответственностью (ООО). Концепция предусматривает, что исполнение корпоративных договоренностей будет контролироваться с помощью инфраструктуры децентрализованных реестров — блокчейна. [20].

В зависимости от решаемой задачи разработчики решений могут использовать существующие крупнейшие блокчейн-сети или запускать собственные, специализирующиеся на узких сценариях со специфическими условиями или более централизованным уровнем контроля. Ключевым определяющим фактором растущей перспективы использования технологии является ее корректная интеграция в существующие решения и обеспечение

взаимодействия с другими модулями систем для достижения максимальной эффективности.

1.3 Структура и архитектура блокчейна

Блокчейн — это сложная система, состоящая из нескольких ключевых компонентов, каждый из которых играет важную роль в обеспечении безопасности, надежности и функциональности сети. Компоненты включают в себя:

– узлы, представляющие устройства, которые участвуют в блокчейн-сети. Они могут выполнять различные функции в зависимости от мощности оборудования и особенностей сети. Например:

а) полные узлы, которые сохраняют всю историю блокчейна и проверяют транзакции и блоки на соответствие правилам сети. Функции полных узлов включают в себя: подтверждение и валидацию транзакций, хранение полной копии блокчейна, участие в процессе консенсуса,

б) легкие узлы, которые сохраняют только заголовки блоков и используют полные узлы для проверки транзакций. Это делает их менее ресурсоемкими. Функции легких узлов включают в себя: проверку транзакций с помощью полных узлов, быстрый доступ к информации о состоянии сети, а также используются для мобильных и менее мощных устройств,

в) майнинг-узлы, которые участвуют в процессе создания новых блоков и подтверждения транзакций через алгоритмы, такие как PoW или PoS. Функции майнинга-узлов включают в себя создание блоков, включая решение математической задачи или другие аналогичные вычисления, а также подтверждение транзакций,

г) узлы-валидаторы, которые подтверждают транзакции и блоки в системах с алгоритмами консенсуса, основанными на долях (например, в PoS). Функции узла включают проверку и утверждение новых блоков, а также проверку транзакций,

д) делегированные узлы, которые выбираются другими узлами для выполнения функций в сети, часто используемые в системах с алгоритмом консенсуса DPoS. Функции таких узлов включают в себя создание блоков, подтверждение транзакций от лица других участников сети, а также представление интересов делегирующих узлов,

е) архивные узлы, которые хранят всю историю блокчейна и могут предоставлять доступ к данным, не содержащимся в текущих блоках. Их функции включают в себя хранение всех состояний и транзакций для анализа и аудита, а также предоставление API для разработчиков и аналитиков.

– блоки, которые представляют из себя формат транспортировки данных в блокчейне. Каждый блок содержит:

а) заголовок блока, который включает хэш предыдущего блока, временную метку, nonce (число, используемое в майнинге) и корень Меркла,

б) тело блока, которое содержит список транзакций, которые были подтверждены в этом блоке.

– транзакции, которые являются минимальной единицей информации в блокчейне. Транзакции представляют собой записи о передаче цифровых активов или данных между пользователями. Каждая транзакция включает в себя:

а) адреса отправителя и получателя, то есть уникальные идентификаторы пользователей в сети,

б) размер количества активов, передаваемых в транзакции или данные о вызываемых методах и передаваемых аргументах,

в) криптографическая подпись отправителя, обеспечивающая аутентичность и целостность транзакции.

– хэш-функции, являющиеся математическими алгоритмами, которые преобразуют входные данные в фиксированной длины строку (хэш). В блокчейне используются для:

а) создания уникальных идентификаторов блоков, так как хэш блока зависит от содержимого, что делает его уникальным,

б) обеспечения целостности данных, так как изменение любого элемента в блоке приводит к изменению его хэша, что сразу же выявляет подделку или вмешательство в работу системы.

– алгоритмы консенсуса, которые необходимы для достижения согласия между узлами о состоянии блокчейна. Наиболее распространенные алгоритмы среди существующих сегодня систем,

– дерево Меркла, представляющее структуру данных, которая используется для организации и проверки целостности транзакций в блоке, чтобы эффективно хранить и проверять большое количество транзакций,

– методы криптографии, так как они отвечают за обеспечение безопасности блокчейна:

а) публичные и приватные ключи, которые используются для создания адресов и подписания транзакций. Приватный ключ должен оставаться секретным, в то время как публичный ключ доступен другим пользователям для верификации, что сообщение было сформировано именно этим пользователем,

б) эллиптическая криптография, которая часто используется для создания ключей и подписей, обеспечивая высокий уровень безопасности при меньшем размере ключа.

– смарт-контракты — это самовыполняющиеся программные скрипты, который позволяют автоматически исполнять условия соглашений без участия третьих лиц без возможности их остановки и удаления.

1.4 Общий принцип работы блокчейна

Базовый принцип работы блокчейн-сетей состоит из нескольких ключевых этапов. Первый этап - формирование блока. Данные транзакций, которые поступают в систему, группируются в блоки. Каждый блок содержит фиксированное количество записей и хэш предыдущего блока. Затем идет этап добавления блока в цепочку. Перед включением блока в цепочку узлы сети

проводят проверку его содержимого с использованием алгоритма консенсуса. Затем происходит обновление копий. После добавления нового блока все узлы сети получают обновленную версию реестра [21].

Более подробно процесс представлен на серии рисунков (Рисунки 1-5), на которых представлены блок-схемы базового сценария и порядка работы блокчейн-сетей.

Этап 1 (Рисунок 2) - процесс формирования и валидации транзакции. Формирование транзакции включает в себя сбор информации об отправителе, получателе и сумме. После сборки транзакции, хэширования данных и подписания транзакции приватных ключом происходит валидация транзакции на соответствие требованиям сети. Валидация транзакции включает в себя проверку подписи: Узел проверяет, что транзакция подписана отправителем с использованием его приватного ключа. Это гарантирует, что только владелец ресурсов может ими распоряжаться.

Проверка наличия ресурсов. Узел проверяет, что отправитель имеет достаточное количество ресурсов для выполнения транзакции, используя информацию о предыдущих подтвержденных действиях (например, истории транзакций). Проверка корректности. Узел проверяет, что транзакция соответствует требованиям сети (например, правильный формат данных, допустимые адреса). Проверка допустимости. Узел проверяет, не нарушает ли транзакция каких-либо ограничений, таких как размер или специфические правила сети. Если транзакция проходит все проверки, она считается валидной и может быть добавлена в memory pool или пул незавершенных (неподтвержденных) транзакций.

Этап 2 (Рисунок 3). Процесс отбора транзакций в блок. Когда узел собирает транзакции для включения в блок, чаще всего отбираются те, которые принесут наибольшую прибыль. Этот механизм присущ общедоступным сетям, однако, механизм может отсутствовать в приватных сетях, специализированных для каких-либо конкретных задач без внешнего общего доступа.

Процесс отбора включает в себя сортировку транзакций по стоимости. Узел сортирует транзакции по размеру вознаграждения (например, комиссии), которое получит за их включение в блок. Затем может произойти ограничение по размеру блока. Узел может ограничивать количество транзакций, которые могут быть включены в блок, в зависимости от максимального размера блока (или частных настроек узла). В некоторых случаях могут быть другие параметры для выбора транзакций, например, срочность или важность, которые также могут влиять на выбор транзакций.

Этап 3 (Рисунок 4). Дерево Меркла — это структура данных, которая используется для эффективной и надежной проверки целостности данных в блоке. Сначала производится хэширование данных: каждая транзакция в блоке преобразуется в уникальный хэш (например, с помощью хэш-функции, такой как SHA-256). Затем объединение хэшей пар. Для каждой пары хэшей вычисляется новый хэш, который является результатом объединения этих двух значений.

Например, хэш пары H(T1, T2) — это новый хэш, который является результатом объединения хэшей двух транзакций T1 и T2. Этот процесс продолжается для каждой пары хэшей на следующем уровне, пока не останется только один хэш — корневой.

Этап 4 (Рисунок 5). Корневой хэш дерева Меркла представляет собой цифровой отпечаток, который объединяет все транзакции в блоке, обеспечивая эффективную проверку целостности данных. Этот хэш служит криптографическим доказательством, что данные блока не были изменены. Значение корневого хэша фиксируется в заголовке блока. Используя только корневой хэш, можно проверить, были ли изменены отдельные транзакции, без необходимости загрузки всех данных блока.

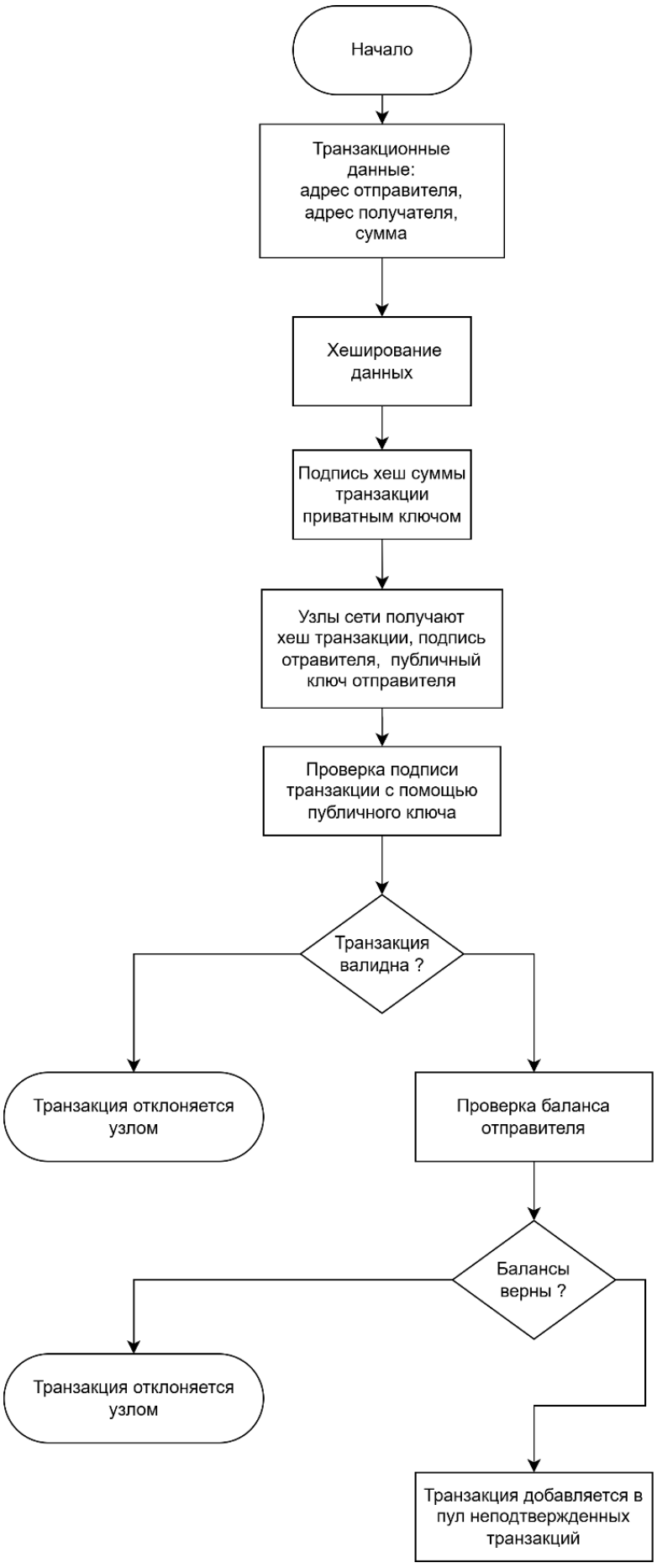


Рисунок 2 – Этап 1. Процесс формирования и валидации транзакции

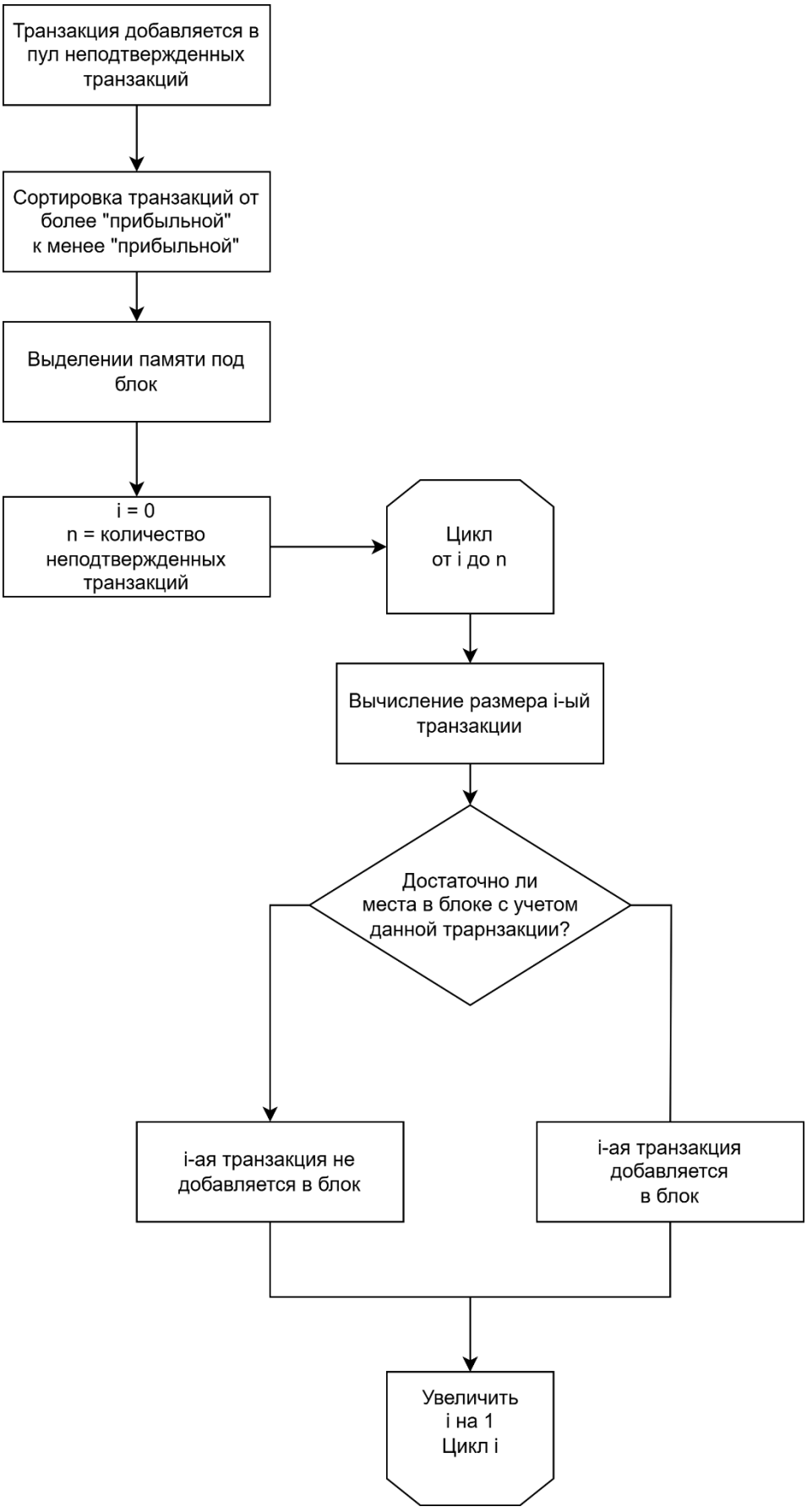


Рисунок 3 – Этап 2. Процесс отбора транзакций в блок

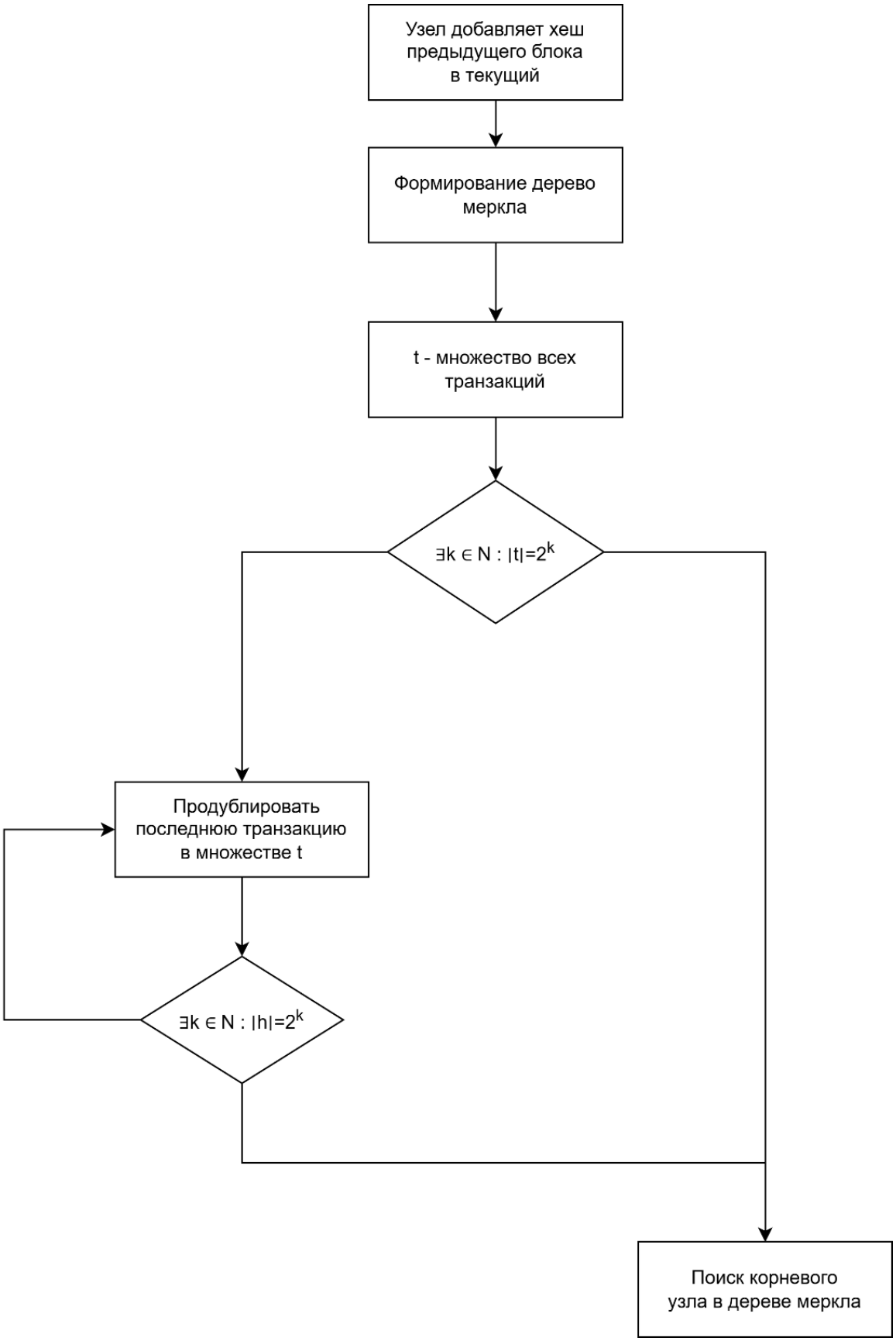


Рисунок 4 – Этап 3. Процесс формирования дерева Меркла

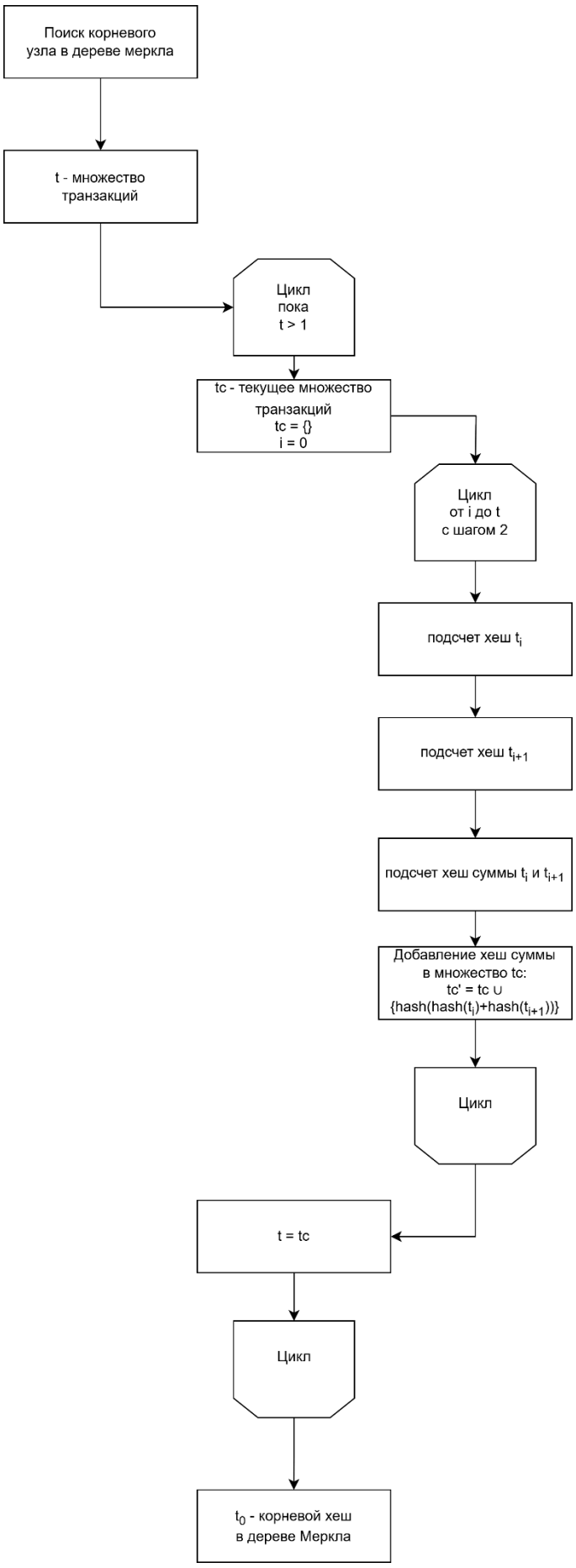


Рисунок 5 – Этап 4. Процесс поиска корневого хэша в дереве Меркла

Этап 5 (Рисунок 6). Процесс поиска хэш-суммы блока и распространение блока по сети. Сначала производится сбор данных блока. Блок включает в себя заголовок (хэш предыдущего блока, корневой хэш дерева Меркла, временную метку и целевую сложность) и транзакции. Затем узел собирает все данные блока и применяет хеш-функцию, чтобы получить хэш блока. Это процесс прямого перебора, где узел меняет значение nonce и пересчитывает хэш до тех пор, пока не найдет подходящий хэш.

После этого начинается процесс распространения блока по сети. Узел, который нашел новый блок, передает его всем своим соседям в сети. Каждый узел, получивший блок, проверяет:

– совпадение хэша предыдущего блока с тем, что хранится в его локальной копии,

– валидность всех транзакций в блоке (подписи, корректность данных и т. д.),

– соответствие хэша блока требованиям сложности сети.

После проверки блок добавляется в копию блокчейна проверяющего узла. Блок передается далее в сеть, и другие узлы также проверяют его и добавляют в свои копии цепи блоков.

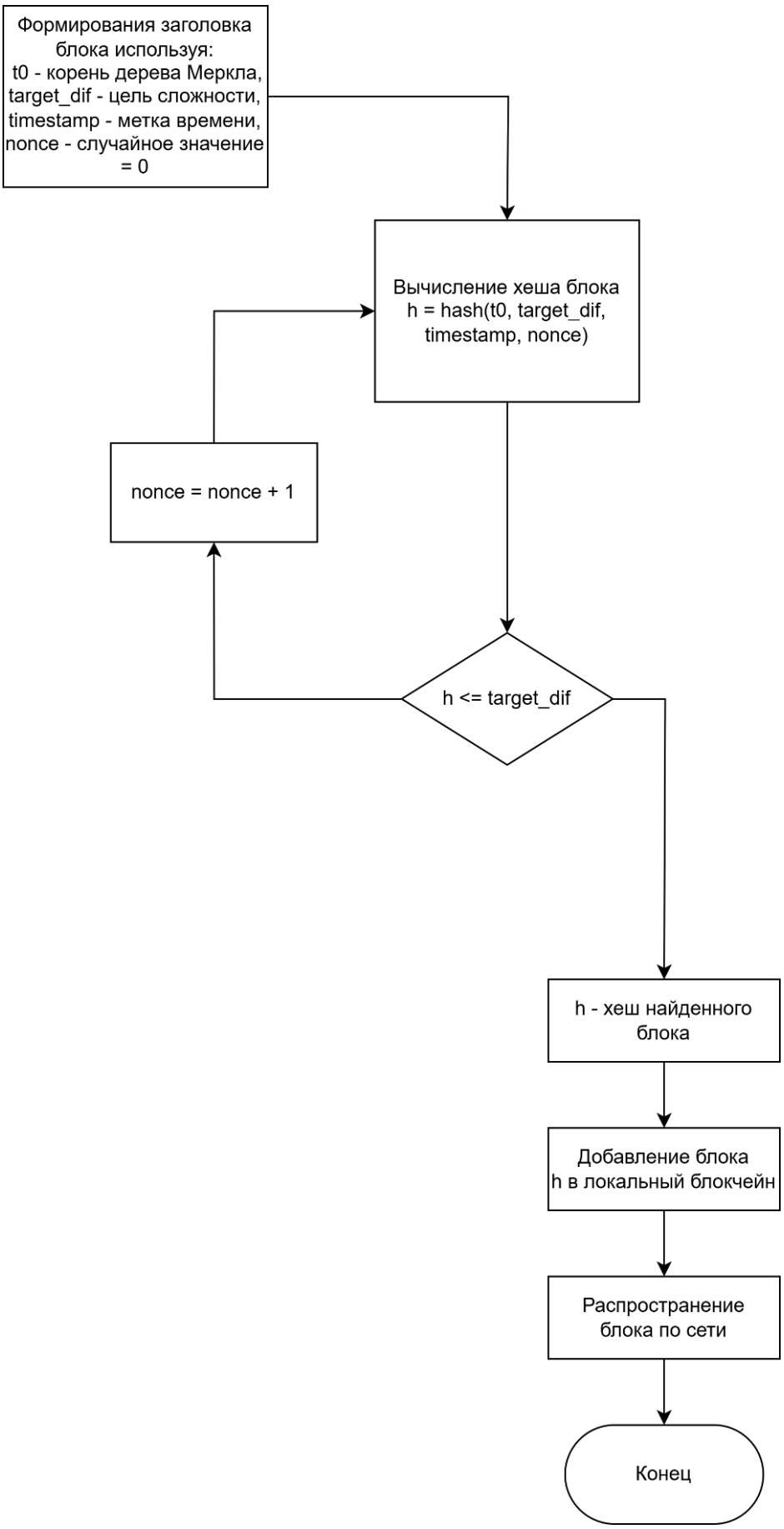


Рисунок 6 – Этап 5. Процесс поиска хэш суммы блока и распространение блока по сети

1.5 Структура сетевого пакета

С сетевой точки зрения передача информации может осуществляться на базе UDP или TCP транспортных протоколов. В рамках представляемого исследования был проведен захват трафика и его анализ, сделанный в приватной блокчейн-сети на основе блокчейн-клиента с открытым исходным кодом, повторяющего конфигурацию сети Ethereum, а также конфигурацию сети TON и сети Cardano. Данные сети были выбраны как наиболее популярные решения.

Захват сетевого трафика производился с применением утилиты Wireshark, а для аналитики пакетов каждого из блокчейна были разработаны специальные диссекторы. Протокольные диссекторы — это модули, которые Wireshark использует для разбора отдельных протоколов, чтобы их можно было интерпретировать по принципу поле за полем. Это позволяет создавать фильтры на основе определенных критериев протокола.

Ethereum был выбран, как решение, которое в преимущественном ряде случаев выступает основой многих существующих блокчейн-сетей по причине открытого исходного кода, виртуальной машины и функциональности смарт-контрактов, построенное на алгоритме консенсуса PoS. Ключевым отличием данной сети является наличие глобального состояния, что в совокупности с открытым исходным кодом позволяет выступать удобной кодовой базой для доработки и внедрения другого функционала. Одноранговая связь между узлами, на которых запущены клиенты Ethereum, выполняется по протоколу devp2p, на основе которого образуется одноранговая сеть [22]. Одноранговые узлы могут предлагать и принимать соединения на любых TCP-портах. Узлы devp2p находят соседние узлы с помощью протокола обнаружения discv4/discv5.

Протокол devp2p имеет два разных режима: основной, который использует в качестве транспортного протокола TCP, и режим обнаружения, который использует в качестве транспортного протокола UDP.

Режим обнаружения предполагаем подключение к определенным серверам, называемыми «узлами начальной загрузки», которые предоставляют некоторый список одноранговых узлов для подключения, чтобы получить информацию о текущих блоках в сети и синхронизироваться. Узлы, к которым произошло подключение для синхронизации также будут предоставлять свои списки узлов, с которыми они соединены. Таким образом появляется возможность сформировать полный список узлов в сети.

Согласно проведенному исследованию, была проанализирована структура пакета devp2p, представленная на Рисунке 7, а пример пакета с передачей транзакции представлен на Рисунке 8.

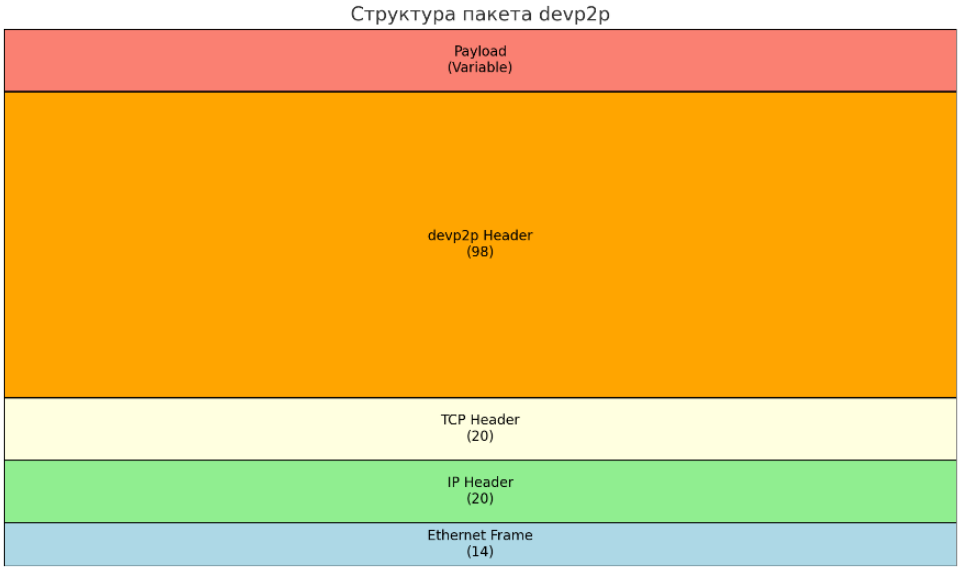


Рисунок 7 – Структура пакета devp2p (в скобках размер в байтах) блокчейна Ethereum

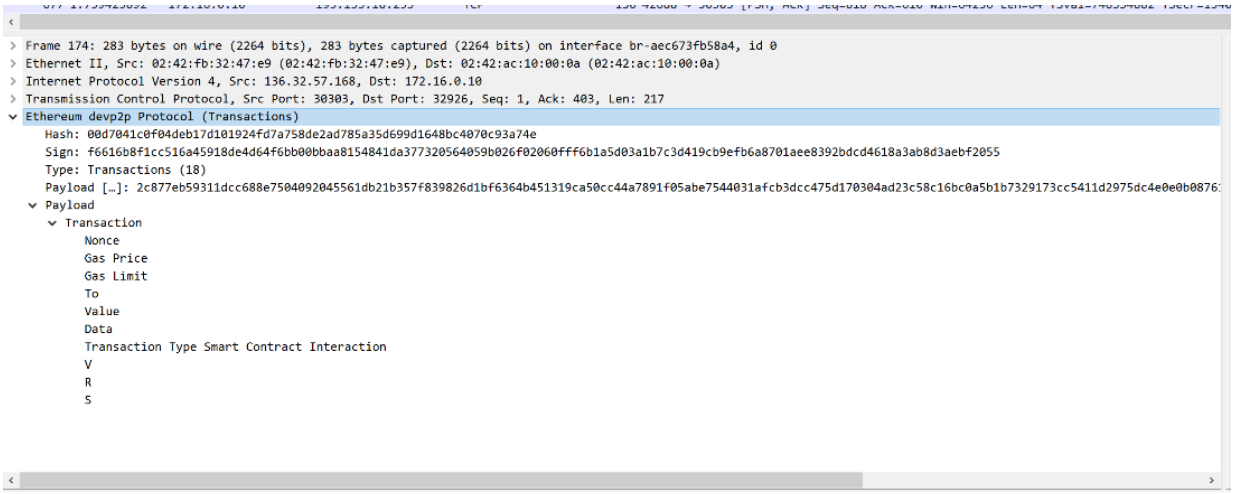


Рисунок 8 – Структура пакета передачи транзакции в блокчейне Ethereum

1) Заголовок TCP/IP.

Компоненты заголовка TCP/IP:

– Ethernet Frame – обертка на уровне канального слоя, включает MAC-адреса отправителя и получателя.

– IP Header – содержит IP-адреса источника и назначения.

– TCP Header – обеспечивает управление соединением, содержит порты отправителя и получателя.

2) Заголовок протокола devp2p.

Заголовок devp2p включает в себя:

– Hash – 32-байтовый хэш сообщения. Используется для проверки целостности пакета.

– Signature (Sign) – подпись отправителя, длина 65 байт. Используется для проверки подлинности отправителя пакета.

– Type – однобайтовый идентификатор типа сообщения, который позволяет определить, что передается: блок, транзакция или сообщение поиска соседних узлов (discovery).

3) Поля полезной нагрузки.

Полезная нагрузка варьируется в зависимости от типа сообщения. В случае передачи транзакции структура следующая:

– Nonce – порядковый номер транзакции,

– Gas Price – цена за единицу газа,

– Gas Limit– лимит газа,

– To – адрес получателя,

– Value – сумма перевода,

– Data – дополнительные данные (например, взаимодействие со смарт-контрактом),

– V, R, S – элементы цифровой подписи для транзакции.

Также следует отметить, что у транзакций перевода после data пустое, а у взаимодействий со смарт-контрактом поле не является пустым, что можно считать

важным признаком для определения типа транзакции по косвенным признакам, если считать, что транзакции взаимодействия со смарт-контрактами будут иметь больший приоритет, чем транзакции перевода.

Важно отметить, что по итогам исследования разных сценариев взаимодействия с блокчейн-сетью было обнаружено, что блоки передаются в разных пакетах, так как могут быть достаточно большими. Размер блока в рамках проводимого исследования достигал 350 Кбайт, что предполагает разбиение на множество пакетов. Размер блока зависит от типа блокчейна, точнее его реализации, а максимальный размер пакета 1500 байт [7].

Также в рамках исследования было выяснено, что классическая транзакция перевода средств зачастую помещаются в 1 пакет, а средний размер транзакции перевода составляет около 245 байт и занимает примерно 40% от размера пакета.

В случае же более сложной транзакции, в рамках которой происходит взаимодействие с методами смарт-контракта составляет в среднем более 350 байт и занимает примерно 50% от размера пакета.

Таким образом, за исключением заголовков в пакете для полезной нагрузки в виде данных блока будет свободно 1348 байт, что составляет примерно 90% от размера пакета. То есть передача одного блока максимального размера в случае сети Ethereum потребует передачи не менее, чем 266 сетевых пакетов. Новые блоки выпускаются в среднем раз в 12-14 секунд. Таким образом, сетевая нагрузка блокчейн-сети в среднем составляет около 20 пакетов в секунду или около 235 Кбит в секунду. Для блоков размером 1 Мбайт эти значения составят 53 пакета или 631 Кбит в секунду.

То есть с учетом изменяемой скорости передачи блоков, параллельной генерации трафика из передаваемых транзакций, а также увеличенном размере блоков нагрузка от блокчейн-сети может составлять достаточно большой объем полосы пропускания канала, так как даже в отсутствие активной нагрузки в виде транзакции сеть будет продолжать выпускать блоки с определенной настройками сети частотой даже с отсутствующим полезным содержимым.

TON — это децентрализованная и открытая интернет-платформа, состоящая из нескольких компонентов. К ним относятся: TON Blockchain, TON DNS, TON Storage и TON Sites. TON Blockchain — это основной протокол, который объединяет базовую инфраструктуру TON, формируя большую экосистему TON.

TON нацелен на достижение широкомасштабной кросс-чейн совместимости, работая в высокомасштабируемой защищенной среде. TON разработан для обработки миллионов транзакций в секунду (TPS) с целью в конечном итоге охватить сотни миллионов пользователей в будущем, построенный на основе алгоритма консенсуса PoS [23].

TON Blockchain разработан как распределенный суперкомпьютер или «суперсервер», предназначенный для предоставления различных продуктов и услуг для содействия развитию децентрализованного видения нового интернета, то означает, что в основе решения лежит виртуальная машина, аналогично принципу построения сетей Ethereum.

Согласно проведенному исследованию, была проанализирована структура пакета передачи данных в блокчейне TON, представленная на Рисунке 9, а пример пакета с передачей транзакции представлен на Рисунке 10.

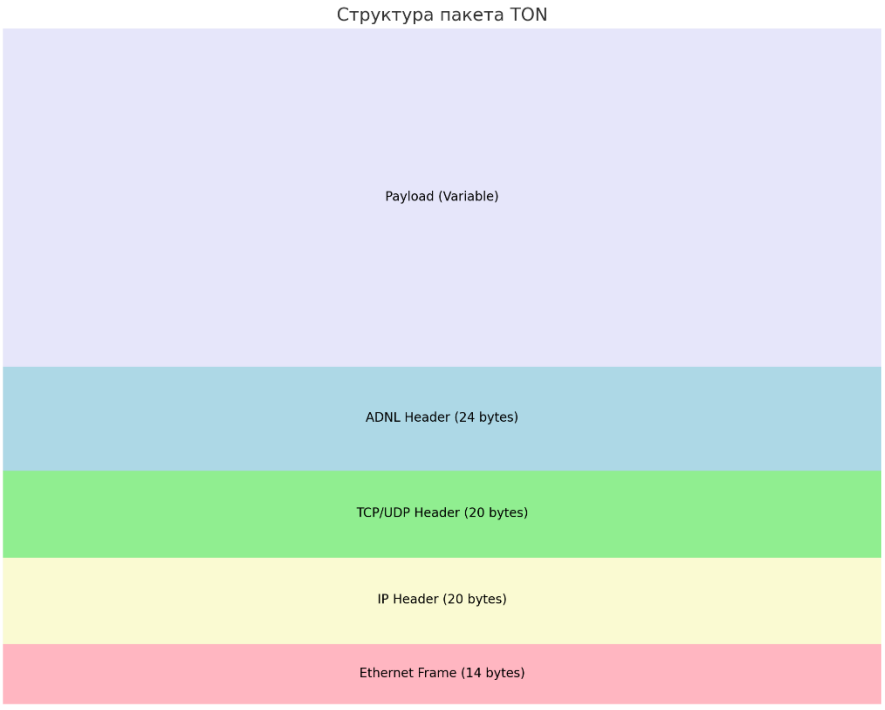


Рисунок 9 – Структура пакета блокчейна TON (в скобках размер в байтах)

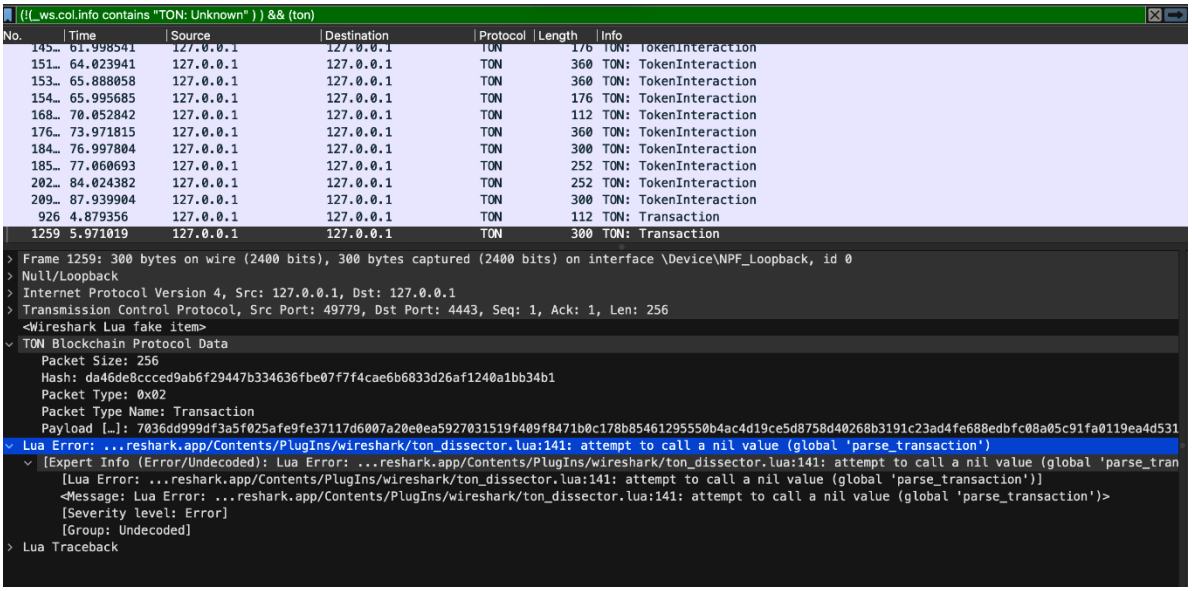


Рисунок 10 – Структура пакета передачи транзакции в блокчейне TON

Структура пакета в блокчейне TON:

1) Заголовок TCP/IP.

Компоненты заголовка:

– Ethernet Frame – обертка на уровне канального слоя, включает MAC-адреса отправителя и получателя (14 байт),

– IP Header – содержит IP-адреса источника и назначения (20 байт для IPv4 без опций),

– TCP/UDP Header – обеспечивает управление соединением, содержит порты отправителя и получателя (20 байт для TCP или 8 байт для UDP).

2) Заголовок протокола ADNL (Abstract Datagram Network Layer).

Компоненты заголовка ADNL:

– Magic Number – уникальный идентификатор протокола (32 бита),

– Session ID – идентификатор сессии для установления безопасного соединения,

– Message Length – длина сообщения в байтах,

– Message Type – однобайтовый идентификатор типа сообщения (например, транзакция, блок, запрос состояния и т.д.),

– Checksum/Hash – контрольная сумма или хэш для проверки целостности пакета.

3) Payload.

Полезная нагрузка зависит от типа сообщения и может включать в себя различные данные.

Размеры пакетов и данных:

– максимальный размер пакета (MTU) составляет 1500 байт для Ethernet,

– заголовки общей сложности:

а) Ethernet Header:14 байт,

б) IP Header 20 байт,

в) TCP/UDP Header 20 байт (TCP) или 8 байт (UDP),

г) ADNL Header около 24 байт (в зависимости от реализации).

– доступно для полезной нагрузки примерно 1446 байт (после вычета заголовков), это около 96% от размера пакета.

Замер среднего размера транзакции также состоял из нескольких этапов. Простой перевод размером около 250–300 байт, примерно 20% от доступной полезной нагрузки.

Взаимодействие со смарт-контрактом размером около 400–600 байт (в зависимости от сложности данных), примерно 30–40% от доступной полезной нагрузки.

Передача блоков более изменчива. Размер блоков может варьироваться от нескольких килобайт до сотен килобайт, в зависимости от количества транзакций и включенных данных. Блоки, превышающие MTU, разбиваются на несколько пакетов и пересобираются на стороне получателя. Также используется специальный механизм сериализации данных под названием Bag of Cells (BoC), который оптимизирует хранение и передачу данных в сети TON. BoC – универсальный двоичный формат сериализации данных в TON, позволяющий эффективно представлять сложные структуры данных и обеспечивать их целостность.

Структуры пакета с транзакцией может состоять из следующим компонентов:

– общий размер пакета до 1500 байт,
– заголовки в сумме около 78 байт (Ethernet + IP + TCP + ADNL),
– полезная нагрузка до 1422 байт.

Cardano — это децентрализованная блокчейн-платформа третьего поколения с алгоритмом консенсуса PoS. Это первая блокчейн-платформа, которая возникла на основе научной философии и подхода, ориентированного на исследования.

Платформа Cardano была разработана с нуля и проверена ведущим в отрасли объединением ведущих инженеров и академических экспертов в области блокчейна и криптографии. Она уделяет особое внимание устойчивости, масштабируемости и прозрачности. Это полностью открытый проект, цель которого — предоставить инклюзивную, справедливую и устойчивую инфраструктуру для финансовых и социальных приложений в глобальном масштабе. Одна из его основных целей — предоставить надежные, безопасные финансовые услуги тем людям, у которых в настоящее время нет доступа.

Cardano был разработан с учетом безопасности как одного из своих основополагающих принципов. Он написан на Haskell, функциональном языке программирования. В функциональном языке, таком как Haskell, поощряется создание вашей системы с использованием чистых функций, что приводит к проектированию, в котором компоненты удобно тестировать изолированно. Кроме того, расширенные возможности Haskell позволяют использовать целый ряд мощных методов для обеспечения корректности кода, таких как реализация на основе формальных и исполняемых спецификаций, обширное тестирование на основе свойств и запуск тестов в симуляции [24].

Платформа смарт-контрактов Cardano стремится предоставить более продвинутые функции, чем любой ранее разработанный протокол, и будет служить стабильной и безопасной платформой для разработки dApps корпоративного уровня. Демократическая система управления Cardano [25], реализуемая на основе

механизмов управления в цепочке CIP-1694, позволит проекту со временем развиваться и устойчиво финансировать себя через дальновидную систему казначейства.

Однако, ключевой особенностью данного блокчейна является модель хранения данных eUTxO [26]. Данная система предполагает отсутствие глобального состояния в системе, позволяя распараллеливать действия с одного аккаунта. В блокчейне Ethereum, например, все действия строго последовательные и число nonce отвечает за отсчет транзакций, производимых с аккаунта пользователя. Транзакция с nonce = N не будет совершена и принята в сеть до тех пор, пока в сети не будет обработана транзакция с nonce = N-1, что свидетельствует о четкой последовательности процесса. В Cardano же ключевой принцип предполагает наличие «входящих» записей. То есть каждый перевод на счет пользователя формирует новую запись. И если следующее действие пользователя суммой перевода равно входящему объему или меньше, то пользователь может враз отправлять несколько транзакций, используя «входящие» записи по-отдельности, что потенциально открывает гораздо большую гибкость, асинхронность и параллельность вычислений и, как результат, большую скорость передачи информации по сети.

Согласно проведенному исследованию, была проанализирована структура пакета передачи данных в блокчейне Cardano, представленная на Рисунке 11, а пример пакета с передачей транзакции представлен на Рисунке 12.

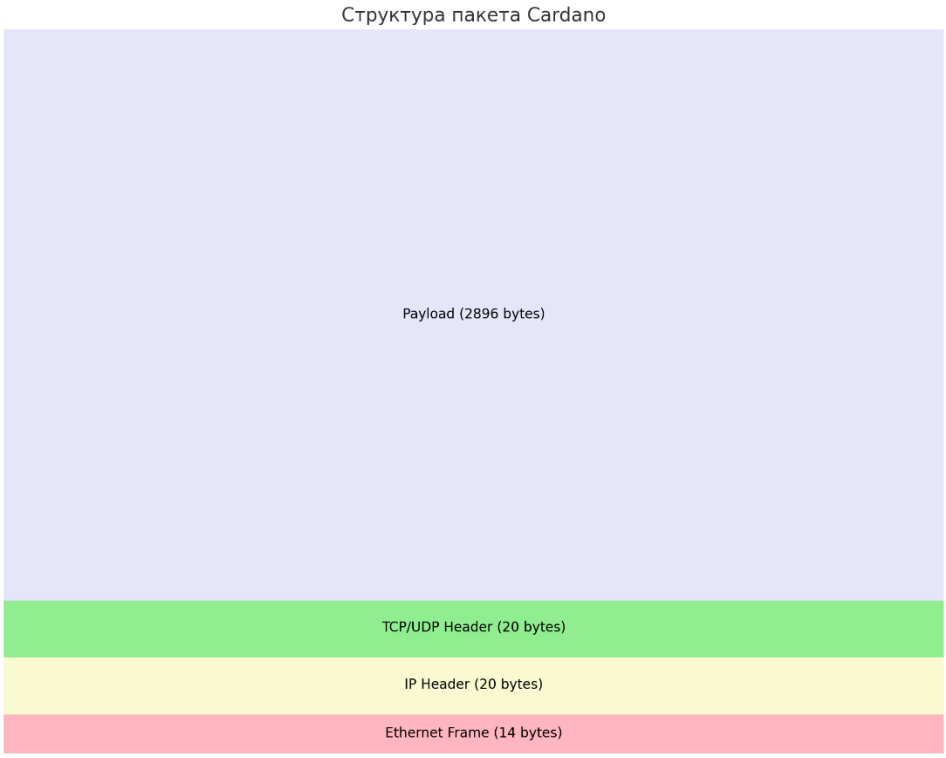


Рисунок 11 - Структура пакета блокчейна TON (в скобках размер в байтах)

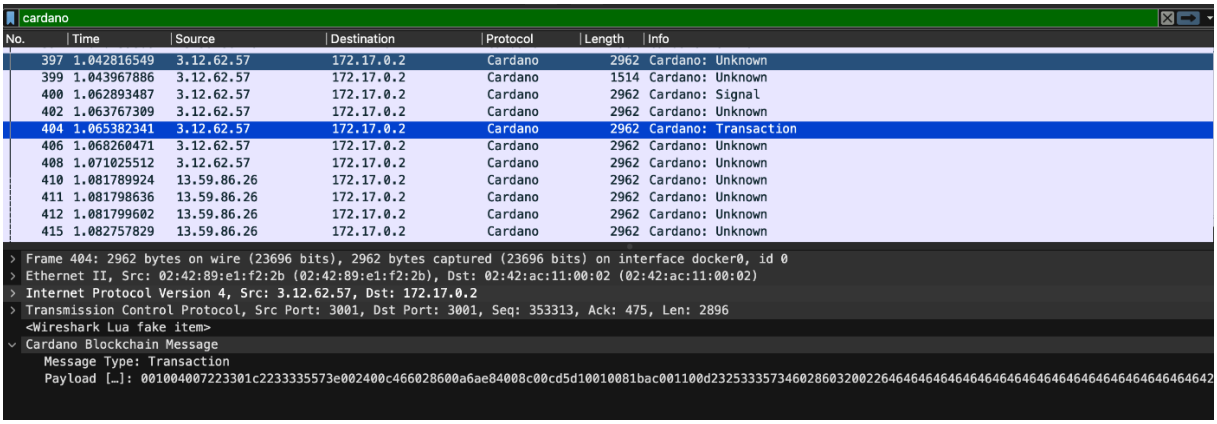


Рисунок 12 – Структура пакета передачи транзакции в блокчейне Cardano

1) Заголовок TCP/IP.

Компоненты заголовка:

– Ethernet Frame – обертка на уровне канального слоя, включает MAC-адреса отправителя и получателя (14 байт),

– IP Header – содержит IP-адреса источника и назначения (20 байт для IPv4 без опций),

– TCP/UDP Header – обеспечивает управление соединением, содержит порты отправителя и получателя (20 байт для TCP или 8 байт для UDP).

2) Заголовок протокола Cardano

Компоненты заголовка Cardano:

– Message Type – идентификатор типа сообщения (например, Handshake, Transaction, Block, Signal),

– Message Length – длина полезной нагрузки сообщения,

– Protocol Version – версия протокола Cardano,

– Network Magic – уникальный идентификатор сети (mainnet, testnet и т.д.).

3) Payload.

Тип Handshake – установление соединения между узлами и согласование параметров протокола.

Поля полезной нагрузки:

– Protocol Version – версия протокола, используемого узлом,

– Software Version – версия программного обеспечения узла,

– Network Magic – идентификатор сети для проверки принадлежности к одной сети,

– Capabilities – список поддерживаемых возможностей и протоколов узла.

Общий размер составляет 2962 байт, размер полезной нагрузки 2896 байт (97% от общ. размера). Пакет Handshake используется для первоначального обмена информацией между узлами. Он обеспечивает согласование версий протокола и проверку принадлежности к одной сети.

Тип Transaction – передача транзакции.

Поля полезной нагрузки:

– Transaction ID – уникальный идентификатор транзакции,

– Inputs – список входов транзакции (UTXO, которые расходуются),

– Outputs – список выходов транзакции (адреса получателей и суммы),

– Amount – сумма перевода,

– Fee – комиссия за транзакцию,

– TTL – время жизни транзакции до истечения,

– Metadata – дополнительные данные, связанные с транзакцией,

– Signatures – цифровые подписи, подтверждающие права на входы.

Общий размер 2962 байт. Размер полезной нагрузки 2896 байт (97% от общ. размера). Пакет с транзакцией содержит всю необходимую информацию для проверки и последующего включения транзакции в блок. Размер транзакции позволяет ей помещаться в один пакет.

Тип Block – передача блока, содержащего транзакции, для добавления в блокчейн.

Поля полезной нагрузки:

– Block Header:

а) Block Number – номер блока в цепочке,

б) Previous Block Hash – хэш предыдущего блока,

в) Merkle Root – корневой хэш дерева Меркла для транзакций,

г) Slot Number – номер слота в протоколе Ouroboros,

д) Timestamp – время создания блока,

е) Block Size – размер блока,

ж) Protocol Version – версия протокола для блока.

– Transaction Bodies – список транзакций, включенных в блок,

– Block Signature – подпись производителя блока,

– VRF Proof – доказательство выбора лидера слота.

Общий размер 2962 байт. Размер полезной нагрузки 2896 байт (97% от общ. размера). Блоки много весят и часто передаются в нескольких пакетах. Одиночный пакет содержит заголовок блока и только часть транзакций.

Тип Signal – передача служебных сигналов для поддержания соединения и обмена статусами.

Поля полезной нагрузки:

– Signal Type – тип сигнала (Keep-Alive, Ping, Acknowledgment),

– Timestamp – время отправки сигнала,

– Node Status – текущий статус узла (загрузка, готовность принимать данные),

– Additional Data – дополнительные сведения или параметры.

Общий размер 2962 байт. Размер полезной нагрузки 2896 байт (97% от общ. размера). Сигнальные сообщения используются для проверки доступности узлов и поддержания активного соединения. Они обнаруживают сбои связи и обеспечивают реакцию на изменения в сети.

Следует отметить, что максимальный размер Ethernet-пакета обычно составляет 1500 байт, а использование более крупных пакетов (2962 байта) в Cardano может говорить о применении агрегирования пакетов. Средний размер транзакции в Cardano позволяет передавать ее в одном пакете, аналогично Ethereum. В Таблице 1 представлены результаты исследования сетевой нагрузки от наиболее популярных блокчейн-сетей. Наибольшее значение имеет оценка предельной сетевой нагрузки, рассчитанная из соотношения максимального размера блока и среднего времени его генерации.

Таблица 1 – Сравнение сетевой нагрузки и характеристик сетевых пакетов блокчейнов Ethereum, TON, Cardano

	Ethereum	TON	Cardano
Размер блока макс.	12 Мб	2 Мб	88 Кб
Размер транзакции макс.	780 Кб	64 Кб	16 Кб
Среднее время создания блока	15 секунд	5 секунд	20 секунд
Общий размер (полезный) нагрузки в байтах для транзакции	1500 (1348)	1500 (1422)	2962 (2896)

Продолжение Таблицы 1

	Ethereum	TON	Cardano
Общий размер (полезный) нагрузки в байтах для блока	1500 (1348)	1500 (1446)	2962 (2896)
% пакета под полезную нагрузку транзакции	90	94,8	97
% пакета под полезную нагрузку блока	90	96	97
Сетевая нагрузка при передаче блоков	9335 пакетов на 1 блок / 623 пакета в секунду	1450 пакетов на 1 блок / 290 пакетов в секунду	32 пакета на 1 блок / 1,5 пакета в секунду
Предельная сетевая нагрузка при передаче блоков	7,476 Мбит/с	3,48 Мбит/с	36,9 Кбит/с

Беря во внимание средние типовые скорости для различных способов доступа в Интернет, приведенные в Таблице 1, для ряда технологий подключения выявленная сетевая нагрузка может оказаться критичной и занять всю полосу пропускания. Указанные в Таблице 2 скорости являются усредненными и могут отличаться в зависимости от провайдера, инфраструктуры, загруженности сети и других факторов.

Таблица 2 – Скорости подключения различных технологий доступа в Интернет

Способ доступа в интернет	Средняя скорость загрузки (Download)	Средняя скорость отправки (Upload)
Dial-up модем (устаревшая технология)	56 Кбит/с	33–48 Кбит/с
DSL цифровая абонентская линия (зависит от расстояния до узла провайдера)	1–20 Мбит/с	0,5–3 Мбит/с

Продолжение Таблицы 2

Способ доступа в интернет	Средняя скорость загрузки (Download)	Средняя скорость отправки (Upload)
Кабельный интернет	50–1000 Мбит/с	5–50 Мбит/с
Оптоволоконный интернет (FTTH)	100–2000 Мбит/с	100–2000 Мбит/с
4G (LTE)	20–150 Мбит/с	5–50 Мбит/с
5G	100–1000 Мбит/с	50–500 Мбит/с
Спутниковый интернет	20–250 Мбит/с	3–20 Мбит/с
Wi-Fi (зависит от стандарта (802.11n/ac/ax) и качества оборудования)	10–1000 Мбит/с	10–1000 Мбит/с
PLC (интернет через электросеть)	2–500 Мбит/с	2–500 Мбит/с
Bluetooth (для интернета)	1–3 Мбит/с	1–3 Мбит/с

1.6 Выводы

1) Представлен базовый сценарий процессов блокчейн-сети.

2) Проведено исследование сетевого трафика при синхронизации узлов сетей в контексте загрузки блоков Ethereum, TON, Cardano, отправки транзакции. Приведена структура сетевых пакетов с трафиком блокчейн. Проведен анализ заголовков и поля полезной нагрузки при передаче блока транзакций и транзакции самостоятельно.

3) Проведен расчет и сравнение оказываемой нагрузки на полосу пропускания при передаче блокчейн-трафика. Оценен объем в соотношении с разными типами доступа в сеть Интернет.

ГЛАВА 2 ВЛИЯНИЕ АЛГОРИТМОВ КОНСЕНСУСА НА СЕТИ СВЯЗИ

Алгоритмы консенсуса являются неотъемлемой частью блокчейн-систем и могут обладать достаточно вариативной логикой и ключевой механикой принятия решения. Эти параметры оказывают непосредственное влияние на объем генерируемого трафика, дополнительные действия от других участников сети (валидация или голосование), что определяет необходимость его дублирования иди генерации дополнительной нагрузки для соответствия требованиям механизма консенсуса.

2.1 Понятие алгоритма консенсуса

Алгоритм консенсуса — это набор правил, используемый в децентрализованных системах, который позволяет участникам (узлам) сети согласовывать состояние данных и достигать единого решения относительно добавления новых записей (например, транзакций или блоков) в распределенный реестр. Он обеспечивает согласие между узлами, предотвращая конфликты и обеспечивая целостность и безопасность данных [27].

Основной функцией алгоритма консенсуса является обеспечение гарантии согласия, то есть того, что все узлы сети имеют одинаковую версию данных, что критически важно для децентрализованных систем, а также защиты от сбоев и атак, таких как двойные траты, обеспечивая, что только валидные транзакции могут быть добавлены в блокчейн.

Алгоритм консенсуса играет ключевую роль в функционировании блокчейн-сети, так как позволяет узлам проверять и согласовывать транзакции, добавляемые в блоки в асинхронной системе без доверия [28]. Важным также является разность в скорости создания блоков и поиска удовлетворяющего условиям хэша в сравнении с быстрой скоростью валидации хэша.

Работа алгоритма консенсуса может быть разбита на несколько ключевых этапов:

1) Инициация. Узел (или узлы) инициируют процесс консенсуса, предлагая новую транзакцию или блок для добавления в блокчейн.

2) Распространение данных. Предложенные данные (транзакции или блоки) рассылаются другим узлам сети для проверки.

3) Проверка. Каждый узел проверяет данные на корректность и соответствие правилам сети. Это может включать проверку цифровых подписей, наличие достаточных средств и другие условия.

4) Голосование. Узлы голосуют за принятие или отклонение предложенных данных. В зависимости от алгоритма консенсуса, это может происходить различными способами: от решения математической задачи (подбора хэша) до прегенерируемого времени ожидания своей очереди на генерацию блока.

5) Достижение консенсуса. Узлы достигают согласия по поводу состояния сети. В зависимости от алгоритма, может потребоваться определенный процент голосов.

6) Добавление в блокчейн. После достижения консенсуса новый блок добавляется в цепочку. Узлы обновляют свои копии блокчейна, синхронизируя состояние данных.

7) Обновление состояния. Система обновляет свое состояние, и узлы продолжают работать с новой информацией.

2.2 Семейства алгоритмов консенсуса

На сегодняшний день существует достаточно большое количество вариаций алгоритмов консенсуса. Безусловно, полностью новые механики реализовать достаточно сложно, по этой причине многие алгоритмы консенсуса повторяют некоторую уже существующую логику, преобразуя ее, оптимизируя или

подстраивая под специфичные условия. Поэтому логично применять категоризацию по подобию поведения, общим принципам поведения [5].

Предлагаемая категоризация содержит следующие консенсусы по принципу ключевого механизма работы [27]:

– алгоритмы византийской отказоустойчивости [29, 30]:

а) BFT (Byzantine Fault Tolerance),

б) FBA (Federated Byzantine Agreement),

в) PBFT (Practical Byzantine Fault Tolerance),

г) DBFT (Delegated Byzantine Fault Tolerance),

д) SBFT (Simplified Byzantine Fault Tolerance).

– PoS-подобные алгоритмы [31]:

а) DPoS (Delegated Proof of Stake),

б) LPoS (Leased Proof of Stake),

в) PoAuth (Proof of Authority),

г) PoI (Proof of Importance),

д) PoWeight (Proof of Weight),

е) PoR (Proof of Reputation).

– PoW-подобные алгоритмы:

а) PoB (Proof of Burn),

б) PoET (Proof of Elapsed Time),

в) PoC (Proof of Capacity),

г) PoA (Proof of Activity)

– Альтернативные алгоритмы с уникальными механиками:

а) PoRes (Proof of Research),

б) DAG (Directed Acyclic Graph).

Так как консенсусы во многом повторяют основную механику за исключением некоторых оптимизаций, для дальнейшего исследования были выбраны следующие алгоритмы консенсуса: PoW, PoS, DPoS, PBFT, PoR, PoA, PoET, FBA. Эти алгоритмы реализуют ключевые механики из трех категорий. Категория альтернативных алгоритмов не учитывалась, так как механизмы внутри

алгоритмов практически не приспособлены к среднестатистическим действиям в блокчейне и специализируются на конкретных частных и редких ситуациях.

Для исследования влияния алгоритмов консенсуса на сеть связи было проведено моделирование с использованием программного языка Python. В рамках моделирования в программном коде были реализованы классы алгоритмов консенсуса, описывающие ключевую логику механизма достижения консенсуса.

На Рисунке 13 приведен результат моделирования, демонстрирующий влияние консенсусов PoW, PoS, PoET, DPoS на заполнение пула неподтвержденных транзакций, как наиболее показательные с точки зрения объема демонстрируемых данных. Накопление пула свидетельствует об образовании очереди в системе. А его переполнение о потерях транзакций, так как они не могут быть записаны в пуле неподтвержденных транзакций, то есть не будут доступны узлам для обработки и включения в блок.

По части графика, демонстрирующего заполнением пула неподтвержденных транзакций можно сделать вывод, что:

– PoW: постоянное накопление транзакций с периодической очисткой каждые 15 секунд,

– PoS: более частая очистка пула неподтвержденных транзакций каждые 5 секунд, что снижает уровень накопления,

– PoET: средний уровень заполнения пула неподтвержденных транзакций с более равномерным распределением очисток,

– DPoS: почти полное отсутствие накопления транзакций из-за высокой частоты обработки.

По части графика, демонстрирующего использование полосы пропускания можно сделать вывод, что:

– PoW: высокие всплески каждые 15 секунд, отражающие передачу больших блоков,

– PoS и PoET: умеренные и равномерные всплески за счет меньших интервалов между блоками,

– DPoS: низкие и частые передачи данных, минимизирующие нагрузку на сеть.

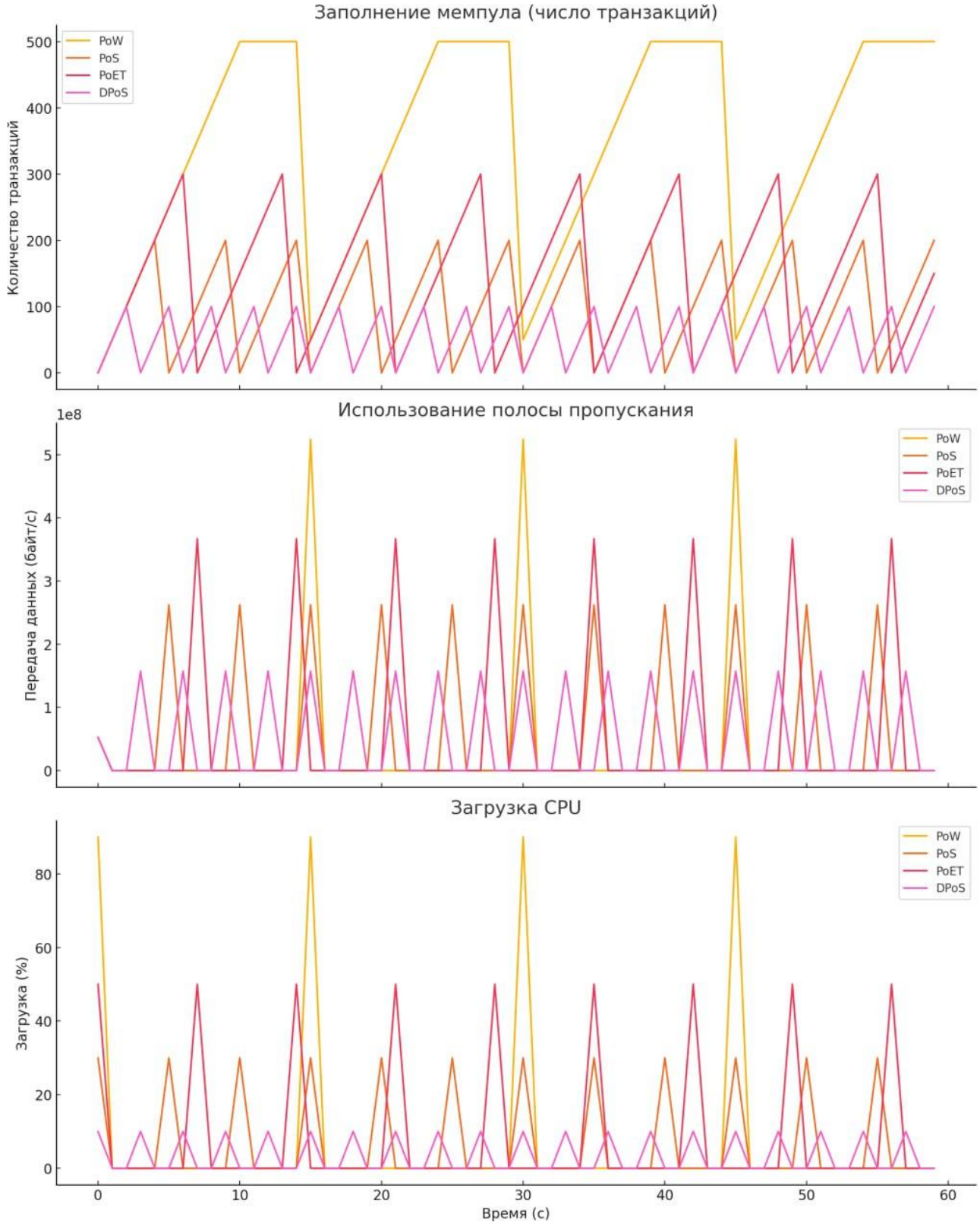


Рисунок 13 – График оценки заполнения пула неподтвержденных транзакций, использования полосы пропускания, загрузки CPU в рамках моделирования алгоритмов консенсуса PoW, PoS, PoET, DPoS

По части графика, демонстрирующего загрузку центрального процессора можно сделать вывод, что:

– PoW: значительная нагрузка на процессор из-за вычислений хэшей,

– PoS и PoET: умеренная нагрузка, отражающая меньшую сложность вычислений,

– DPoS: минимальная нагрузка, так как вычисления практически отсутствуют.

Эти результаты иллюстрируют различия в производительности и влиянии алгоритмов консенсуса на сеть и ресурсы узла и явные отличия в воздействии на сеть и аппаратные компоненты. Также следует отметить преимущественно негативные характеристики алгоритма консенсуса PoW, однако, неоспоримым его преимуществом является механика регуляции сложности, что провоцирует долгую сборку блока, но снимает нагрузку с валидаторов в сети, что также влияет на скорость распространения. Также данный алгоритм будет применяться при подозрительной транзакционной нагрузке на узел, которая может быть интерпретирована как атака семейства Denial of Service. В таком случае нагрузка равномерно распределяется по узлам и дальнейшие процессы регулируются сложностью.

Проведенное моделирование подтверждает явные отличия влияния алгоритмов консенсуса на сеть связи и возможность регуляции нагрузки с помощью адаптивного подбора наиболее эффективного алгоритма консенсуса в текущих сетевых условиях.

2.3 Нагрузка на аппаратное обеспечение

В рамках оценки нагрузки на аппаратное обеспечение был осуществлен эксперимент с замером нагрузки на центральный процессор, оперативную память и жесткий диск в момент генерации пустых блоков развернутой приватной блокчейн-сети с классической конфигурацией, предусматривающей генерацию

блоков каждые 12-14 секунд, размером блока, достигающего 350 Кбайт (пустой блок размером около 152 байт), алгоритмом консенсуса PoS. Пустые блоки использованы в исследовании для демонстрации минимальных требований в случае рассматриваемой реализации алгоритма PoS. На Рисунке 14 приведены графики замера нагрузки на аппаратные компоненты оборудования.

По результатам эксперимента можно сделать вывод о том, что учет аппаратной нагрузки является необходимым, так как вычислительные затраты могут оказывать значительное влияние, а также замедлять процесс обработки других сетевых пакетов, не связанных с работой блокчейн-сети, так как нагрузка при алгоритме консенсуса PoS даже с пустыми блоками является сравнительно большой и альтернативный алгоритм консенсуса был бы более выгодным по используемым ресурсам в момент простоя.

Рисунок 14 - Графики замера нагрузки блокчейн-сети при генерации пустых блоков на центральный процессор, жесткий диск и оперативную память (количество пакетов, загрузка сети, загрузка жесткого диска, загрузка процессора, загрузка оперативной памяти)

Для корректности эксперимента был произведен замер небольшой нагрузки, средней нагрузки и большой нагрузки на тот же узел сети. Для этого был реализован модельный стенд, схема которого приведена на Рисунке 15.

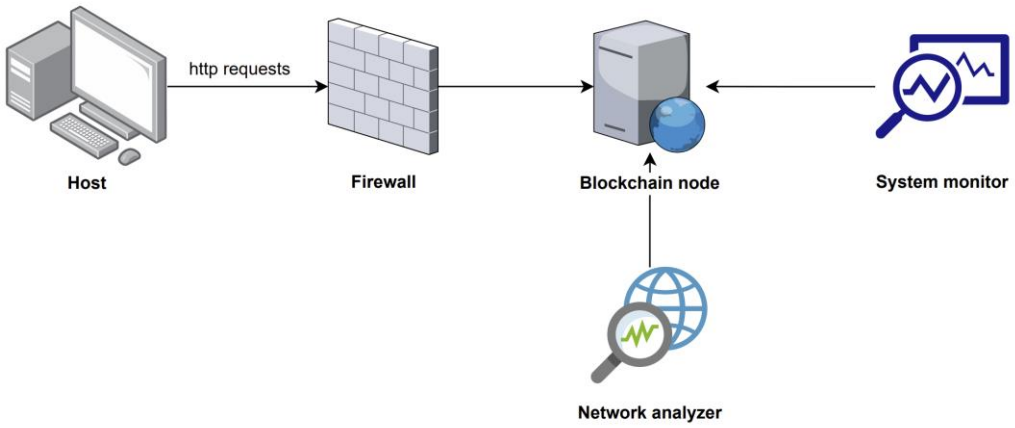


Рисунок 15 – Схема испытательного стенда для замера изменения аппаратных параметров оборудования с узлом блокчейн-сети

Для оценки реакции аппаратного обеспечения сервера, где был развернут узел блокчейн-сети был проведен тест с 10, 1000 и 1500 параллельными потоками, которые безостановочно вызывали метод sendTransaction, то есть генерировали транзакционную нагрузку. По итогам замеров было выяснено, что в 1 потоке за 1 секунду отправляется в среднем 7 транзакций. То есть количество запросов в секунду к узлу блокчейн-сети в рамках 1 потока составляет 7. На 10, 1000 и 1500 потоков выходит 70, 7000 и 10500 вызовов метода отправки транзакций в секунду.

Схема отправки запросов в рамках параллельных потоков представлена на Рисунке 16.

Сначала в основном потоке происходит подключения к узлу блокчейн-сети, затем подготовка данных для транзакций (газ, адреса. приватный ключ для подписания танзакции) и затем в отдельных потоках идет отправка транзакций.

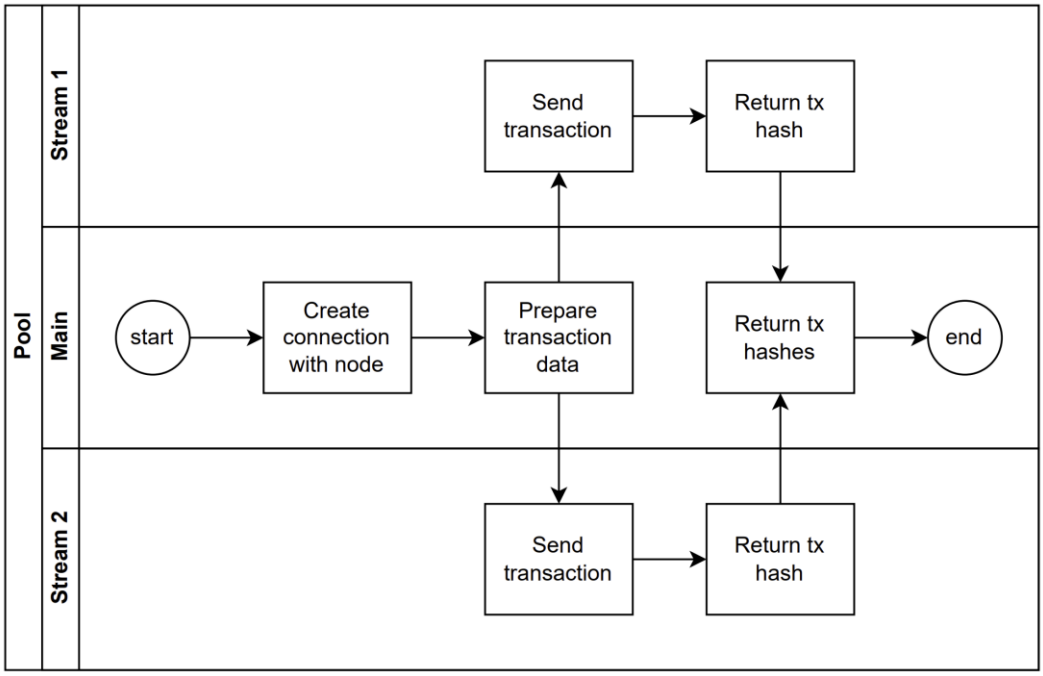


Рисунок 16 – Схема отправки транзакций в рамках параллельных потоков

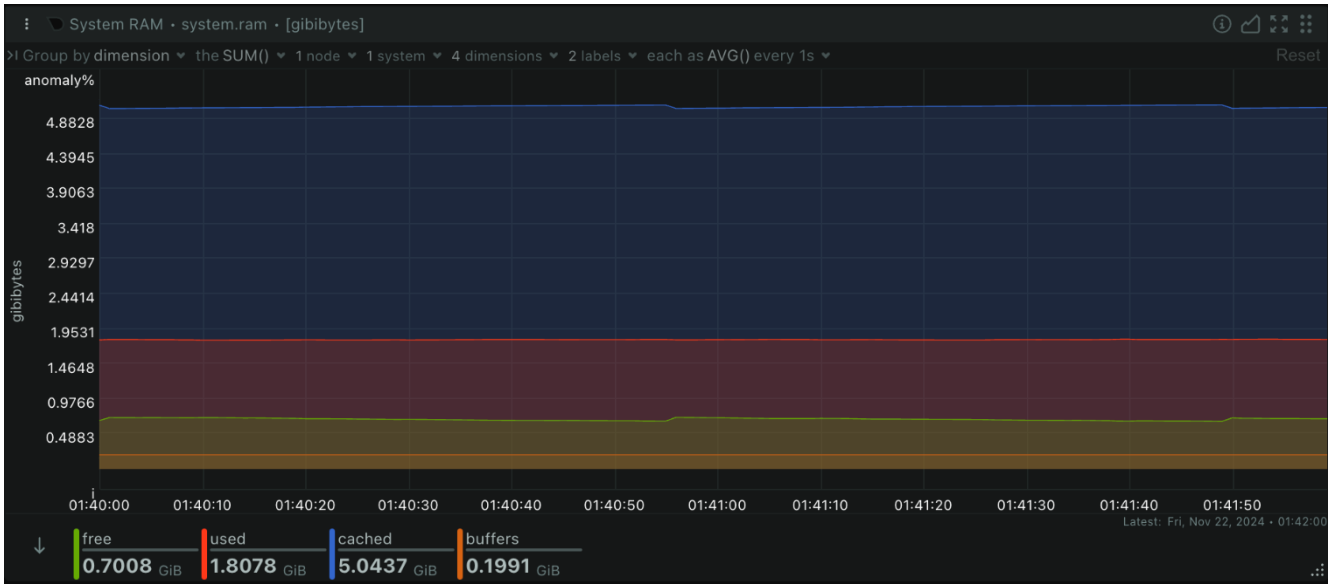
На Рисунке 17 приведены результаты замера нагрузки на аппаратное обеспечение при нагрузке узла

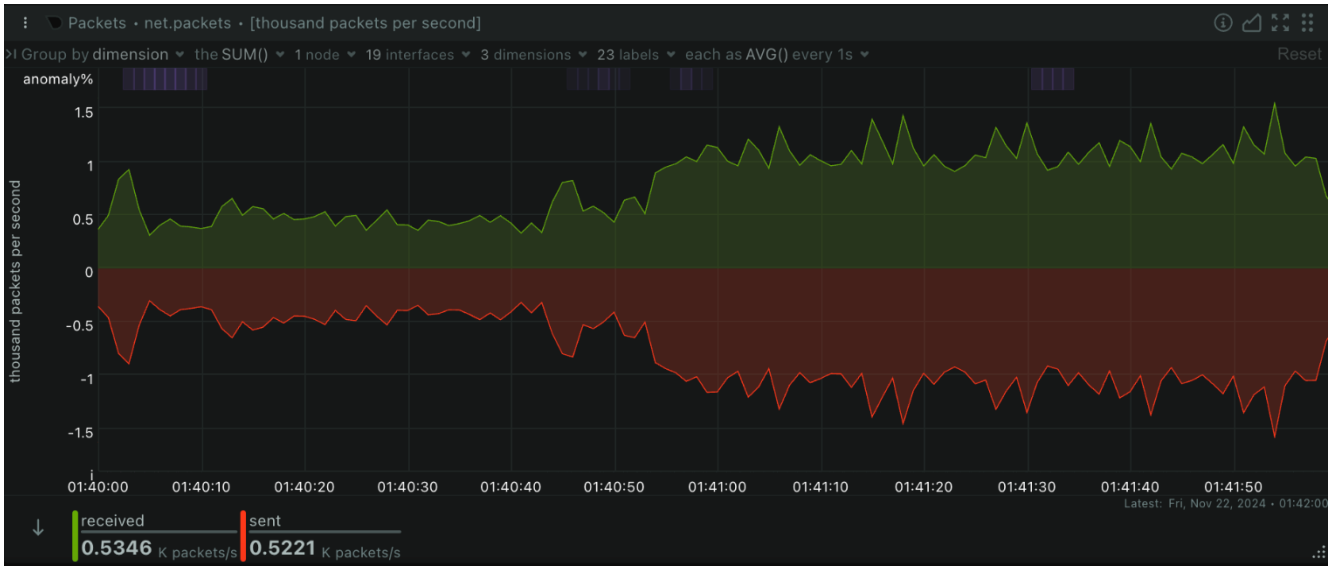
При тесте с 10 параллельными потоками графики нагрузок аналогичны состоянию покоя, так как нагрузка сравнительно небольшая. CPU, RAM без заметных изменений, наблюдается небольшой рост объема сетевого трафика и количества пакетов (в пике достигает 1300 пакетов в секунду).

На Рисунке 18 представлены результаты реакции аппаратных компонентов при старте эксперимента с 1000 параллельных потоков.

При тесте с 1000 параллельными потоками на графиках наблюдается значительный рост метрик CPU резко растет с 8-10% до 75-78%. Потребление RAM вырастает почти в 2 раза - с 1.8 Гб до 3 Гб. Также наблюдается значительном рост объема сетевого трафика и количества пакетов (в пике достигает 26000 пакетов в секунду).

На Рисунке 19 демонстрируется результат аналогичного эксперимента, но с запуском 1500 параллельных потоков.


System RAM • system.ram • [gibibytes]
Group by dimension the SUM() 1 node 1 system 4 dimensions 2 labels each as AVG() every 1s
Reset
anomaly%
4.8828
4.3945
3.9063
3.418
2.9297
2.4414
1.9531
1.4648
0.9766
0.4883
gibibytes
01:40:00 01:40:10 01:40:20 01:40:30 01:40:40 01:40:50 01:41:00 01:41:10 01:41:20 01:41:30 01:41:40 01:41:50
Latest: Fri, Nov 22, 2024 • 01:42:00
free 0.7008 GiB
used 1.8078 GiB
cached 5.0437 GiB
buffers 0.1991 GiB


Packets • net.packets • [thousand packets per second]
Group by dimension the SUM() 1 node 19 interfaces 3 dimensions 23 labels each as AVG() every 1s
Reset
anomaly%
1.5
1
0.5
0
-0.5
-1
-1.5
thousand packets per second
01:40:00 01:40:10 01:40:20 01:40:30 01:40:40 01:40:50 01:41:00 01:41:10 01:41:20 01:41:30 01:41:40 01:41:50
Latest: Fri, Nov 22, 2024 • 01:42:00
received 0.5346 K packets/s
sent 0.5221 K packets/s

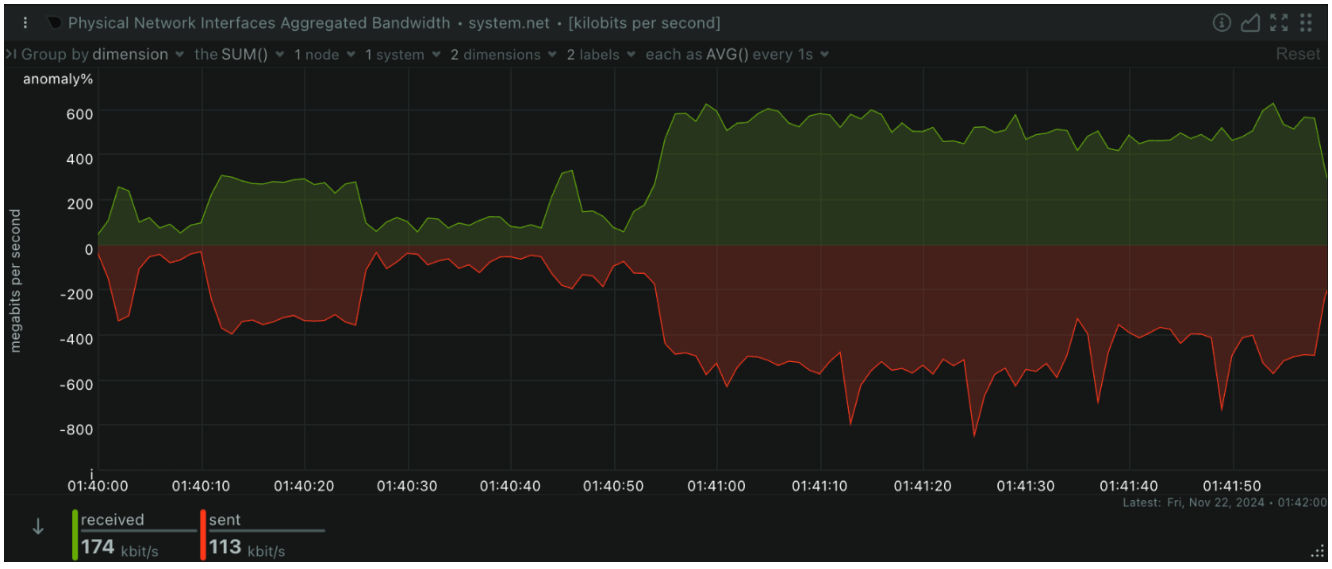
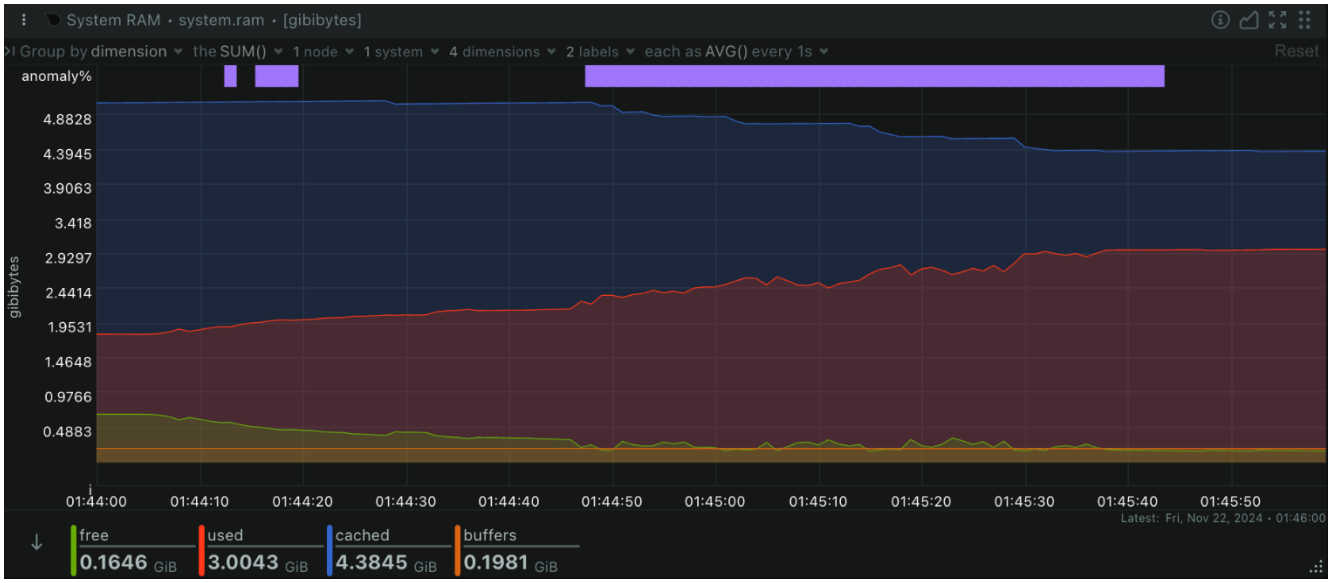
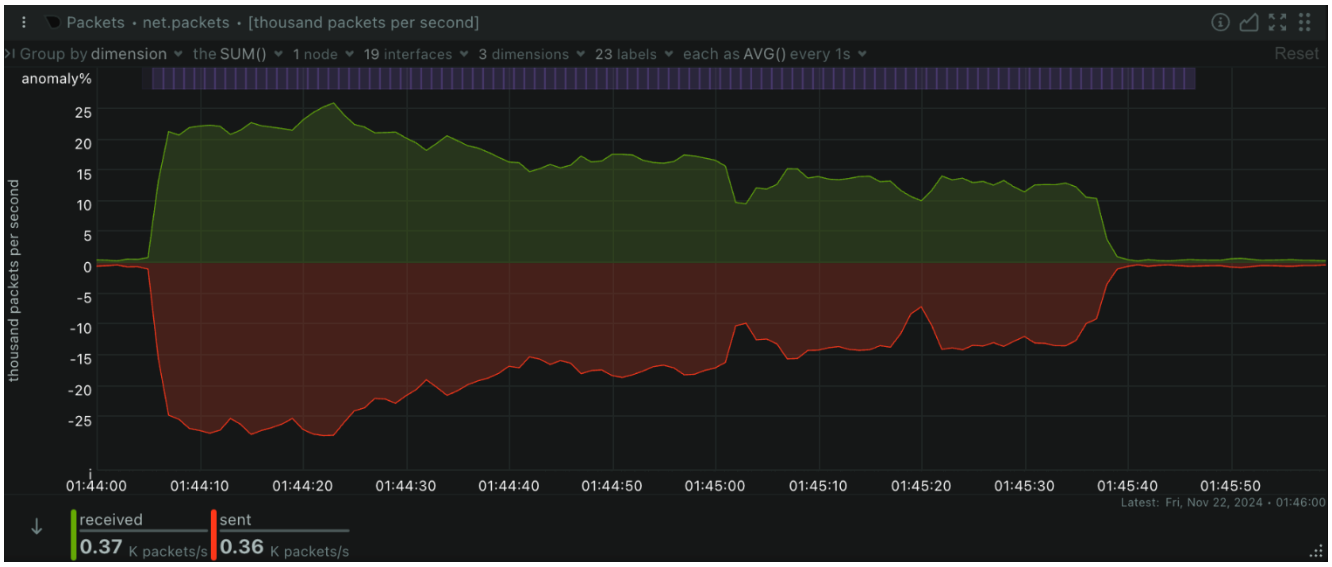

Physical Network Interfaces Aggregated Bandwidth • system.net • [kilobits per second]
Group by dimension the SUM() 1 node 1 system 2 dimensions 2 labels each as AVG() every 1s
Reset
anomaly%
600
400
200
0
-200
-400
-600
-800
megabits per second
01:40:00 01:40:10 01:40:20 01:40:30 01:40:40 01:40:50 01:41:00 01:41:10 01:41:20 01:41:30 01:41:40 01:41:50
Latest: Fri, Nov 22, 2024 • 01:42:00
received 174 kbit/s
sent 113 kbit/s

Рисунок 17 – Результаты замеров нагрузки при 10 параллельных потоках

При тесте с 1500 параллельными потоками на графиках наблюдается рост метрик, CPU так же растет до 75-78%, в пиковых значениях доходит до 80%. Потребление RAM растет до 3.8 Гб. Значительно растет объем сетевого трафика и количество пакетов (в пике достигает 17400 пакетов в секунду).

При тесте 1000 параллельных потоков нагрузка на сеть больше и потребление RAM меньше, чем при 1500 параллельных потоках, это говорит о том, что срабатывают механизмы, помогающие справляться с высокими нагрузками на сеть, что является особенностями реализации конкретного блокчейн-клиента.


System RAM • system.ram • [gibibytes]
Group by dimension the SUM() 1 node 1 system 4 dimensions 2 labels each as AVG() every 1s
Reset
anomaly%
4.8828
4.3945
3.9063
3.418
2.9297
2.4414
1.9531
1.4648
0.9766
0.4883
gibibytes
01:44:00
01:44:10
01:44:20
01:44:30
01:44:40
01:44:50
01:45:00
01:45:10
01:45:20
01:45:30
01:45:40
01:45:50
Latest: Fri, Nov 22, 2024 • 01:46:00
free
0.1646 GiB
used
3.0043 GiB
cached
4.3845 GiB
buffers
0.1981 GiB


Packets • net.packets • [thousand packets per second]
Group by dimension the SUM() 1 node 19 interfaces 3 dimensions 23 labels each as AVG() every 1s
Reset
anomaly%
25
20
15
10
5
0
-5
-10
-15
-20
-25
thousand packets per second
01:44:00
01:44:10
01:44:20
01:44:30
01:44:40
01:44:50
01:45:00
01:45:10
01:45:20
01:45:30
01:45:40
01:45:50
Latest: Fri, Nov 22, 2024 • 01:46:00
received
0.37 K packets/s
sent
0.36 K packets/s

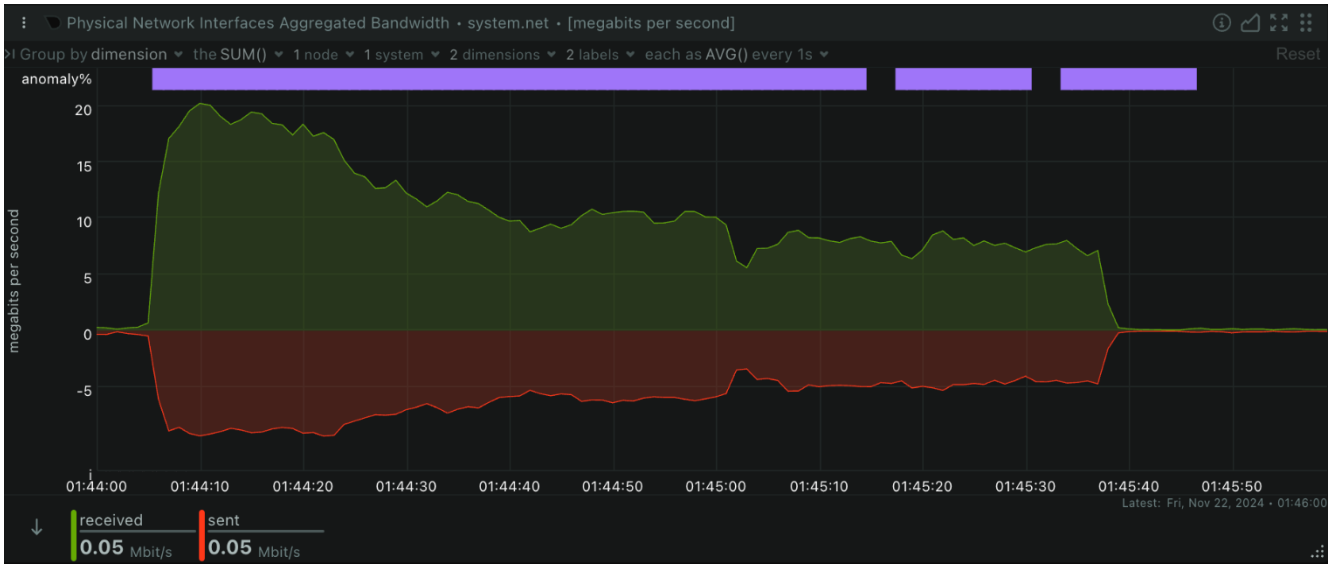
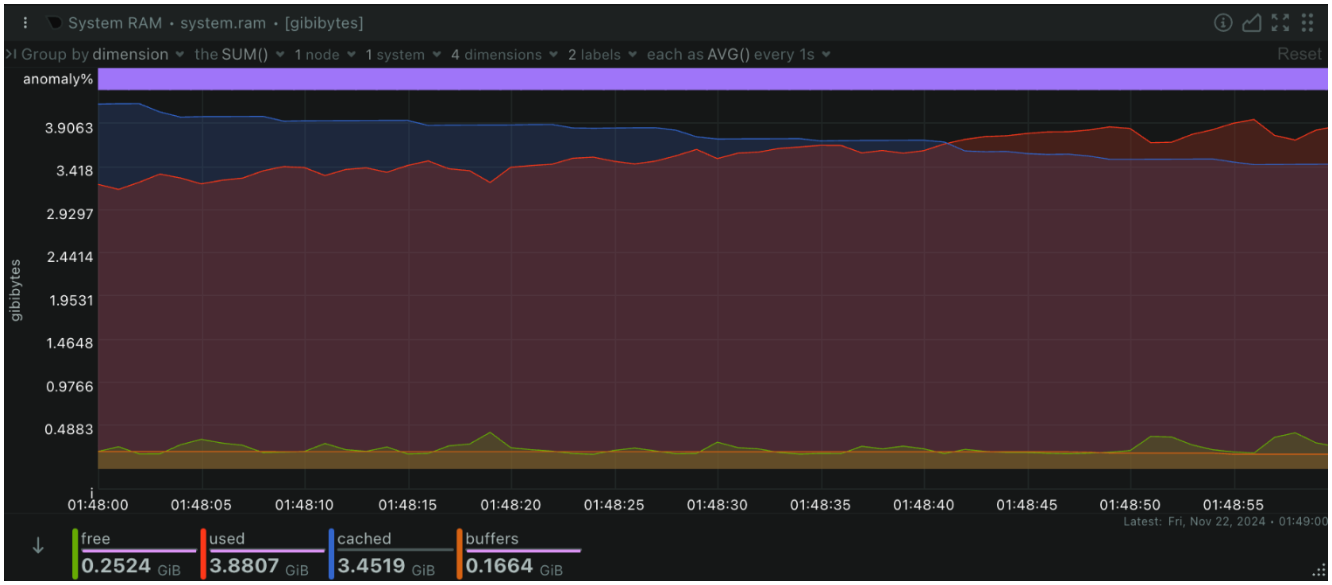
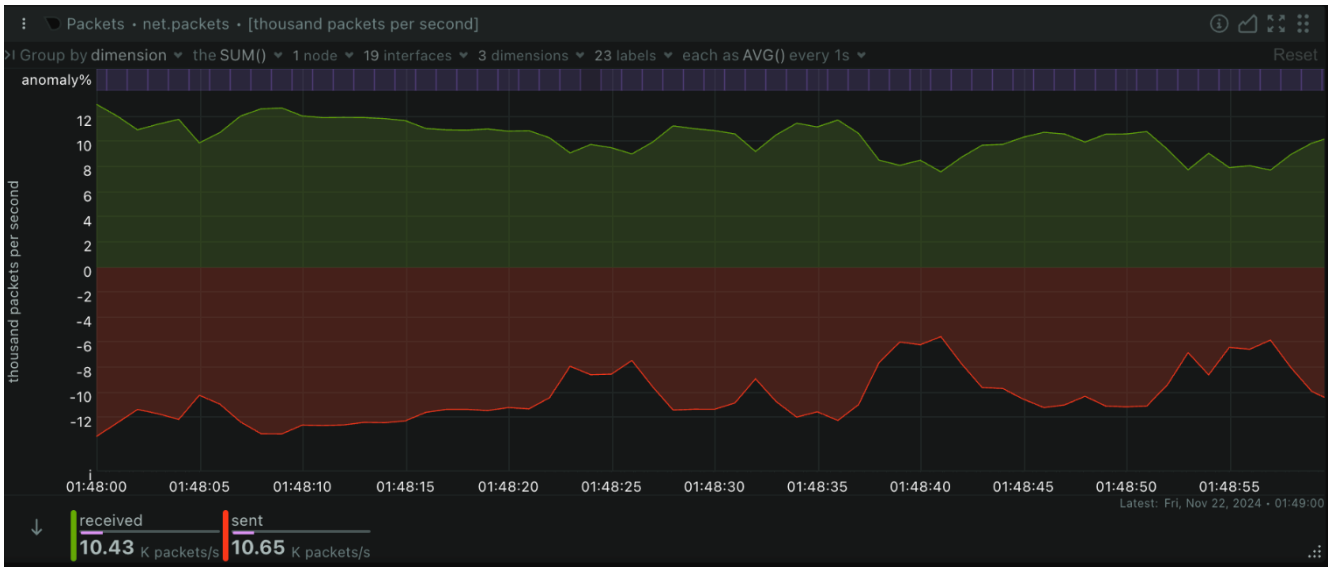

Physical Network Interfaces Aggregated Bandwidth • system.net • [megabits per second]
Group by dimension the SUM() 1 node 1 system 2 dimensions 2 labels each as AVG() every 1s
Reset
anomaly%
20
15
10
5
0
-5
megabits per second
01:44:00
01:44:10
01:44:20
01:44:30
01:44:40
01:44:50
01:45:00
01:45:10
01:45:20
01:45:30
01:45:40
01:45:50
Latest: Fri, Nov 22, 2024 • 01:46:00
received
0.05 Mbit/s
sent
0.05 Mbit/s

Рисунок 18 – Результаты замеров нагрузки при 1000 параллельных потоках

В рамках рассматриваемой концепции предполагается, что узлы блокчейн-сети могут располагаться на сетевом оборудовании непосредственно [32] (Рисунок 20). Узел блокчейна функционирует на уровне приложения (уровень 7 модели OSI). Для его полноценной работы требуется надежная поддержка на сетевом уровне (уровень 3) и транспортном уровне (уровень 4), используя протоколы IP и TCP/UDP соответственно. То есть, интеграция узла блокчейна осуществляется на сетевом уровне и выше, с особым вниманием к инфраструктуре уровней 3 и 4 для обеспечения стабильного и эффективного соединения.


System RAM • system.ram • [gibibytes]
Group by dimension the SUM() 1 node 1 system 4 dimensions 2 labels each as AVG() every 1s
Reset
anomaly%
3.9063
3.418
2.9297
2.4414
1.9531
1.4648
0.9766
0.4883
gibibytes
01:48:00 01:48:05 01:48:10 01:48:15 01:48:20 01:48:25 01:48:30 01:48:35 01:48:40 01:48:45 01:48:50 01:48:55
Latest: Fri, Nov 22, 2024 • 01:49:00
free 0.2524 GiB
used 3.8807 GiB
cached 3.4519 GiB
buffers 0.1664 GiB


Packets • net.packets • [thousand packets per second]
Group by dimension the SUM() 1 node 19 interfaces 3 dimensions 23 labels each as AVG() every 1s
Reset
anomaly%
12
10
8
6
4
2
0
-2
-4
-6
-8
-10
-12
thousand packets per second
01:48:00 01:48:05 01:48:10 01:48:15 01:48:20 01:48:25 01:48:30 01:48:35 01:48:40 01:48:45 01:48:50 01:48:55
Latest: Fri, Nov 22, 2024 • 01:49:00
received 10.43 K packets/s
sent 10.65 K packets/s

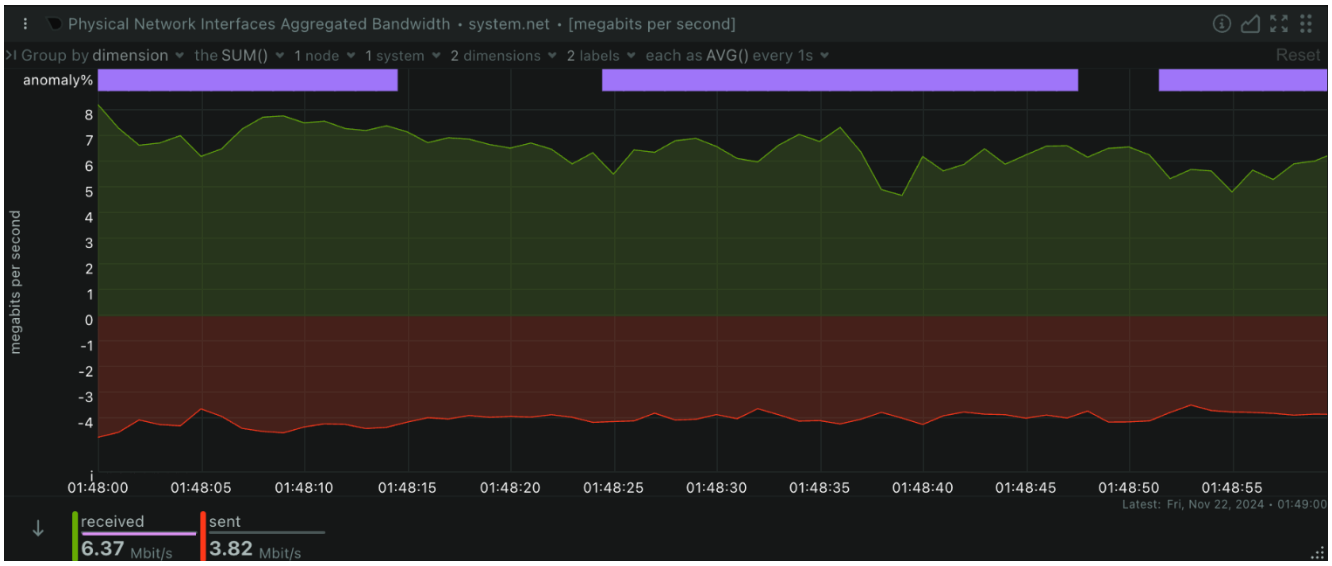

Physical Network Interfaces Aggregated Bandwidth • system.net • [megabits per second]
Group by dimension the SUM() 1 node 1 system 2 dimensions 2 labels each as AVG() every 1s
Reset
anomaly%
8
7
6
5
4
3
2
1
0
-1
-2
-3
-4
megabits per second
01:48:00 01:48:05 01:48:10 01:48:15 01:48:20 01:48:25 01:48:30 01:48:35 01:48:40 01:48:45 01:48:50 01:48:55
Latest: Fri, Nov 22, 2024 • 01:49:00
received 6.37 Mbit/s
sent 3.82 Mbit/s

Рисунок 19 – Результаты замеров нагрузки при 1500 параллельных потоках

Для оценки возможности использования сетевого оборудования для текущих реализаций блокчейн-клиентов рассмотрим модель маршрутизатора Cisco ISR 4451-X:

– CPU: Многоядерный процессор с частотой ~2.5 ГГц,

– оперативная память: 8–16 ГБ,

– пропускная способность: до 10 Гбит/с с сервисами,

– средняя загрузка CPU:

а) при средней нагрузке: 30–50%,

б) при пиковых нагрузках: 70–90%.

Назначение: Предоставление высокопроизводительной маршрутизации с поддержкой множества сервисов (VPN, QoS, firewall).

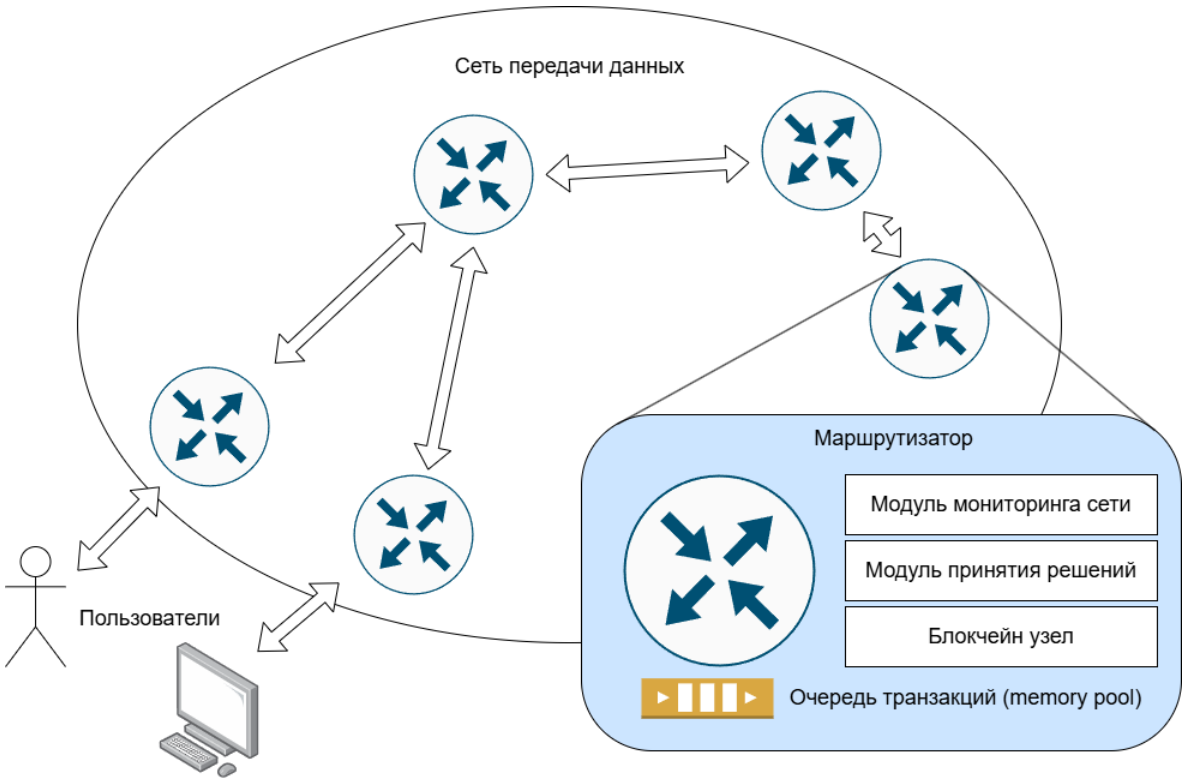


Рисунок 20 - Архитектура сети связи с интегрированным слоем блокчейн и модулем адаптивного алгоритма

Для оценки возможности использования сетевого оборудования для текущих реализаций блокчейн-клиентов рассмотрим модель коммутатора Cisco Catalyst 9500 Series:

– CPU: специализированные ASIC для обработки трафика,

– оперативная память: 16 ГБ,

– пропускная способность: до 480 Гбит/с,

– средняя загрузка CPU:

а) при средней нагрузке: 5–15% (CPU задействован в основном для управленческих задач),

б) при пиковых нагрузках: 20–30%.

Назначение: высокоскоростная коммутация с возможностью маршрутизации для ядра сети.

Рассмотрим официальные требования наиболее популярного блокчейн-узла с открытым исходным кодом Ethereum к аппаратному обеспечению для его запуска (имеется в виду запуск узла главной сети, предполагающий синхронизацию полного объема данных и дальнейшую обработку транзакций сети, и создание блоков):

- CPU: 4 ядра и выше,
- оперативная память: 8-16 ГБ,
- жесткий диск: 1 ТБ,
- сетевое подключение: 25 Мбит/с,
- тип соединения: проводной.

Следует учитывать, что блокчейн-сеть Ethereum является одной из наиболее нагруженных блокчейн-сетей с ежедневными крайне высокими показателями количества транзакций за день [11]. По этой причине рассмотрение требований к узлу главной сети Ethereum не является полностью корректным, так как в сетевой архитектуры использовались бы легкие клиенты по передаче данных в главную сеть, а не их обработки. Но даже при таких конфигурационных требованиях развертывание узла возможно на сетевом оборудовании.

Также для специфичных кейсов следует учитывать возможность использования специализированного оборудования для размещения узлов блокчейн-сети (Рисунок 21).

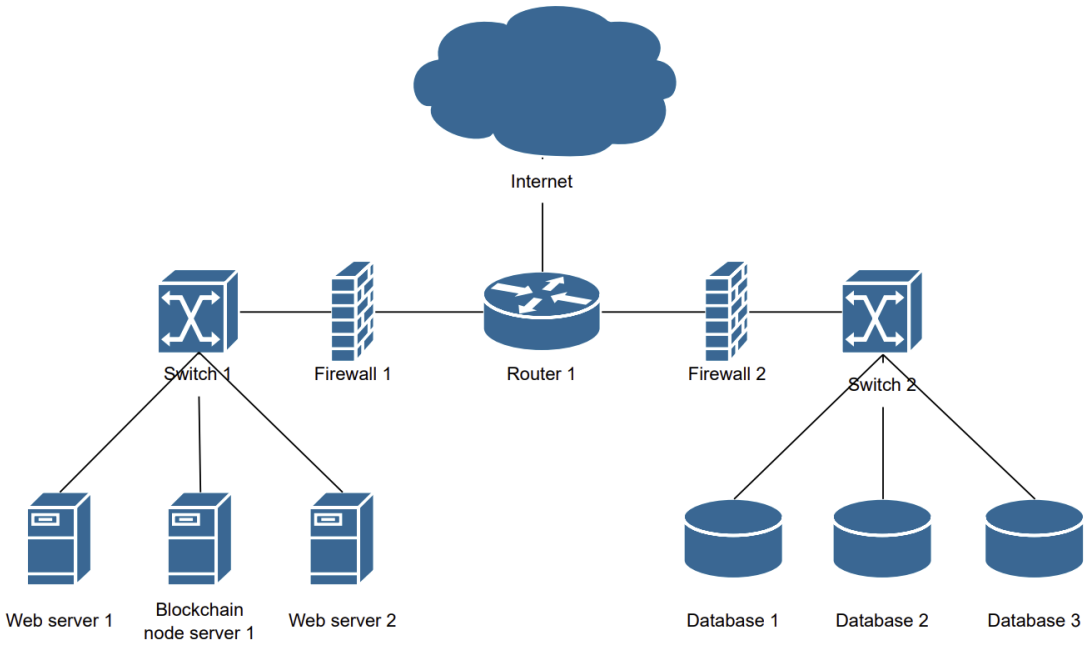


Рисунок 21 – Альтернативная схема подключения блокчейна к сети связи

Для реализации архитектуры, приведенной на Рисунке 20 некоторые производители оборудования предлагают специализированные решения. Например, сервера для размещения узла блокчейна Ethereum предлагается сервер Dell PowerEdge R750 с характеристиками:

– CPU: два процессора Intel Xeon Scalable (3-го поколения), каждый с 16–28 ядрами, частота 2.4–3.6 ГГц,

– оперативная память: 64–512 ГБ DDR4,

– хранилище: NVMe SSD объемом 2–8 ТБ для быстрого доступа к данным блокчейна,

– сетевая подсистема: двухпортовые 10–25 Гбит/с сетевые адаптеры.

При этом ожидается, что средняя загрузка ресурсов при работе узла Ethereum составит:

– CPU: 40–60% при обработке транзакций и поддержании сети,

– RAM: 16–32 ГБ используются для кэширования и быстрых операций.

Сетевая загрузка составит:

– средняя: 100–200 Мбит/с,

– пиковая: до 500 Мбит/с при интенсивной синхронизации или обработке крупных блоков.

Еще одной конфигурацией оборудования являются сервера для размещения узла приватного блокчейна (testnet/devnet), если речь идет о корпоративном контуре или каких-то других закрытых решениях. Характеристики оборудования Dell PowerEdge T340 выглядят следующим образом:

– CPU: Один процессор Intel Xeon E-2124, 4 ядра, частота 3.3 ГГц,

– оперативная память: 16–32 ГБ DDR4,

– хранилище: SSD объемом 1–2 ТБ для хранения данных блокчейна,

– сетевая подсистема: Два гигабитных Ethernet-порта,

Средняя загрузка ресурсов при работе узла приватного блокчейн-клиента на основе исходного кода Ethereum составит:

– CPU: 20–40% при обработке транзакций и поддержании сети,

– RAM: 8–16 ГБ используются для кэширования и операций.

Сетевая загрузка составит:

– средняя: 10–50 Мбит/с,

– пиковая: до 100 Мбит/с при интенсивной синхронизации или тестировании.

Для приватного блокчейна или тестовой сети (testnet/devnet) не требуется столь мощное оборудование, как для публичного узла в основной сети Ethereum. Сервер начального или среднего уровня, такой как Dell PowerEdge T340, предоставляет достаточную производительность для разработки, тестирования и внутренних внедрений блокчейна.

2.4 Исследование применимости блокчейн-сетей для записи данных в контексте ITS

Для оценки ряда параметров, а также жизнеспособности концепции интеграции технологии блокчейн в современные сети связи и IT ландшафт

проводился ряд экспериментов в контексте интеллектуальной транспортной сети, так как такая сеть по своей архитектуре и структуре отражает все основные важные для измерения особенности. Например, аппаратное обеспечение участков сети, различающийся уровень сетевой активности передачи данных в контексте центра крупного города и отдаленной трассы в северных регионах. По этой причине многие эксперименты и апробации проводились именно в условиях эмуляции ITS [33-35].

Главной целью проводимого исследования является оценка работоспособности узла, скорости его работы в рамках оказываемой нагрузки со стороны элементов инфраструктуры концепции интеллектуальной транспортной сети. На сегодняшний день в различных исследованиях уже представлены сценарии передачи информации от автомобильных и дорожных датчиков, но без взаимодействия и интеграции блокчейн-сети. С точки зрения исследования работоспособности классического блокчейн-узла наиболее показательным является тестирование, приближенное к нагрузочному тестированию узла, при котором количество запросов к нему достаточно велико за некоторый ограниченный период времени.

В качестве опорных данных были взяты исследования [36] по оценке интенсивности трафика на пункте взимания платы при съезде на платную дорогу по адресу Россия, Санкт-Петербург, Богатырский пр./Планерная ул., съезд. Данные исследования позволили выявить, какое количество машин в среднем в неделю проезжает мимо пункта взимания платы в каждый из часов в сутках. Таким образом, в Таблице 3 представлены данные количества машин в каждый из часов, а также процент машин, проезжающих данный участок в каждый из часов суток в процентах.

В случае проводимого исследования предполагается, что каждое транспортное средство при проезде через пункт взимания платы отправляет одну транзакцию на узел блокчейн-сети [33]. Отправляемая транзакция может содержать различную информацию, начиная от факта оплаты проезда, факта проезда до состояния дорожного полотна. Задачей данного исследования является

именно оценка работоспособности узла и возможности обработать все поступающие транзакции без потерь данных.

Таблица 3 - Соотношение времени суток и количества проезжающих транспортных средств

Часы	**Количество OBU**	**OBU %**	**Часы**	**Количество OBU**	**OBU %**
0	600	1,65%	12	1920	5,26%
1	350	0,96%	13	2000	5,48%
2	250	0,69%	14	2200	6,03%
3	200	0,55%	15	2300	6,31%
4	150	0,41%	16	2500	6,85%
5	100	0,27%	17	2900	7,95%
6	125	0,34%	18	3100	8,50%
7	225	0,62%	19	3000	8,23%
8	1000	2,74%	20	2850	7,81%
9	2100	5,76%	21	1850	5,07%
10	2200	6,03%	22	1500	4,11%
11	2050	5,62%	23	1000	2,74%
Общее количество OBU: 36470					

На основе табличных данных был получен график, демонстрирующий пиковые значения, а также общий профиль интенсивности движения после корреляции значений в 12 раз, он представлен на Рисунке 22.

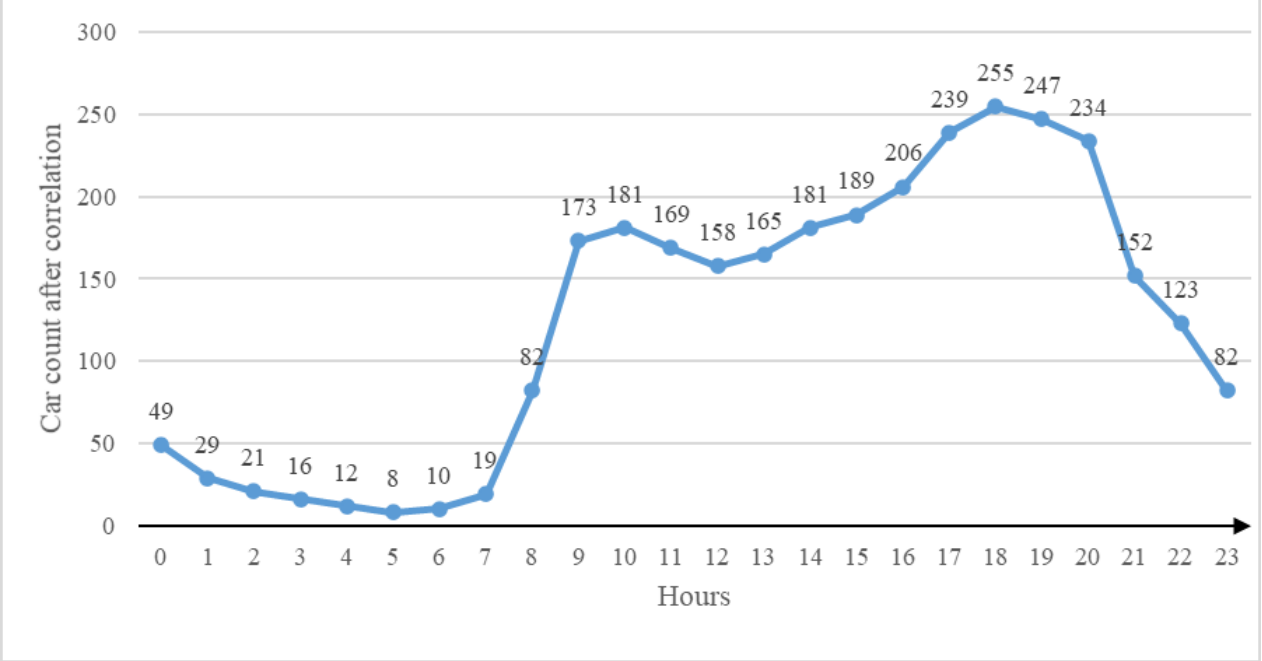


Рисунок 22 – Расчет количества автомобилей в соотношение с часами после корреляции

Таким образом, каждый автомобиль совершает отправку транзакции соответственно полученным данным. Для симуляции данных транзакций авторы работы разработали программный скрипт на программном языке Питон. Это позволяет имитировать нагрузку на сеть согласно данным исследования интенсивности дорожно-транспортного движения на рассматриваемом участке. Процесс корреляции позволяет ускорить проводимое исследования и оценить с применением нагрузочного тестирования, справляется ли узел блокчейн-сети с повышенной нагрузкой.

Проведенные расчеты позволили ускорить процесс симуляции. При проведении исследования 1 час реального времени становится 5 минутами исследования, что также влияет на количество автомобилей, что показано на Рисунке 22.

С точки зрения работы скрипта подобная симуляция также требует многопоточности, чтобы корректно отправлять транзакции согласно полученному распределению нагрузки. Во время отправки транзакции скрипт взаимодействует с API библиотеки Web3, которая позволяет наладить взаимодействие с узлом частного экземпляра рассматриваемых блокчейн-сетей на разработанном модельном стенде.

Для разработки модельного стенда и проведения дальнейшего исследования были использовали две блокчейн-платформы – Ethereum, работающий на базе алгоритма PoW (т.к. поддерживается две реализации – на PoW и на PoS [37]) и Waves, работающий на базе алгоритма LPoS. В связи с особенностями работы частных экземпляров сети рассматриваемых платформ было организовали два модельных стенда. Выбор данных блокчейн-сетей обусловлен рядом причин. Данные блокчейны позволяют организовать частные сети на базе открытого исходного кода, позволяют запускать смарт-контракты, а также строить гибридные системы и создавать решения с интеграцией сторонних технологий.

Ключевым отличием данных блокчейн-платформ являются лежащие в основе алгоритмы консенсуса, что сказывается на производительности блокчейн-сети, ее защищенности, а также аппаратных требованиях к узлу организуемой сети.

Waves Enterprise - это блокчейн-платформа, которая позволяет создавать различные блокчейн-решения на базе смарт-контрактов. Распределенная инфраструктура сети позволяет производить большое количество транзакций за небольшие промежутки времени. Одним из преимуществ платформы является возможность создания собственной частной сети на основе открытого jar файла, содержащего все необходимые файлы и данные конфигурации для настройки собственного узла закрытой блокчейн-сети.

Алгоритм консенсуса блокчейн-сети Waves - Leased Proof of Stake (LPoS). Данный алгоритм консенсуса не требует произведения энергоемких вычислений, в отличие от алгоритма консенсуса Proof of Work. Главным параметром, определяющим величину вероятности генерации нового блока, является количество валюты платформы на счету узла. Таким образом, количество валюты является аналогом количества лотерейных билетов. Чем их больше, тем выше вероятность выиграть. Аналогичный принцип распространяется и в сети Waves.

Схема модельного стенда с использованием узлов блокчейн-сети Waves представлена на Рисунке 23.

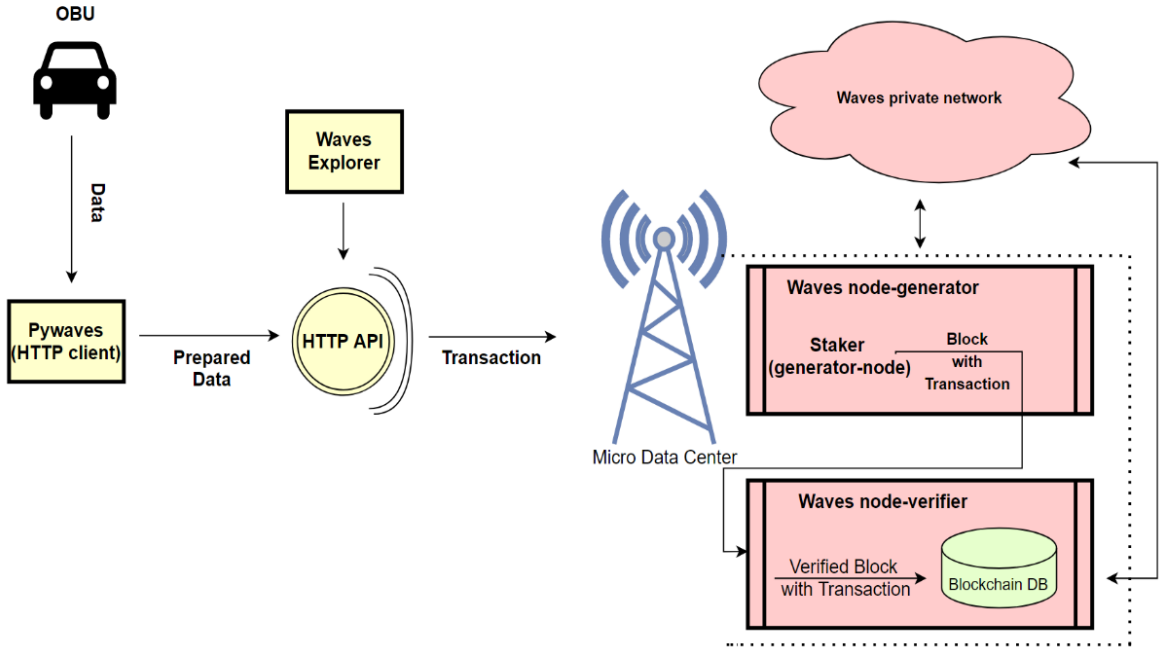


Рисунок 23 – Модельный стенд для проведения эксперимента на базе открытого блокчейн-клиента LPoS Waves

Модельный стенд Waves состоит из следующих элементов:

– OBU - пользовательское устройство, инициирующее отправление транзакции на основе текущих данных,

– pywaves (HTTP client) - фреймворк, основной задачей которого является взаимодействие с программным интерфейсом HTTP узла Waves,

– HTTP API - программный интерфейс узла Waves, который, предоставляет возможность взаимодействия с узлом Waves,

– узел Waves – стейкер (узел-генератор) - основной программный узел сети блокчейн Waves. Запускает процесс сборки блока и обработки транзакций,

– узел Waves - верификатор - дополнительный программный узел сети блокчейн Waves основной задачей которого является подтверждение транзакций и блоков,

– Waves Explorer, веб интерфейс, позволяющий проверять текущее состояние узлов и деплоить смарт контракты,

– сеть блокчейн Waves - в тестовых целях была создана приватная частная сеть.

OBU инициирует отправку транзакции. Передает информацию о своем текущем состоянии Pywaves. Pywaves, используя полученные данные, создает запрос о совершении транзакции в формате читаемым HTTP интерфейсом и отправляет его на программный интерфейс HTTP. HTTP API обрабатывает полученный запрос и фиксирует информацию о попытке совершить транзакцию в основном узле (узле-генераторе). Узел-генератор (стейкер), получая транзакцию, отправляет ее на верификацию в собственный кворум - собрание всех узлов в сети и ожидает верификации от узла-верификатора. Узел-генератор, собрав определенное количество верифицированных транзакций или подождав определенный промежуток времени, инициирует формирование блока, который объединит все накопленные транзакции. Информация об успешно сформированном блоке возвращается по аналогичному пути обратно клиенту.

В рамках проводимого исследования было рассмотрено 2 сценария – в первом случае транзакции отправлялись на запись в блокчейн сеть, а во втором – запрашивались из блокчейн-сети, что позволяло оценить время ответа от блокчейн-узла. Необходимо отметить, что запрос на запись в блокчейн-сеть и на чтение из блокчейн сети производилось одновременно. Запрос на запись в блокчейн-сеть происходил согласно графику, приведенному на Рисунке 24. Запрос на чтение из блокчейн-сети, в свою очередь, происходил каждые 1 секунду.

Таким образом, на Рисунке 24 (A) демонстрируются результаты записи данных в блокчейн-сеть Ethereum и Рисунке 24 (B) результаты записи данных блокчейн-сеть Waves. Необходимо отметить, что все эксперименты производятся с ожиданием подтверждения записи транзакции в сеть. То есть для записи следующей транзакции необходимо получить ответ от блкочейн-сети о том, что транзакция успешно обработана и записана. Необходимость подтверждения записи обусловлена чувствительностью системы к потерям, так как потеря данных от OBU, в случае фиксации данных об оплате проезда, чревата некорректными сведениями о транспортном средстве.

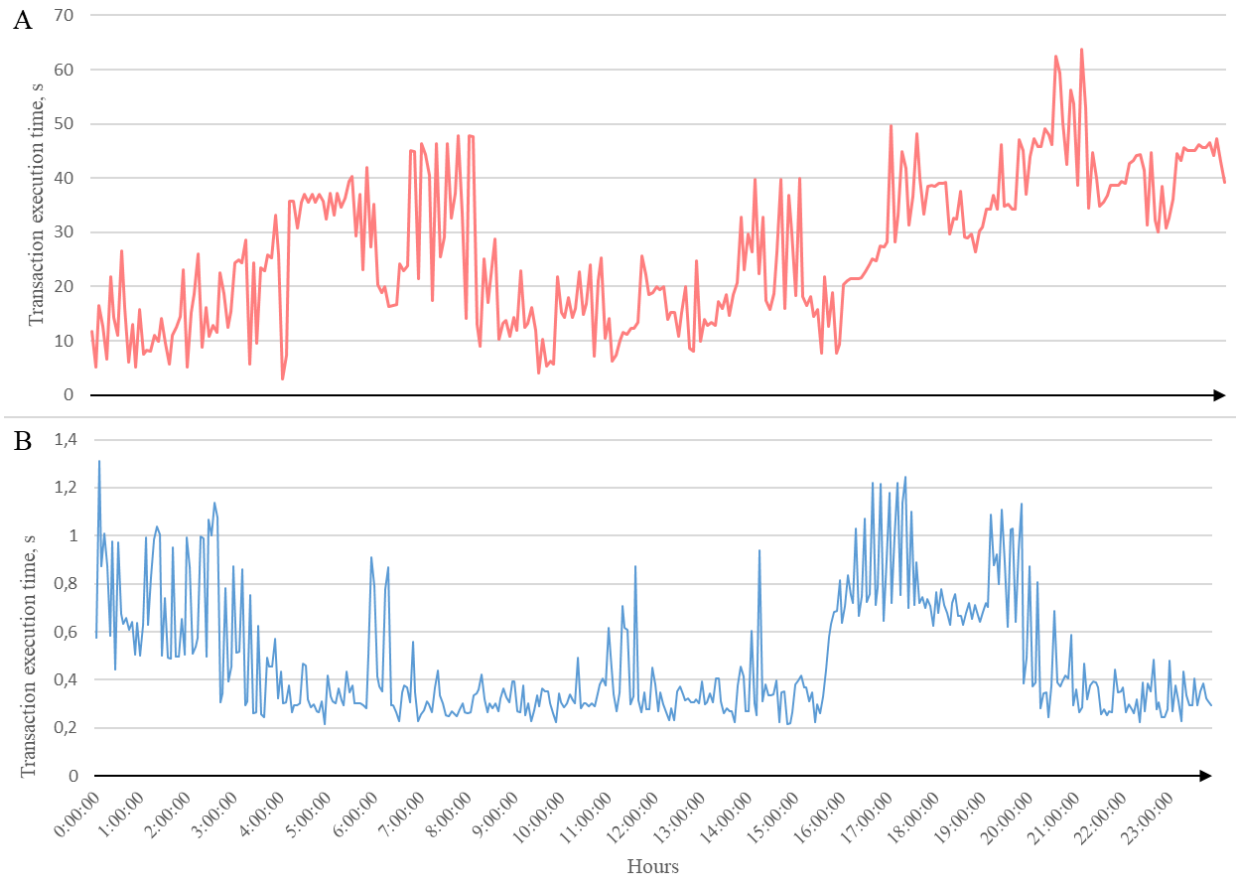


Рисунок 24 - Время обработки транзакции (A – в случае сети Ethereum, B – в случае сети Waves)

Как можно заметить, на Рисунке 24 (A) явно заметно длительное время обработки транзакции, а также явные потери. При проведении эксперимента в блокчейн-сеть было записано 3000 транзакций, из которых успешно обработано было лишь 2554 транзакции, что означает, что 446 транзакций были отброшены и не зарегистрированы в блокчейн-сети. Потери составляют 14,9%.

В случае же записи данных в блокчейн сеть Waves потери при записи транзакций в блокчейн-сеть отсутствуют. На Рисунке 24 (B) наглядно демонстрируется, что скорость записи транзакций в 20 – 40 раз выше, чем в блокчейн-сети Ethereum. Необходимо также учитывать, что приватные сети были настроены таким образом, чтобы максимально быстро обрабатывать входящие транзакции при сравнительно невысоких аппаратных характеристиках. Таким образом, при построении архитектуры сети SD-IoV-Blockchain [38 - 40] реализация

одного узла блокчейн-сети возможна только в случае использования алгоритма консенсуса LPoS.

На Рисунке 25 демонстрируется результат обработки запроса на чтение из блокчейн-сети. Аналогичным образом можно наблюдать явное преимущество у блокчейн-сети Waves с алгоритмом LPoS, где скорость обработки запроса на чтение в среднем в 10 раз выше, чем у блокчейн-сети Ethereum PoW.

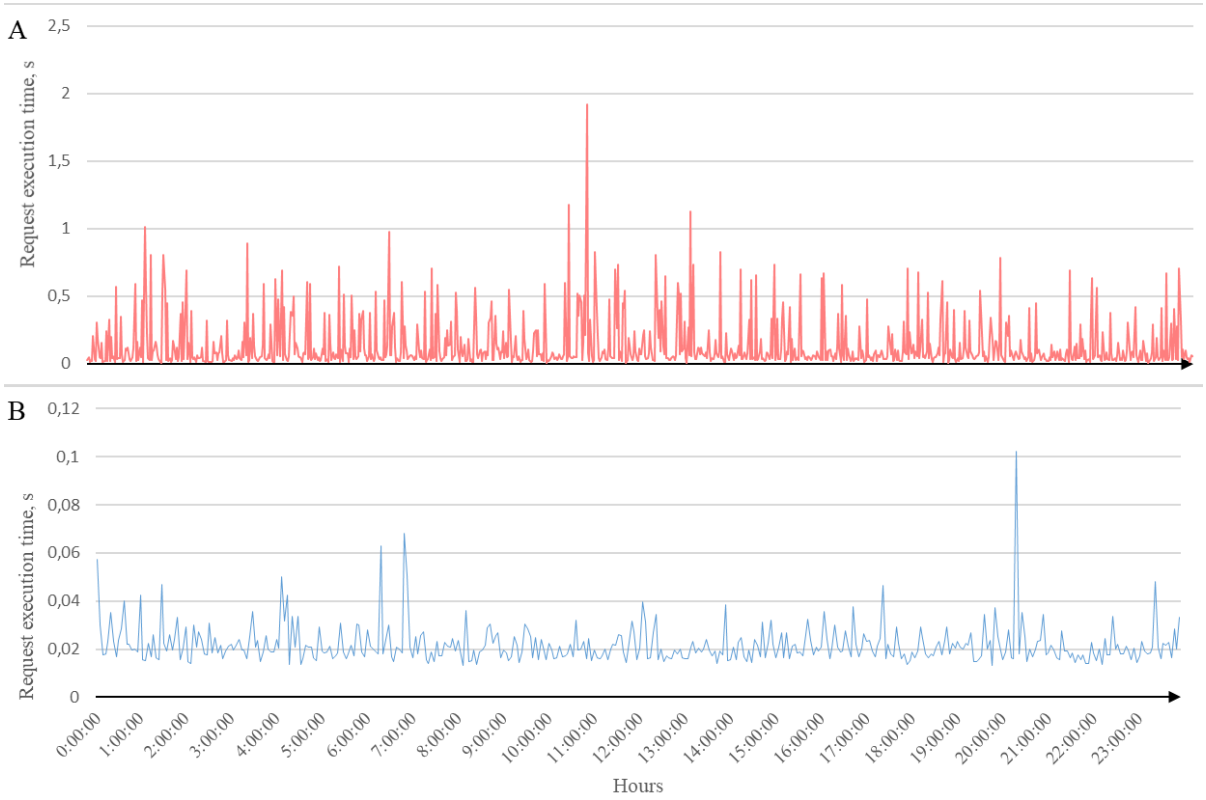


Рисунок 25 – График обработки на чтение данных (A – для реализации на базе Ethereum, B – для реализации на базе Waves)

На основе полученных результатов исследования можно сделать вывод о наибольшей пригодности алгоритмов консенсуса семейства Proof of Stake и блокчейн-сети Waves по сравнению с классической блокчейн-сетью Ethereum в условиях ограниченных аппаратных характеристик.

Проведенное исследование демонстрирует возможность интеграции и применения технологии блокчейн в концепции ITS и пригодности для использования в целом [41, 42]. А также ярко демонстрирует необходимость балансировки в зависимости от текущих сетевых условий, так как на приведенных

графиках видны периоды эффективности каждого из алгоритмов. Реализация адаптивного алгоритма для информационных блокчейн-потоков позволила бы эффективно записывать информацию, а также балансировать нагрузку на аппаратное обеспечение.

2.5 Сетевые граничные значения переключения алгоритмов консенсуса

Проведенные моделирования и значения, полученные эмпирическим путем в рамках исследования влияния на аппаратные компоненты, а также сетевые характеристики [4, 43], рассмотренные в предыдущих пунктах, позволили собрать информацию о допустимых граничных значениях, при которых следует смена алгоритма консенсуса и при которых алгоритмы работают наиболее эффективно. Значения представлены в Таблице 4.

Таблица 4 – Диапазоны сетевых характеристик канала связи эффективности работы алгоритмов консенсуса, выявленные эмпирическим путем

Алгоритм	PoW	PoS	DPoS	PBFT	PoR	PoA	PoET	FBA
Пропускная способность (B), Мбит/с	10–100	5–50	5–50	10–100	10–100	10–100	10–100	10–100
Задержка (L), мс	100–200	50–100	50–150	50–100	50–100	50–120	100–200	50–150
Джиттер (J), мс	10–30	10–20	10–30	5–15	5–15	10–30	20–50	10–20
Размер блока (St), КБ	500–2000	250–500	250–1000	100–1000	100–1000	500–2000	500–1000	100–1000
Частота транзакций (ft), транзакций/с	1–10	1–10	10–50	10–50	10–50	5–30	5–30	5–20
Частота генерации 1 блока (Tblock), с	10–30	5–15	5–10	3–10	3–15	10–20	10–15	5–10
Заполненность memory pool (Mpool), %	30–50	30–60	20–40	20–50	20–40	30–50	30–50	20–40
Загрузка CPU, %	60–90	40–70	30–60	50–70	50–80	30–50	40–60	40–70
Использование памяти, МБ	1024–2048	512–1024	256–512	512–1024	512–2048	512–1024	256–512	512–1024
Использование диска, МБ/с	20–50	10–30	5–20	10–30	10–40	10–20	5–20	5–15

Загрузка CPU, % – показывает, сколько вычислительных ресурсов требуется для поддержания работы консенсуса. Например, PoW требует высокой загрузки из-за вычислений хэшей, а PoS и DPoS имеют более низкие требования.

Использование памяти, МБ – характеризует объём оперативной памяти, необходимый для обработки транзакций, поддержки пула необработанных транзакций и генерации блоков. Например, DPoS требует меньше памяти, чем PoW, за счёт меньшей сложности вычислений.

Использование диска, МБ/с – отражает объём данных, записываемых или считываемых на диске. Например, PoW и PoR имеют повышенные значения, так как они часто генерируют крупные блоки и интенсивно используют хранилище.

Таблица характеристик алгоритмов консенсуса сформирована на основе совокупного анализа теоретических моделей, экспериментальных данных и результатов имитационного моделирования. В её основе лежат сведения из существующей научной литературы, посвящённой сравнению различных алгоритмов, а также данные мониторинга реальных блокчейн-сетей, таких как Bitcoin, Ethereum, Hyperledger Fabric и EOS. Экспериментальные измерения проводились в контролируемых условиях тестовых стендов, где оценивалась загрузка аппаратных ресурсов, пропускная способность, задержка и другие параметры при различных уровнях сетевой нагрузки.

Имитационное моделирование дополнило результаты, позволив протестировать алгоритмы в широком диапазоне сценариев, включая высокую интенсивность транзакций и переполненность пула неподтвержденных транзакций. Для новых и менее изученных алгоритмов, таких как PoET, параметры были уточнены с использованием экстраполяции данных и теоретического анализа архитектурных особенностей.

2.6 Выводы

1) Произведена категоризация алгоритмов консенсуса по семействам, с выделением наиболее подходящих реализаций, которые будут применены при разработке адаптивного алгоритма.

2) Проведен анализ и оценка влияния блокчейн-трафика на канал связи и пропускную способность.

3) Представлена Таблица 4, демонстрирующая диапазоны сетевых характеристик канала связи эффективности работы алгоритмов консенсуса, выявленные эмпирическим путем и с применением имитационного моделирования на основе представленных исследований. Число сетевых характеристик для эффективного выбора блокчейн-систем в сетях связи не превышает десяти штук с учетом реальных диапазонов их пороговых значений.

ГЛАВА 3 МОДЕЛЬ МОДУЛЯ ПРИНЯТИЯ РЕШЕНИЯ ПО ВЫБОРУ БЛОКЧЕЙН-СИСТЕМ

Ключевой задачей модуля принятия решения по выбору блокчейн-систем является подбор наиболее эффективного алгоритма консенсуса в момент времени с учетом сетевых и аппаратных характеристик, а также скорость распространения информации о смене консенсуса, чтобы минимизировать потерю блоков и транзакций.

3.1 Принцип работы модуля принятия решения

Согласно проведенному анализу текущих исследований на стыке технологии блокчейн и сетей связи, наиболее актуальным направлением является модернизация и доработка алгоритмов консенсуса для получения наиболее универсального решения [44]. Однако для решения этой задачи может быть использован альтернативный подход с адаптивным алгоритмом выбора алгоритма консенсуса в некоторый момент времени, с учетом состояния компонентов аппаратного обеспечения, на котором разворачивается узел блокчейн-сети, а также состояния канала связи.

Общая блок-схема принципа работы адаптивного алгоритма представлена на Рисунке 26.

В предлагаемой концепции каждый узел блокчейн-сети обладает собственным модулем оценки сетевых характеристик в момент времени T, для принятия решения о смене алгоритма консенсуса и корректировки параметров формирования новых блоков данных. Далее, информация об обновлении правил создания и валидации блоков распространяется от узла к другим узлам с целью смены общего алгоритма консенсуса в сети. Этот процесс делится на несколько этапов передачи информации, что напрямую зависит от количества связей узла.

На Рисунке 27 представлена общая схема сценария изменения алгоритма консенсуса в связи с изменением сетевых параметров. Как можно заметить, вторым этапом после обнаружения необходимости узлом сменить алгоритм консенсуса является передача сведений о смене. Первыми сообщение о смене консенсуса получат узлы, адреса которых знает исходный узел, с которым производится синхронизация данных. После получения информации эти узлы передают запрос на смену консенсуса далее.

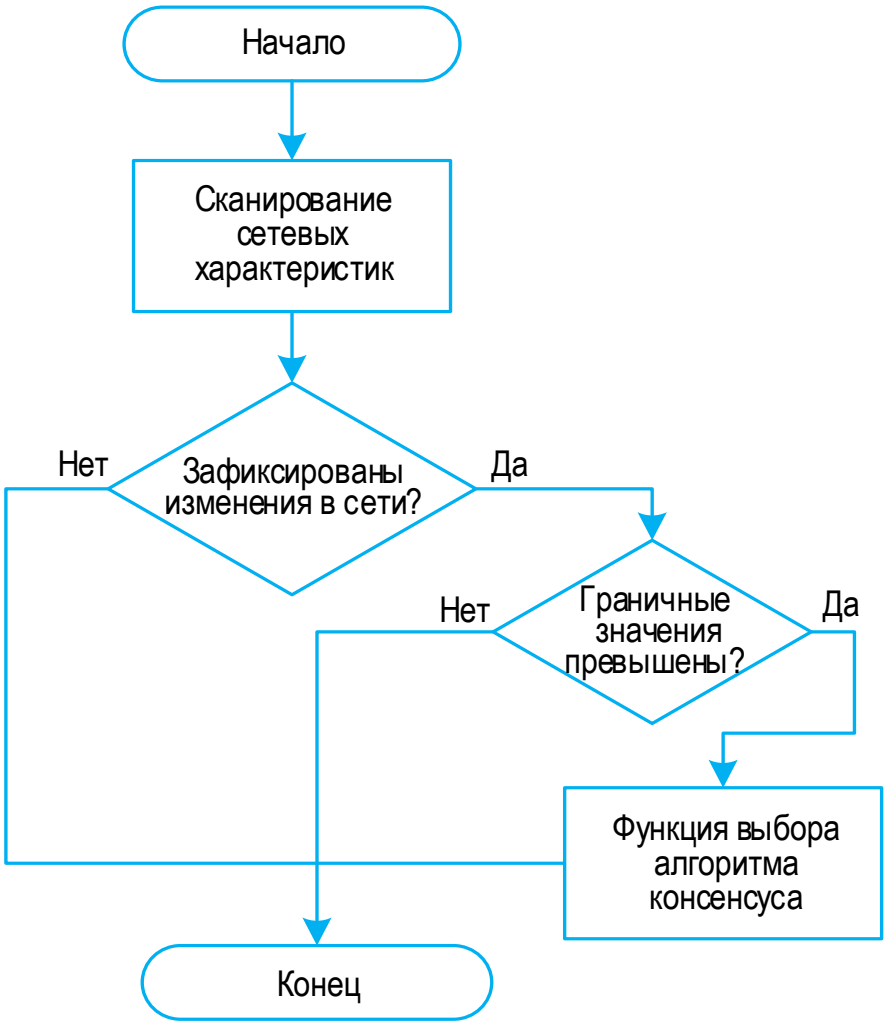


Рисунок 26 - Блок-схема адаптивного алгоритма выбора консенсуса на основе данных изменения сетевых характеристик

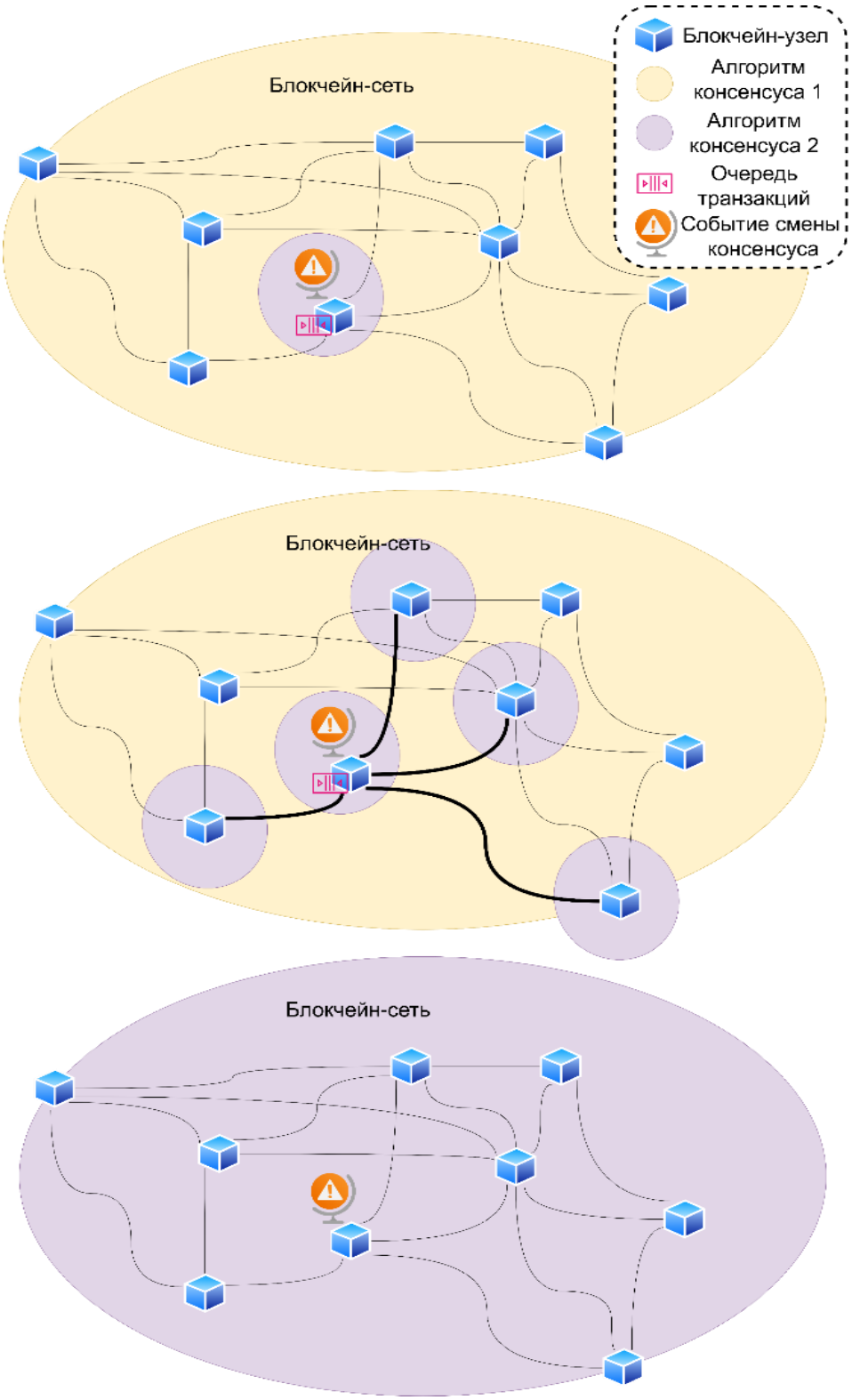


Рисунок 27 - Схема распространения информации о смене алгоритма консенсуса в сети устройств

Однако эффективность данного подхода ограничена количеством устройств и их связей, количеством «уровней» передачи информации – чем меньше связей у узла сети, тем больше будет этапов передачи информации до границы сети

(аналогично принципам систем с графами). Ограничение появляется в связи с функционированием сети во время смены алгоритма консенсуса, так как каждый узел продолжает работу по формированию блоков транзакций. И распространение блока транзакций, сформированного на основе правил алгоритма консенсуса 1, не может быть провалидировано и добавлено в сеть узлом, чей активный алгоритм консенсуса при валидации отличается, что влечет за собой потерю блоков транзакций, и, как следствие, задержку в обработке транзакций.

Согласно Рисунку 20, компонент, устанавливаемый на оборудование, включает в себя модуль мониторинга и модуль принятия решения в предлагаемой в работе концепции.

Алгоритм работы модуля мониторинга представлен на Рисунке 28, а его программный код представлен в Приложении А.

Модуль мониторинга отвечает за контроль состояния выделенных характеристик сети, на основе значений которых далее принимается решение о необходимости смены алгоритма консенсуса в сети. Модуль мониторинга сетевого трафика представляет собой систему, предназначенную для анализа и оценки сетевых пакетов в реальном времени с целью оптимизации производительности сетевой инфраструктуры. Он использует библиотеку Scapy для захвата пакетов и предоставляет ключевые метрики, такие как пропускная способность, задержка и джиттер, которые помогают оценить качество сетевого соединения.

Процесс работы модуля мониторинга включает в себя несколько этапов. На первом этапе осуществляется сбор данных, где каждый захваченный пакет анализируется, фиксируется его размер и временная метка. Это создает обширный набор данных для дальнейшего анализа сетевой активности.

На втором этапе происходит анализ задержек между последовательными пакетами, что позволяет выявить потенциальные проблемы с производительностью сети. Затем вычисляются метрики, такие как пропускная способность, средний размер пакета, а также средняя задержка и джиттер, что позволяет получить полное представление о состоянии сети. Собранные данные записываются в лог-файл для модуля принятия решений. На Рисунке 29

представлен алгоритм работы модуля принятия решения, код его представлен в Приложении А.

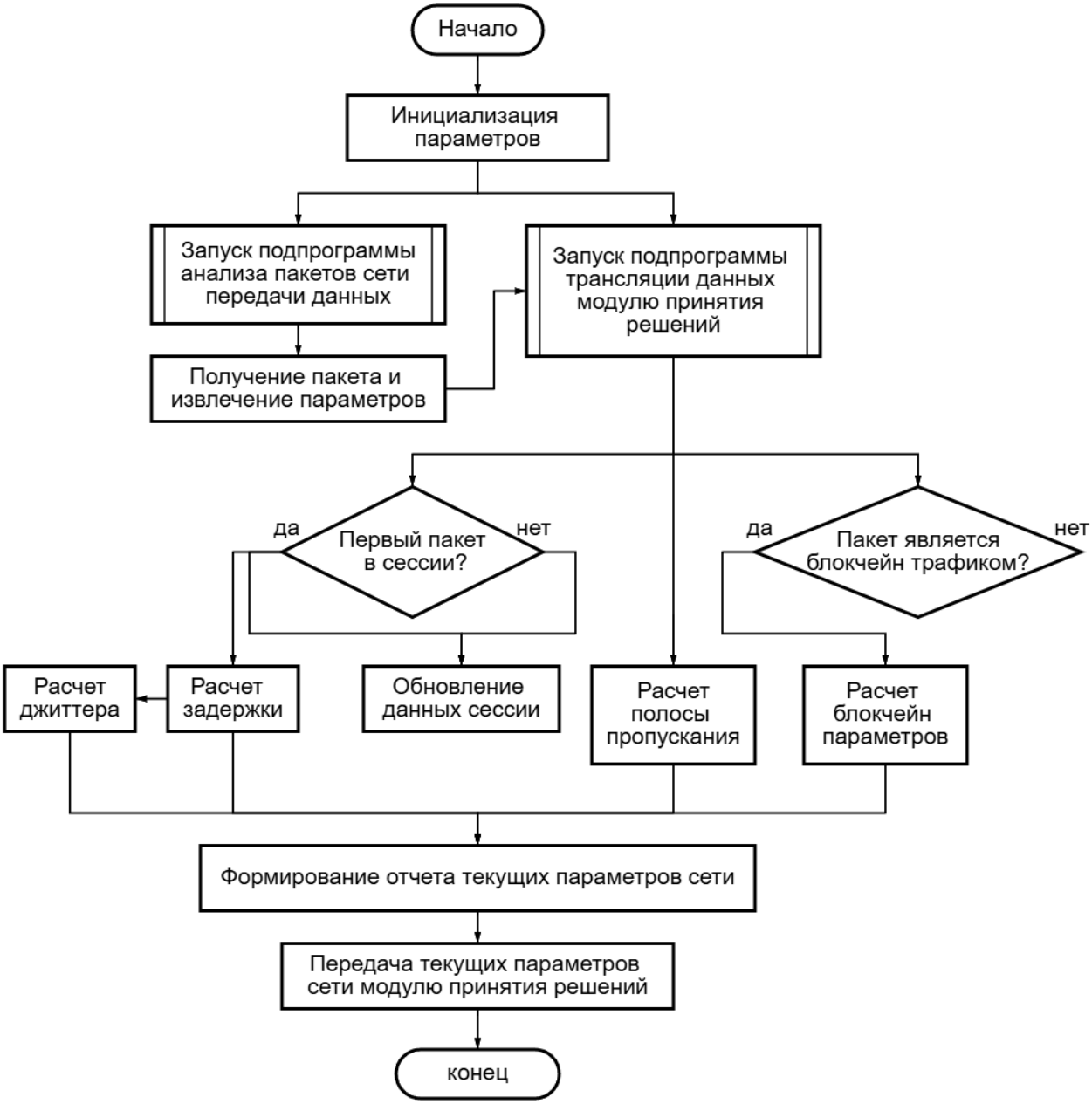


Рисунок 28 – Алгоритм работы модуля мониторинга

Модуль принятия решения адаптивного выбора алгоритма консенсуса в блокчейн-сети предназначен для динамического управления выбором алгоритма консенсуса в зависимости от текущих условий работы сети. Основная идея заключается в том, чтобы оптимизировать производительность сети, автоматически подбирая наиболее подходящий алгоритм консенсуса в ответ на

изменения в метриках сети. Алгоритм использует заранее определённую таблицу, в которой указаны условия для каждого алгоритма консенсуса. Эта таблица помогает определить, какой алгоритм будет наиболее эффективным в текущих условиях.

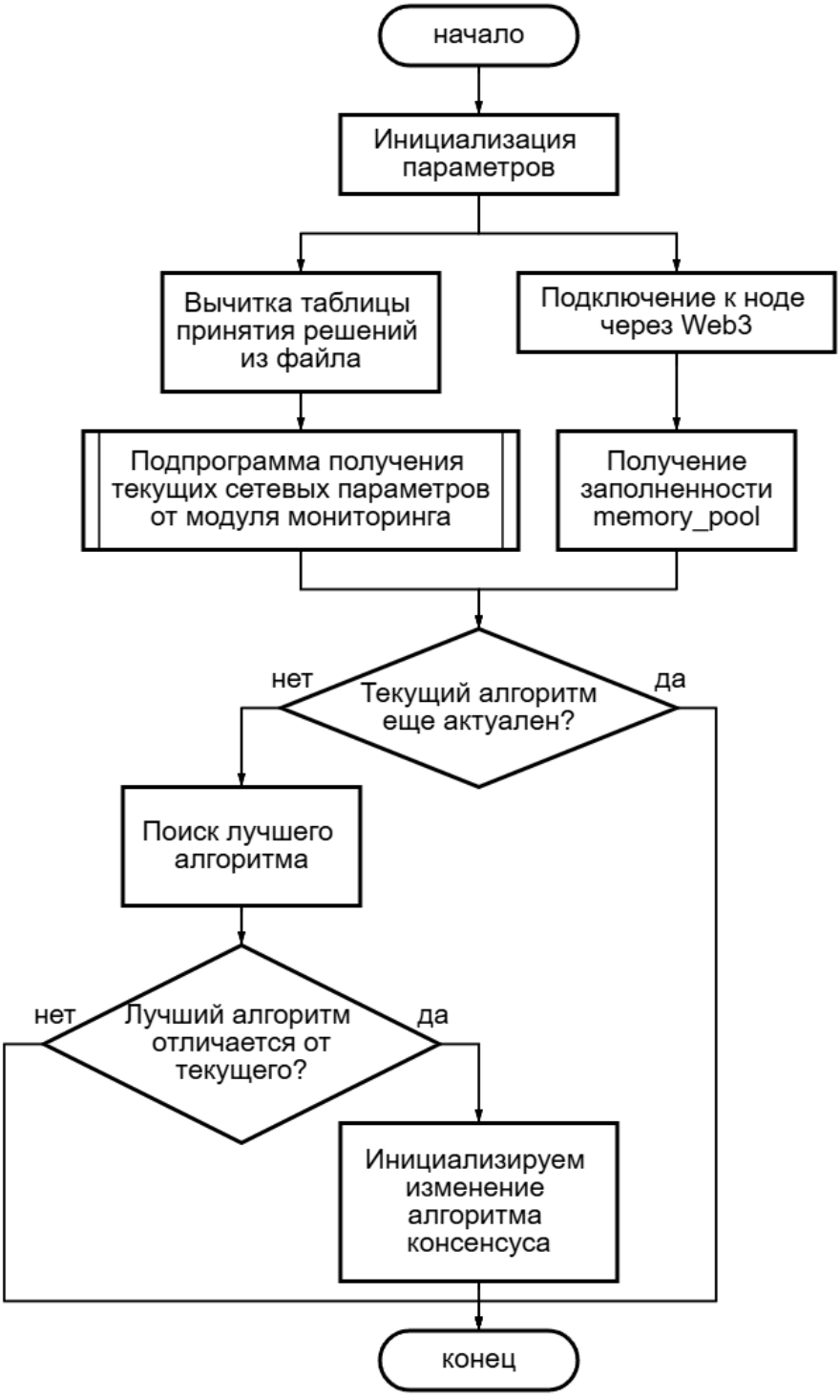


Рисунок 29 – Алгоритм работы модуля принятия решения

На основе собранных метрик, модуль анализирует, соответствует ли текущий алгоритм консенсуса условиям из таблицы. Если нет, он ищет альтернативные алгоритмы, которые лучше подходят для текущей ситуации. При необходимости модуль переключает текущий алгоритм консенсуса на более подходящий. Это переключение фиксируется в логах, что позволяет отслеживать изменения и их влияние на производительность сети.

Алгоритм работает в бесконечном цикле, обновляя метрики и проверяя необходимость переключения каждую определенный промежуток времени. Это обеспечивает оперативное реагирование на изменения в состоянии сети.

3.2 Аналитическая модель сети связи с блокчейном

Процесс обработки трафика, проходящего через узел блокчейн-сети, можно представить в виде многоканальной системы массового обслуживания с ограниченными очередями, неоднородным потоком запросов и системой относительных приоритетов.

Рассмотрим многоканальную СМО с возможностью отказа. В такой системе запросы поступают в однородном потоке и распределяются между свободными каналами. Если на момент поступления запроса все каналы заняты, он направляется в буфер с ограниченным объемом, при условии, что там есть свободное место. Если же буфер полностью заполнен, запросу отказывают в обслуживании, и он покидает систему необработанным. Таким образом, система формирует три потока: входящие запросы с интенсивностью λ, успешно обработанные запросы с интенсивностью λ', и запросы, которым было отказано в обслуживании, с интенсивностью λ''. Очевидно, что $\lambda=\lambda'+\lambda''$.

Определим, что $y = \lambda/\mu$ - обслуженных заявок в единицу времени, λ'– абсолютную пропускную способность СМО [45], $P_{отк}$ – вероятность отказа, $P_{оч}$ – вероятность образования очереди, m – среднее число находящихся в системе заявок, u – среднее время пребывания заявки в системе.

Вероятность p_0, что все каналы n будут находиться в свободном состоянии, вероятность p_1 о занятом только одном канале и другие можно получить по формулам (3.1, 3.2), что соответствует модели многоканальной системы массового обслуживания с ограниченной очередью [46]. Также с помощью формул (3.3), можно получить вероятности попадания заявки в определенные позиции очереди.

$$p_0 = \left(1 + \frac{y}{1!} + \frac{y^2}{2!} + \ldots + \frac{y^n}{n!} + \frac{y^{n+1}}{n \cdot n!} \cdot \frac{1 - \left(\frac{y}{n}\right)^n}{1 - y/n} \right)^{-1} \tag{3.1}$$

$$p_1 = y \cdot p_0, \ldots, p_n = \frac{y^n}{n!} \cdot p_0 \tag{3.2}$$

$$p_{n+1} = \frac{y^{n+1}}{n \cdot n!} \cdot p_0, \ldots, p_{n+M} = \frac{y^{n+M}}{n^M \cdot n!} \cdot p_0 \tag{3.3}$$

Соответствующие вероятности отклонения заявки $P_{отк,}$ и образования очереди $P_{оч}$ можно получить с помощью формул (3.4, 3.5) [47, 48].

$$P_{отк} = p_{n+M} = \frac{y^{n+M}}{n^M \cdot n!} p_0 \tag{3.4}$$

$$P_{оч} = \sum_{i=0}^{M-1} p_{n+i} = \frac{y^n}{n!} \cdot \frac{1 - \left(\frac{y}{n}\right)^M}{1 - y/n} \cdot p_0 \tag{3.5}$$

Среднее число заявок, находящихся под обслуживанием, найдем с помощью формулы (3.6) как среднее число занятых каналов, т. е. как среднее число обслуживаемых заявок за среднее время обслуживания одной заявки.

$$L_{об} = \bar{k}_{зан} = \lambda' / \mu = y \cdot \left(1 - \frac{y^{n+M}}{n^M \cdot n!} p_0\right) \quad (3.6)$$

Среднее число занятых каналов и среднее число заявок, находящихся в очереди можно получить с помощью формул (3.7, 3.8).

$$L_{оч} = l = \frac{y^{n+1}}{n \cdot n!} \cdot \frac{1 - \left(\frac{y}{n}\right)^M \left(M + 1 - \frac{M}{n} \cdot y\right)}{\left(1 - \frac{y}{n}\right)^2} \cdot p_0 \quad (3.7)$$

Соответственно можно получить среднее число заявок в системе, которое будет равняться сумме занятых каналов и количества заявок в очереди.

$$m = \bar{k}_{зан} + l \quad (3.8)$$

Для модуля принятия решения адаптивного выбора консенсуса блокчейн-сети основной задачей является регулирование блокчейн трафика для более эффективного использования полосы пропускания. В таком случае с перспективы модуля принятия решения пользовательский трафик является приоритетным, а трафик блокчейн менее приоритетным. Заявки, ожидающие обработки, распределяются между несколькими буферами с ограниченной вместимостью. Каждому классу заявок соответствует свой буфер. Для определения порядка обработки используется система относительных приоритетов: в первую очередь обслуживаются заявки с более высоким приоритетом. При этом, если поступает приоритетный запрос, он не прерывает выполнение уже начатой обработки менее приоритетного. Если буфер, соответствующий классу поступившего запроса, заполнен, запрос отклоняется [49].

Для уточнения модели можно добавить следующие характеристики:

1) Поток входящих запросов является неоднородным и делится на два класса.

2) Каждый класс заявок имеет свой отдельный буфер с ограниченной емкостью.

3) Буферы работают по принципу отсутствия вытеснения: если накопитель полностью занят, новые заявки отклоняются.

4) Обработка запросов ведется по приоритетам: запросы первого класса обслуживаются раньше, чем запросы второго класса.

Тогда среднее время пребывания заявок суммарного потока *и* будет рассчитываться по формуле (3.9).

$$u = \frac{\lambda'_{\text{польз}} w_{\text{польз}} + \lambda'_{\text{блок}} w_{\text{блок}}}{\lambda'} = \frac{m}{\lambda'} = \frac{\bar{k}_{\text{зан}} + l}{\lambda'} = \frac{L_{\text{об}} + L_{\text{оч}}}{\lambda'} \quad (3.9)$$

С учетом (3.4, 3.9) становится ясно, что регуляция трафика блокчейн напрямую влияет на трафик пользователя, давая ему больше пространства в случае необходимости и используя незадействованные каналы для обработки очереди блокчейн заявок. Таким образом задача адаптивного алгоритма сводится к нахождению оптимального значения $\lambda_{блок,}$ и минимизации вероятности отклонения заявки (3.9) для пользовательского трафика и трафика блокчейн.

3.3 Имитационная модель сети с модулем принятия решения адаптивного выбора алгоритма консенсуса блокчейн-сети

Имитационное моделирование играет ключевую роль в разработке алгоритма для сетей и описания процессов. Оно представляет собой методику изучения поведения сложных систем путём создания их компьютерной модели и проведения экспериментов с целью анализа и прогнозирования различных сценариев работы. Этот подход позволяет получить ценные данные о функционировании системы без

необходимости развертывания её в реальных условиях, что особенно важно в контексте сложных распределённых систем, таких как блокчейн.

Актуальность имитационного моделирования для модуля принятия решения обусловлена рядом факторов. Во-первых, блокчейн-сети являются распределёнными системами с большим количеством взаимодействующих узлов, что затрудняет аналитическое исследование всех аспектов их работы. Например, поведение сети зависит от множества переменных, таких как интенсивность пользовательского и блокчейн-трафика, параметры консенсуса, текущая пропускная способность сети, задержки и джиттер. Имитационное моделирование позволяет учесть эти параметры и их взаимосвязь, создавая реалистичную картину работы сети.

Во-вторых, модуль принятия решения адаптивного выбора консенсуса блокчейн-сети требует тестирования в условиях, максимально приближённых к реальным. Например, необходимо учитывать, как изменения параметров, таких как размер блока или частота его генерации, влияют на производительность системы, уровень задержек и заполненность пула необработанных транзакций. Проведение таких экспериментов в реальной сети сопряжено с высокими затратами времени и ресурсов, а также рисками возникновения сбоев. Имитационное моделирование устраняет эти риски, предоставляя безопасное пространство для проверки гипотез и корректировки алгоритма.

Кроме того, имитационное моделирование позволяет исследовать поведение сети при различных алгоритмах консенсуса. Каждый из этих алгоритмов имеет свои особенности, влияющие на потребление ресурсов, распределение нагрузки и сетевые характеристики. Моделирование дает возможность изучить, как адаптивный алгоритм будет реагировать на изменения условий в сети, использующей тот или иной консенсус. Это особенно важно для обеспечения универсальности и гибкости алгоритма.

Одним из ключевых преимуществ имитационного моделирования является возможность создания стресс-тестов для системы. Например, можно смоделировать резкое увеличение пользовательского трафика, сбои в работе

отдельных узлов или другие нестандартные ситуации, чтобы оценить устойчивость модуля и его способность адаптироваться к изменениям. Это помогает выявить слабые места в работе модуля принятия решения до его внедрения в реальную сеть.

Имитационное моделирование также важно для оценки эффективности модуля принятия решения. Сравнивая результаты работы сети до и после внедрения модуля, можно получить объективные данные о том, насколько он улучшает производительность, снижает задержки и оптимизирует использование ресурсов. Кроме того, моделирование позволяет прогнозировать долгосрочные последствия изменений в параметрах сети, что помогает избежать принятия ошибочных решений.

Для представления итоговой модели сети блокчейн с модулем принятия решения был использован метод имитационного моделирования и ПО Anylogic (Рисунок 30).

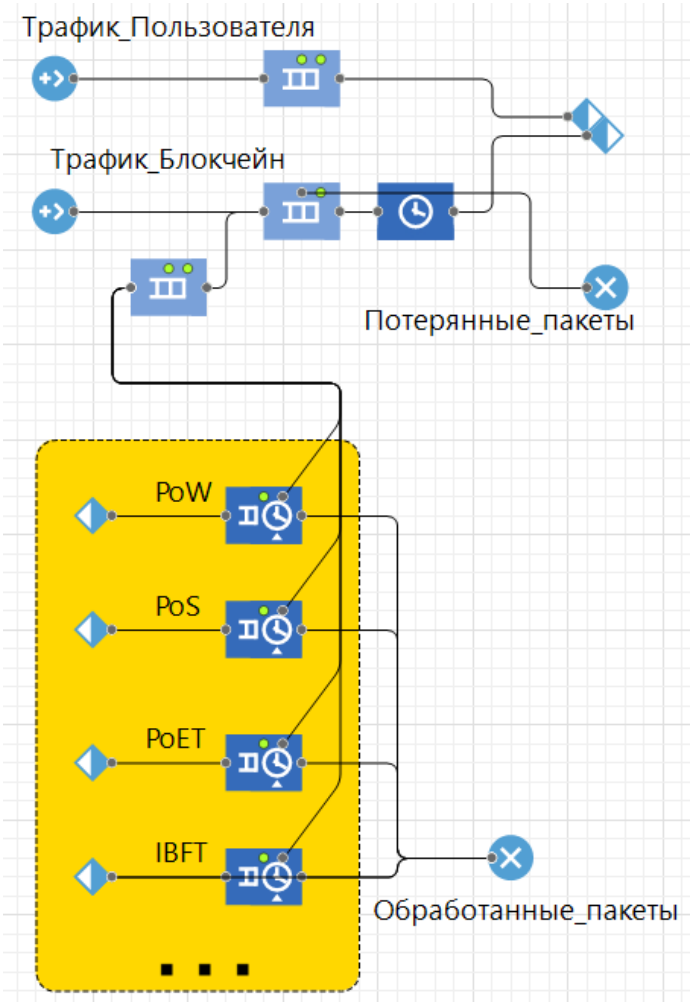


Рисунок 30 - Имитационная модель в программе AnyLogic

В качестве источника выступает пользовательский трафик и трафик сети блокчейн, включающий транзакции, блоки. В описанной модели данный тип трафика можно охарактеризовать как регулярный, стационарный и описать с помощью функции распределения $B_k(t)$ и преобразования Лапласа-Стилтьеса [50, 51], формула (3.10).

$$b_k^{(m)} = \int_0^\infty t^m dB_k(t) \tag{3.10}$$

Будем считать, что пакеты блокчейн трафика менее приоритетны, чем пакеты пользовательского трафика. Приоритет проявляется в момент выбора новой заявки на обработку – предпочтение отдастся пользовательскому трафику.

Также можно рассмотреть вопрос использования концепции абсолютного приоритета. Она заключается в поведении системы в случае получения приоритетной заявки. Система обязана прервать обработку текущей заявки одним из способов: выкинуть заявку из системы, приостановить обработку заявки для освобождения ресурсов на приоритетную или вернуть заявку в очередь. Система возвращается к обработке менее приоритетных заявок в случае приостановления потока приоритетных. Однако в текущем представлении сети данный подход крайне неэффективен, поэтому рассматривается только концепция относительного приоритета обработки потоков с приоритетами.

Далее в модели для блокчейн трафика представлен пул неподтвержденных транзакций, а также исходящий поток неподтвержденных (необработанных) транзакций, не вошедших в пул. Его цель - отразить эффективность алгоритма консенсуса в аспекте разбора потока транзакций в текущих сетевых условиях.

В следующем переходе трафик блокчейна и пользовательский трафик объединяются и следуют до алгоритма консенсуса. Объединенный трафик также подвергается анализу модулем мониторинга и модулем принятия решений для решения вопроса переключения сети на более эффективный алгоритм консенсуса.

Далее если для обработки приоритетного пользовательского трафика у сети не хватает ресурсов, то такой трафик выходит из системы не обработанным и не переданным далее, что в сформулированных требованиях системы представляется крайне редко и описывается формулой (3.10) о вероятности занятости всех каналов, однако для блокчейн трафика, который более подвергнут проблеме быть не обработанным имеется встроенный в блокчейн механизм – обратные возврат в пул неподтвержденных транзакций.

Обработка пакетов и транзакций проходит в большом модуле, который представляет собой множество возможных состояний системы в момент времени t. Этот модуль можно представить графом переходов марковских процессов [52] (Рисунок 31).

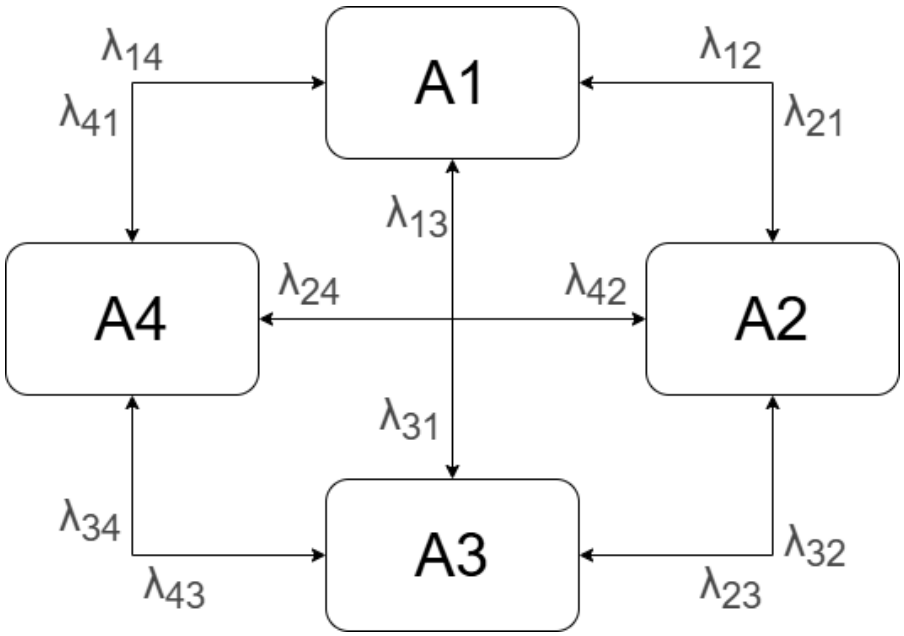


Рисунок 31 - Схематичное представление состояния системы только в одном алгоритме консенсуса с вероятностями перехода по состояниям

Исходя из данного графа можно обозначить, что вероятность того, что за время Δt система перейдет из состояния S_i в S_j приближенно равна интенсивности перехода на время переключения системы, формула (3.11).

$$P_{\Delta t}(S_i \to S_j) \approx \lambda_{ij} \Delta t \tag{3.11}$$

В качестве моделирования был рассмотрен сценарий увеличения нагрузки пользовательского трафика. Ожидаемое поведение системы включало в себя переключение алгоритма консенсуса сети блокчейн при выполнении запрограммированных триггеров на менее ресурсозатратный вариант. На Рисунке 32 приведены результаты аналитического и имитационного моделирования изменения использования полосы пропускания.

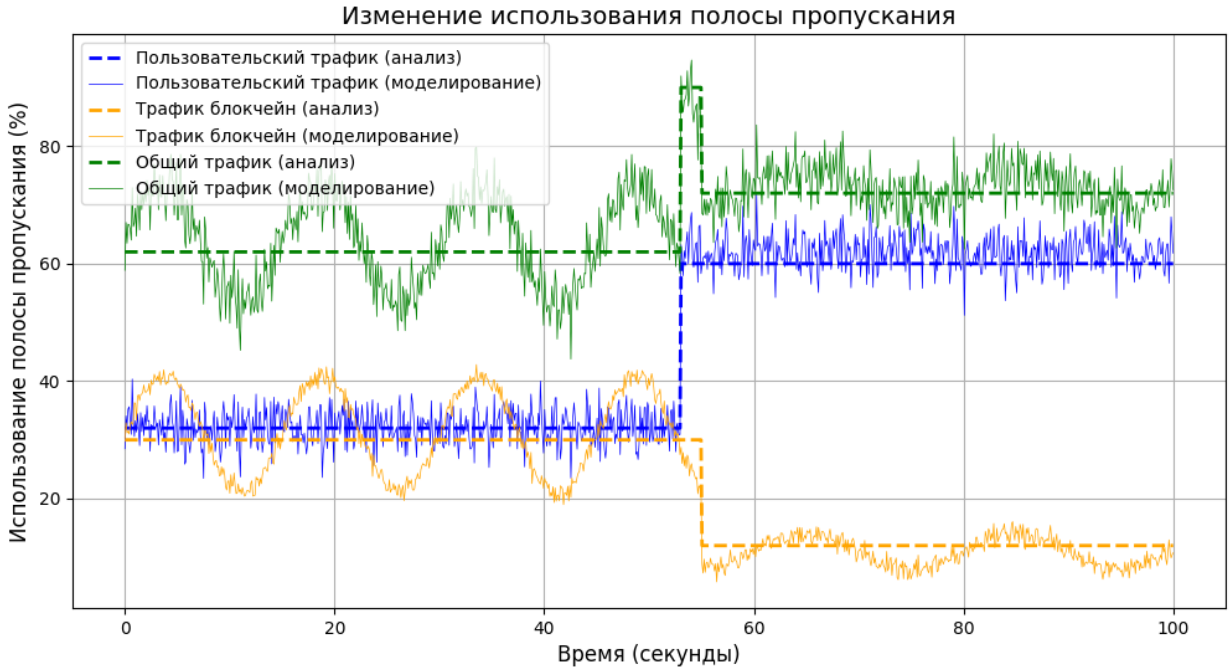


Рисунок 32 – Результаты аналитического и имитационного моделирования

Можно заметить, что в данном сценарии триггером для запуска протокола переключения алгоритма консенсуса выступило превышение использования полосы пропускания более чем на 80% после чего модуль принятия решения снизил активность сети блокчейн тем самым предоставляя ресурсы для пользовательского трафика.

3.4 Результаты имитационного моделирования

С помощью метода имитационного моделирования были сформированы и проанализированы однородные ситуации с одинаковыми сетевыми характеристиками, но различающимися алгоритмами консенсуса.

Результаты имитационного моделирования подтвердили важность использования модуля принятия решения о смене алгоритма консенсуса для управления нагрузкой в блокчейн-сети, особенно в условиях высокоинтенсивного трафика и значительных изменений сетевых характеристик. Например, алгоритм Proof of Work (PoW) [53, 54], несмотря на свою распространенность, продемонстрировал значительную неэффективность при увеличении потока транзакций (Рисунок 33). Потери полезной нагрузки блокчейн-трафика достигли порядка 30%, что свидетельствует о неспособности алгоритма справляться с интенсивной нагрузкой на сеть. Такая ситуация приводит к увеличению времени подтверждения транзакций, росту пула неподтвержденных транзакций и снижению общего качества обслуживания.

Введение механики адаптации к сетевым условиям в рамках модуля принятия решения, позволяющего переключаться между различными алгоритмами консенсуса, существенно изменило ситуацию. В рамках имитационного моделирования было показано, что при активации модуля адаптации сеть смогла автоматически перейти с PoW на более подходящий алгоритм Proof of Stake (PoS) в условиях увеличенной нагрузки. Это позволило значительно снизить задержки и увеличить пропускную способность сети. Далее, при дальнейшем росте заполненности пула неподтвержденных транзакций и усложнении условий обработки транзакций, сеть переключилась на алгоритм IBFT, который обеспечил устойчивую обработку транзакций даже при высоком уровне нагрузки.

Анализ показал, что адаптивный подход позволяет не только минимизировать потери, но и увеличивать эффективность использования ресурсов сети. Например, благодаря смене консенсуса на PoS и затем IBFT (PBFT) (Рисунок

34) [55], удалось обработать поток транзакций практически без потерь, сохранив стабильную работу сети и поддерживая минимальную задержку передачи данных. Это подчёркивает, что адаптивные механизмы обеспечивают более гибкое и точное распределение ресурсов, чем статичные подходы.

Важно отметить, что переход между алгоритмами консенсуса происходил в режиме реального времени, что было достигнуто за счёт постоянного мониторинга сетевых характеристик и автоматической корректировки параметров. Это не только позволило сети справляться с переменными нагрузками, но и предотвратило возникновение узких мест, таких как переполнение пула необработанных транзакций или значительное увеличение джиттера.

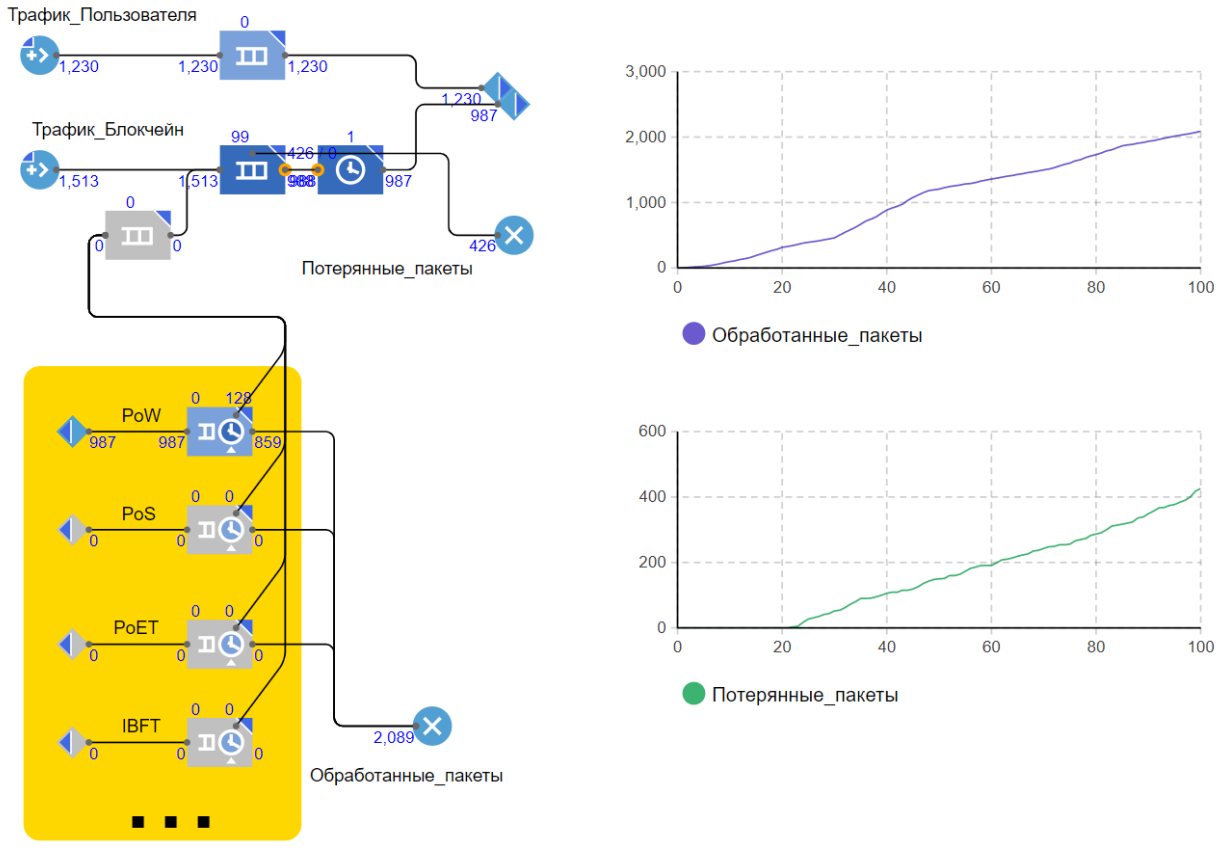


Рисунок 33 - Имитационная модель работы алгоритма консенсуса PoW

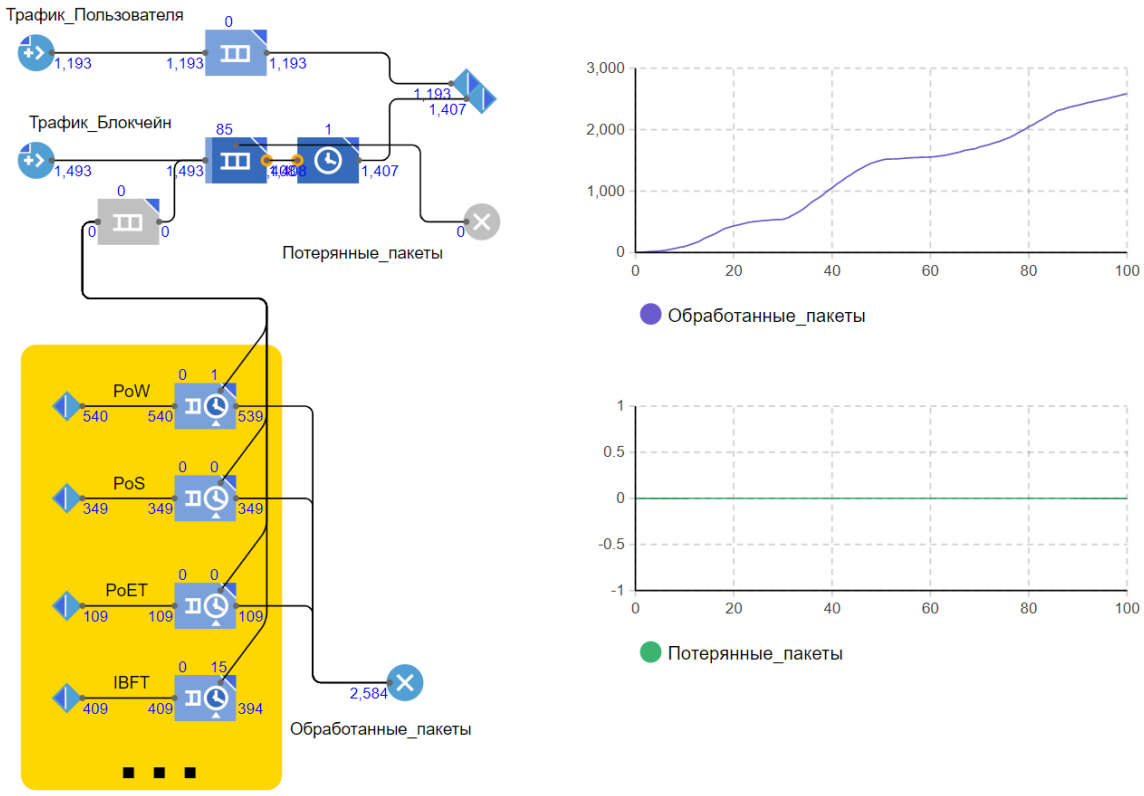


Рисунок 34 - Имитационная модель работы модуля принятия решения адаптивного переключения алгоритмов консенсуса на сети блокчейн

Результаты моделирования также демонстрируют преимущества модуля принятия решения в условиях долгосрочной работы сети. Например, в сценариях, где нагрузка на сеть изменялась в течение продолжительного времени, модуль принятия решения позволял сети сохранять стабильную производительность, оптимизируя параметры без необходимости вмешательства операторов. Это подтверждает, что внедрение такого подхода делает сеть более автономной и устойчивой к внешним факторам, включая резкое увеличение числа транзакций или нестабильность инфраструктуры.

В научной перспективе полученные результаты подчёркивают необходимость дальнейшего развития адаптивных механизмов для блокчейн-сетей. Сочетание мониторинга, анализа и динамической коррекции параметров позволяет существенно повысить производительность сети, её масштабируемость и устойчивость. Более того, имитационное моделирование подтвердило, что подобные системы могут быть эффективно применены не только в блокчейне, но и

в других распределённых системах, требующих адаптации к изменяющимся условиям.

Таким образом, результаты имитационного моделирования доказывают, что предлагаемый модуль принятия решения не только улучшает текущие показатели работы сети, но и открывает возможности для её устойчивого развития в условиях меняющихся требований и нагрузок.

3.5 Выводы

1) Представлена концепция модуля принятия решения адаптивного выбора блокчейн консенсуса на сети связи в зависимости от сетевых и аппаратных показателей оборудования и участка сети связи.

2) Разработана аналитическая модель сети с блокчейном.

3) Разработана имитационная модель сети с модулем принятия решения адаптивного выбора алгоритма консенсуса блокчейн-сети с использованием ПО Anylogic, проведено имитационное моделирование, подтверждающее расхождение между аналитическим расчетом и моделированием не более 5%.

ГЛАВА 4 МОДЕЛЬ И МЕТОДИКА ОЦЕНКИ ЭФФЕКТИВНОСТИ АДАПТИВНОГО ВЫБОРА БЛОКЧЕЙН-СИСТЕМ

Разработанный модуль принятия решения имеет ряд ограничений, обусловленных принципов функционирования блокчейн, в частности, скорость генерации блоков и количество участников, среди которых происходит распространение информации. Для эффективного использования модуля требуется разработать модель оценки эффективности предлагаемого подхода с учетом особенностей и структуры сети связи.

4.1 Оценка времени синхронизации узлов сети на примере ITS

Оценки времени синхронизации узлов может рассматриваться на примере регистрации данных об инцидентах, координаты, время, а также автомобили-свидетели (или потенциальные свидетели) [35, 56, 57]. Схема представлена на Рисунке 35.

Для реализации оперативной реакции системы на изменение и запуска одного из сценариев необходимы следующие компоненты:

– датчики, приложение на мобильном телефоне и другие компоненты системы, передающие сигнал о происходящем,

– смарт-контракт в блокчейн-сети, фиксирующий ключевую информацию об инцидентах (дата, время, координаты, идентификаторы участников),

– IPFS для хранения более подробной информации и файлов (опционально),

– слушатель событий смарт-контракта, отслеживающий появление новых записей и передающий сигнал соответствующим службам в случае возникновения чрезвычайной ситуации.

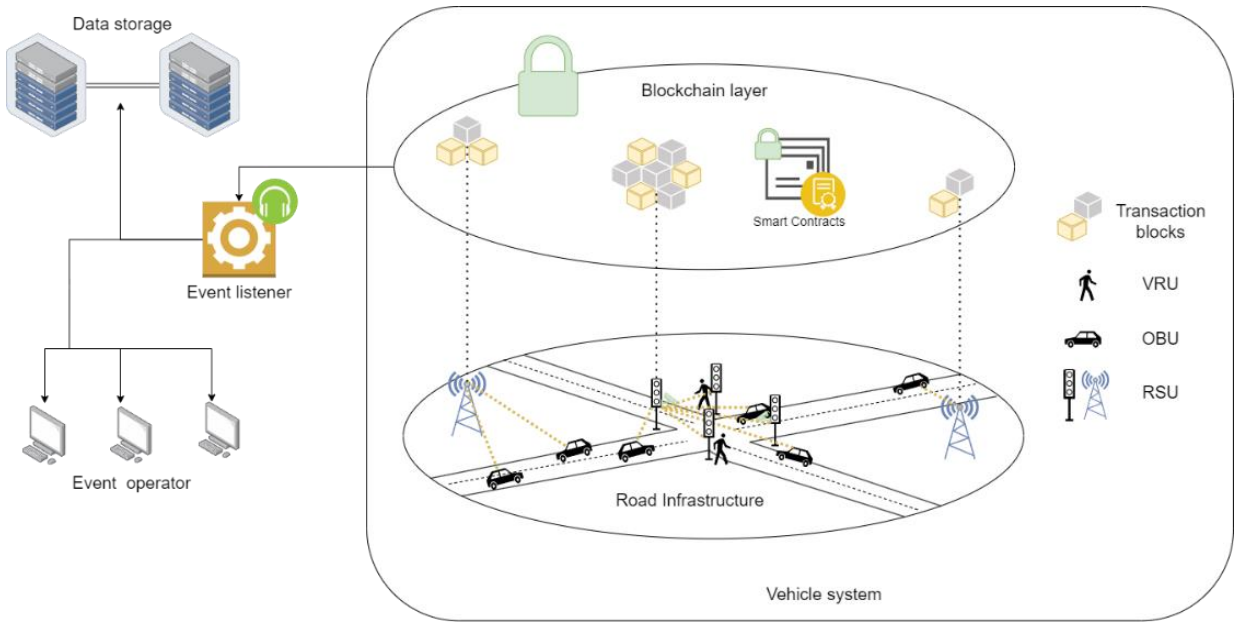


Рисунок 35 – Архитектура ITS для ситуации оперативного реагирования на происшествие

С учетом предлагаемой архитектуры возможны 2 основных сценария с VRU. Сценарий 1 предполагает, что информация о VRU, находящемся на маршруте движения транспортного средства, своевременно обнаружена, сигнал передан в RSU и OBU, и аварийная ситуация предотвращена за счет своевременной реакции всей системы. Сценарий 2 предполагает, что информация о VRU была получена или не получена, реакция системы RSU и OBU оказалась недостаточной. В этом случае информация об инциденте записывается в блокчейн-сеть, вызываются спецслужбы, а также записывается информация о сегменте инцидента с определенным радиусом для дальнейшего анализа и выяснения обстоятельств [58 - 61].

В интеллектуальных транспортных системах на основе технологии блокчейн критически важным параметром является скорость распространения информации по узлам сети, а также синхронизация. Это обусловлено необходимостью обеспечения своевременной передачи данных, что влияет на эффективность управления трафиком и быстрое принятие решений.

Ниже перечислены основные факторы, определяющие скорость распространения информации и синхронизации.

Архитектура сети блокчейн:

– топология сети (количество и расположение узлов),

– алгоритм консенсуса,

– пропускная способность каналов связи между узлами.

Объем и частота передаваемых данных:

– количество и размер транзакций/блоков,

– частота обновления информации о состоянии трафика.

Вычислительная мощность и производительность узлов:

– скорость процессора,

– объем оперативной памяти,

– эффективность алгоритмов обработки данных.

Сетевая инфраструктура:

– скорость и стабильность интернет-подключений,

– задержки при передаче данных (задержка),

– наличие резервных каналов связи.

Из этих факторов наиболее значимыми являются:

– размер блока: чем больше размер блока, тем больше времени требуется для передачи, это напрямую влияет на скорость,

– пропускная способность сети: определяет максимальный объем данных, которые могут быть переданы за единицу времени,

– задержка передачи в сети может существенно повлиять на общее время передачи блока,

– время проверки: узлы должны проверить блок перед его передачей, что также занимает время.

Перечисленные выше параметры оказывают общее влияние на скорость передачи блоков в сети, а также на дальнейшее распространение блоков информации.

Базовая формула для расчета средней скорости распространения блока V между узлами может быть представлена следующим образом (4.1).

$$V = \frac{S}{T}, \tag{4.1}$$

где S — размер блока (в байтах или мегабайтах), T — среднее время передачи блока (в секундах).

Однако, для учета факторов, влияющих на скорость передачи, формула должна содержать сведения о временной структуре (4.2).

$$T = T_{propagation} + T_{validation} + T_{network-delay}, \tag{4.2}$$

где *Tpropagation* — время распространения блока между узлами, *Tvalidation* — время, необходимое для проверки блока, *Tnetwork-delay* — задержки в сети. Для обеспечения высокой скорости распространения информации система должна быть оптимизирована по всем этим параметрам.

Высокая скорость распространения информации является ключевым фактором, обеспечивающим своевременное реагирование интеллектуальной транспортной системы на изменения дорожной ситуации и принятие оптимальных управленческих решений.

Согласно приведенным формулам (4.1, 4.2), скорость распространения информации (синхронизации) по сети состоит из 3 компонентов.

Рассмотрим аспект скорости валидации и создания блоков. Этот компонент зависит от выбора архитектуры сети, а именно алгоритма консенсуса, размера блока и других параметров сети.

Для исследования этого параметра использовался метод моделирования с использованием утилиты SIMBA, которая является усовершенствованной версией BlockSim [62, 63].

Основной задачей моделирования было рассмотрение скорости распространения информации (синхронизации сети) в зависимости от удаленности узлов друг от друга.

Для проведения моделирования были определены параметры сети с малым расстоянием (Рисунок 36) между узлами и большим расстоянием между узлами и запущена симуляция, в которой фиксировался факт отправки блоков транзакций с узлов и получения их на других узлах (Рисунок 37).

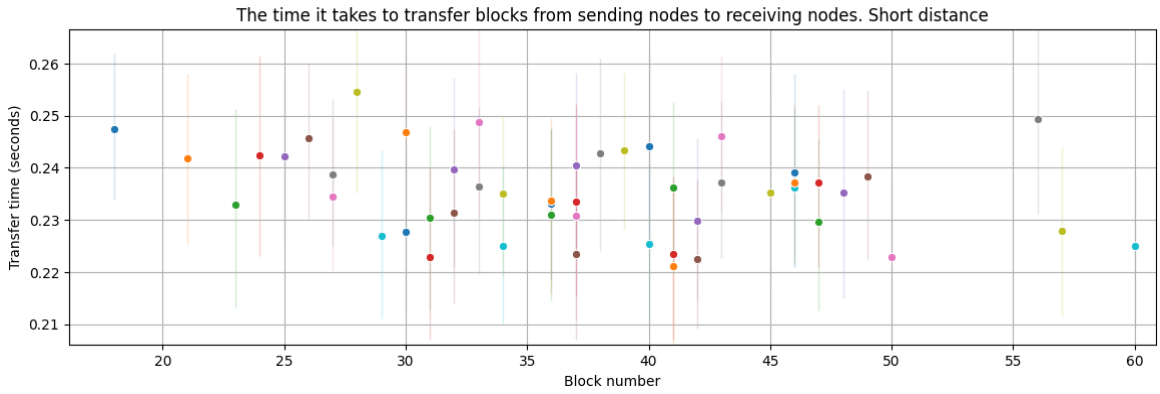


Рисунок 36 – Передача блока между узлами близкого расстояния

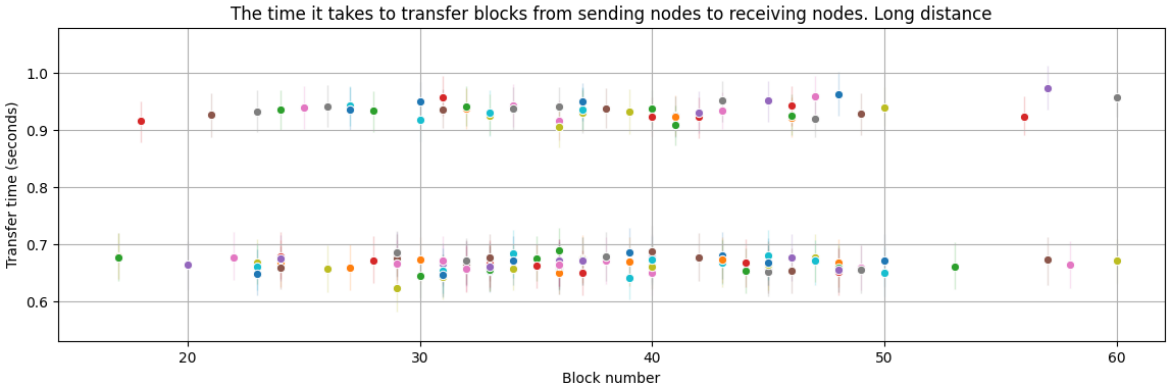


Рисунок 37 – Передача блока между узлами дальнего расстояния

Каждую группу узлов, на которые был отправлен новый блок с узла-отправителя или узла-создателя блока, можно увидеть на графике в виде одной одноцветной линии с точкой посередине, где также выделено среднее время передачи блока данных.

Также на графиках можно увидеть случаи, когда одному блоку соответствует несколько подобных групп узлов.

Когда на одной оси находятся две и более точек разного цвета, это означает, что отправленный узлами блок был получен другими узлами, что инициировало процесс дальнейшей рассылки блока данных на другие узлы, то есть процесс

синхронизации сети был инициирован и поддержан другими участниками. Другими словами, оси, на которых находятся две или более точек, являются осями, на которых синхронизированы два или более узлов соответственно.

По результатам, полученным в рамках моделирования, видно, что скорость синхронизации сети для удаленных узлов может достигать 1 секунды, что является хорошим показателем по сравнению со скоростью создания транзакции и ее проверки, а также скоростью создания блока в системе [64, 65].

Однако для сценариев быстрого реагирования больший интерес представляет график близко расположенных узлов, скорости синхронизации которых колеблются около 0,25 секунды, поскольку узлы в проектируемой архитектуре будут располагаться в достаточной близости к другим узлам и центрам для быстрого реагирования на нештатные инциденты. Данный результат показывает приемлемый уровень скорости и доказывает возможность использования сценариев быстрого реагирования в предлагаемой архитектуре с интеграцией технологии блокчейн.

4.2 Расчет временных характеристик распространения информации по сети связи

Как можно заметить из приведенной схемы на Рисунке 26, распространение блоков информации напоминает систему графов.

Одной из важных характеристик графа является его диаметр *D(G)*, который определяет максимальное расстояние между любыми двумя узлами. Диаметр графа может быть использован для оценки времени полного распространения информации [66].

Для сетей с топологией малого мира, например, в децентрализованных системах (сеть малого мира — это тип графа, в котором большинство узлов не являются соседями друг друга, но до большинства узлов можно добраться из

любого другого с помощью небольшого числа переходов) диаметр графа *D(G)* приближенно равен $log\ N$, где *N* — количество узлов в сети.

Если предположить, что каждое сообщение передается с фиксированной средней частотой c (обратная величина к времени передачи сообщения τ_{ij}, где τ_{ij} — время передачи от узла i к узлу j,), то время полного распространения информации по графу G можно оценить через диаметр (4.3).

$$T_{full} \approx \frac{D(G)}{c}, \tag{4.3}$$

где c — частота передачи сообщений между узлами (обратная среднему времени передачи сообщения), *D(G)* — диаметр графа сети *G* (максимальное расстояние между узлами), T_{full} — время полного распространения информации о переключении алгоритма консенсуса по сети.

Во время периода T_{full} узлы, которые еще не получили информацию о переключении, продолжают генерировать блоки по старому алгоритму. Это может привести к конфликтам или форкам в блокчейне, что неизбежно ведет к потери блоков транзакций [53].

Пусть $P_{informed}(t)$ — доля узлов, получивших информацию к моменту времени *t*. Смоделируем распространение информации по экспоненциальной функции (4.4).

$$P_{informed}(t) = 1 - e^{-\beta t}, \tag{4.4}$$

где β — коэффициент скорости распространения информации, зависящий от c и топологии сети.

Доля узлов, не получивших информацию к моменту *t* представлена в (4.5).

$$P_{uniformed}(t) = e^{-\beta t} \tag{4.5}$$

Поскольку процессы происходят в сетях с множеством узлов и зависят от множества случайных факторов, экспоненциальная функция помогает отразить вероятностные свойства систем и их динамику.

С помощью полученных выражений для расчета скорости распространения информации на основе графов можно представить модель для расчета эффективности адаптивного алгоритма на сети связи.

4.3 Модель оценки эффективности разработанного модуля адаптивного выбора консенсуса

Оценка эффективности модуля принятия решения напрямую зависит от количества создаваемых блоков, которые не будут утеряны в рамках смены алгоритма консенсуса [67, 68].

Для оценки общего числа блоков, созданных по исходному алгоритму консенсуса B_{old} необходимо вычислить интеграл, представляющий собой произведение количества узлов N, интенсивности генерации блоков λ и вероятности того, что узел ещё не получил информацию о переключении алгоритма консенсуса $P_{uninformed}$, в течение времени от $t = 0$ до $t = T_{full}$ — полного времени переключения всех узлов сети:

$$B_{old} = N\lambda \int_0^{T_{full}} P_{uninformed}(t)dt = N\lambda \int_0^{T_{full}} e^{-\beta t}dt \qquad (4.6)$$

Выполнив преобразования получим:

$$B_{old} = N\lambda(\frac{1-e^{-\beta T_{full}}}{\beta}) \qquad (4.7)$$

Если $\beta T_{full} \gg 1$, то $e^{-\beta T_{full}} \approx 0$, и тогда:

$$B_{old} \approx \frac{N\lambda}{\beta} \tag{4.8}$$

В таком случае, общее количество блоков, созданных всеми узлами за период времени T составит:

$$B_{total} = N\lambda T, \tag{4.9}$$

где N – общее количество узлов, λ – средняя частота создания блоков узлом.

Разумно будет ввести понятие «полезных» блоков. Так как часть блоков, создаваемых во время смены консенсуса на сети, может быть утеряна, то абсолютно корректными с точки зрения гарантии принятия сетью можно считать блоки, созданные в сети, когда она полностью перешла на другой алгоритм консенсуса.

Полезные блоки — это блоки, созданные после полного переключения на новый алгоритм консенсуса, то есть после времени T_{full}:

$$B_{useful} = N\lambda\left(T - T_{full}\right) \tag{4.10}$$

Эффективность η адаптивного алгоритма можно определить как отношение числа полезных блоков к общему числу созданных блоков:

$$\eta = \frac{B_{useful}}{B_{total}} = \frac{N\lambda\left(T - T_{full}\right)}{N\lambda T} = 1 - \frac{T_{full}}{T} \tag{4.11}$$

Выражение демонстрирует (4.11), что эффективность уменьшается линейно с увеличением времени распространения информации T_{full} относительно общего времени T.

Учтем влияние количества узлов и скорости передачи данных. Подставим выражение для T_{full}:

$$\eta = 1 - \frac{T_{full}}{T} = 1 - \frac{D(G)}{cT} \quad (4.12)$$

При фиксированном T эффективность η зависит от:

– диаметра графа $D(G)$: чем больше диаметр, тем больше T_{full} и ниже эффективность,

– частота передачи сообщений c: чем больше частота передачи, тем меньше T_{full} и выше эффективность.

Также необходимо учесть влияние скорости передачи транзакций на обработку и формирования блока, который затем распространяется по сети, уменьшая пропускную способность всего другого трафика канала связи.

Пусть частота создания транзакций λ_{tx} влияет на нагрузку сети и потенциально на время передачи сообщений, можно выразить с как функцию от λ_{tx}:

$$\mathrm{c} = \mathrm{c}_0 - k\lambda_{tx}, \quad (4.13)$$

где c_0 — базовая частота передачи при отсутствии транзакций, k — коэффициент, показывающий, как увеличение скорости транзакций уменьшает частоту передачи сообщений из-за сетевой загруженности.

Обобщенная математическая модель для оценки эффективности адаптивного алгоритма с учетом количества узлов N, скорости передачи данных c, скорости создания транзакций λ_{tx} и времени T:

$$\eta = 1 - \frac{D(G)}{(c_0 - k\lambda_{tx})T} \quad (4.14)$$

Учтем, что диаметр графа $D(G)$ обычно зависит от количества узлов N. Для графов типа «малый мир» или случайных графов Эрдаша-Реньи [69, 70] принимает вид $D(G) \approx \log N$ и выражение эффективности адаптивного алгоритма примет вид:

$$\eta = 1 - \frac{\log N}{(c_0 - k\lambda_{tx})T}, \quad (4.15)$$

Теперь расширим модель, добавив в нее зависимость от времени генерации блоков T_{block} и размера блоков B_{size} с коэффициентом α для корректировки уровня влияния размера блока на сеть. Время генерации блоков и их размер ключевые параметры, так как при активной генерации транзакций в сети начнет формироваться буфер (очередь), который в блокчейне является пулом неподтвержденных транзакций, что влияет на нагрузку сети и эффективность работы консенсуса [71, 72].

С учетом обозначенных добавляемых параметров модель примет следующий вид:

$$\eta = 1 - \frac{D(G)*\alpha*B_{size}}{(c_0 - k\lambda_{tx})*T*T_{block}} \quad (4.16)$$

Исходя из выражений, представленных выше, можно заключить следующее:

– эффективность рассчитывается в процентах, как отношение потерянных блоков к общему количеству блоков с учетом набора параметров,

– при увеличении количества узлов N: эффективность η уменьшается логарифмически, так как увеличивается диаметр графа,

– при увеличении частоты передачи c: эффективность η увеличивается, так как уменьшается время распространения информации,

– при увеличении времени создания блока T_{block}: эффективность η увеличивается, так как уменьшается количество некорректно создаваемой информации,

– при увеличении частоты создания транзакций λ_{tx}: эффективность η уменьшается, если λ_{tx} существенно влияет на c.

4.4 Апробация математической модели оценки эффективности модуля принятия решения

Финальный вид математической модели расчета эффективности модуля принятия решения представлен в выражении 4.16, где:

- η — эффективность адаптивного алгоритма для участка сети,
- $D(G)$ — диаметр сети (графа),
- α — коэффициент, учитывающий влияние размера блока на сеть,
- B_{size} — размер блока (в транзакциях),
- c_0 — частота передачи данных,
- λ_{tx} — частота создания транзакций (в транзакциях в секунду),
- k — коэффициент, корректирующий влияние транзакций на сеть,
- T — общее время распространения информации по сети,
- T_{block} — время генерации блока.

Для оценки корректности работы представленной модели проведем теоретический расчет и моделирование для сравнения результатов [73]. Для экспериментального моделирования примем, что параметры сети:

- количество узлов N: 100 шт.,
- частота передачи данных c_0: 100 сообщений в секунду,
- размер блока B_{size}: 50 транзакций на блок,
- частота создания транзакций λ_{tx}: от 50 до 200 транзакций в секунду,
- время генерации блока T_{block}: от 10 до 40 секунд на блок,
- время наблюдения T: 200 секунд.

Результаты проведенного аналитического расчета и моделирования с применением программного скрипта, написанного на Python, приведены в Таблице 5. На Рисунке 38 приведен сравнительный график теоретических и экспериментальных расчетов, демонстрирующий, что величина расхождения между аналитическим расчетом и экспериментальным составляет не более 5%.

Таблица 5 - Сравнительные данные теоретических и экспериментальных расчетов

Кол-во узлов N (шт.)	**Диаметр D(G)**	**Время генерации блока Tblock (сек)**	**Частота создания транзакций λtx (транзакций/сек)**	**Эффект-ть теоретическая η (%)**	**Эффект-ть экспериментальная η (%)**	**Отклонение (%)**
100	4.6	10	50	79.8%	77.9%	-1.9%
100	4.6	20	50	82.5%	81.2%	-1.3%
100	4.6	40	50	85.0%	84.1%	-0.9%
100	4.6	50	50	86.2%	85.4%	-0.8%
100	4.6	10	200	64.5%	62.1%	-2.4%
100	4.6	20	200	67.3%	65.8%	-1.5%
100	4.6	40	200	70.2%	69.0%	-1.2%
100	4.6	50	200	72.1%	71.3%	-0.8%

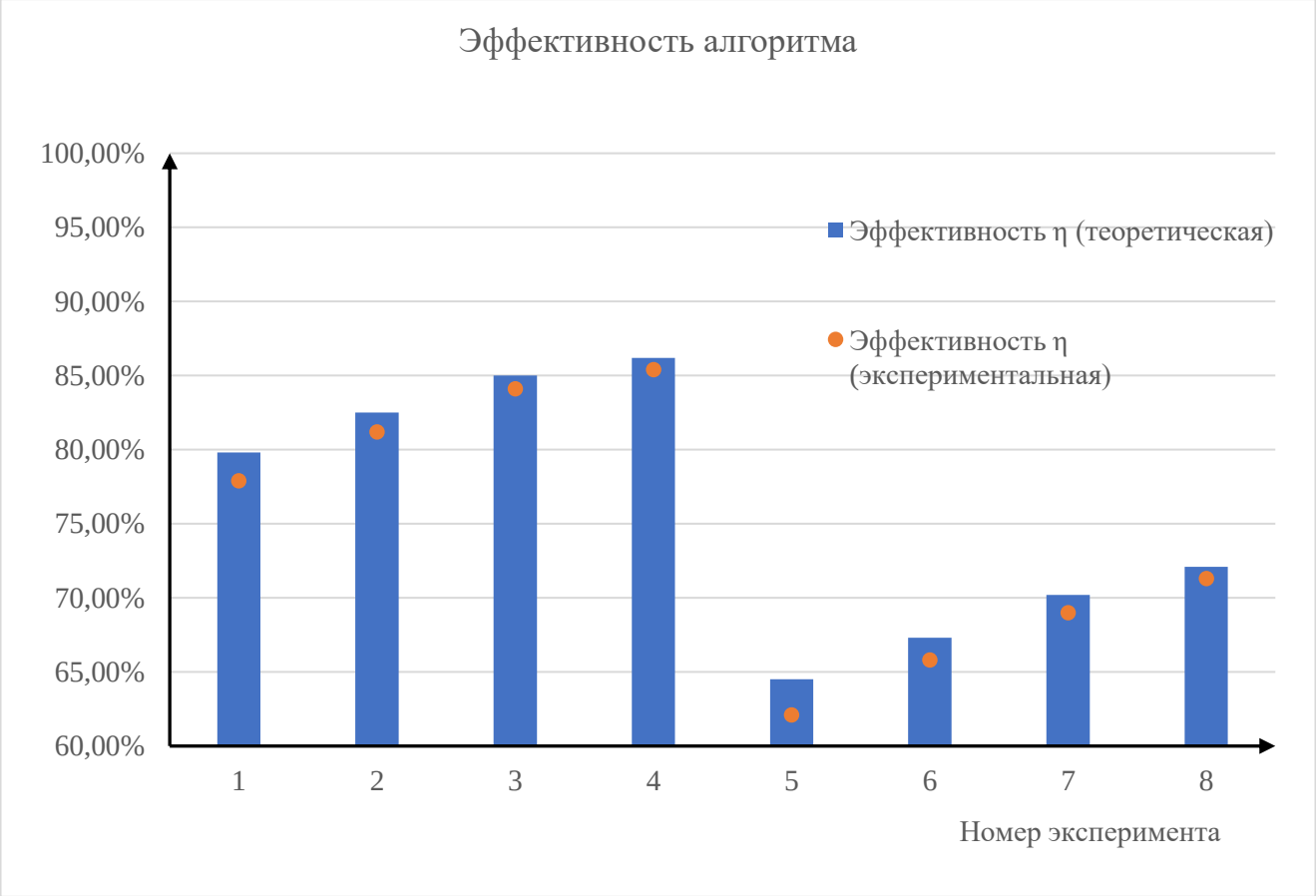


Рисунок 38 - Сравнительный график теоретических и экспериментальных расчетов

С увеличением времени генерации блоков эффективность алгоритма повышается, что связано с меньшим количеством блоков, созданных по старому алгоритму.

С увеличением частоты создания транзакций λ_{tx} эффективность снижается, особенно при более высоких значениях λ_{tx}. Это связано с тем, что в блоке с большей частотой создаётся большее количество транзакций, и, соответственно, больше из них теряется, если блок сгенерирован узлами, не перешедшими на новый алгоритм консенсуса.

Высокая частота создания транзакций существенно снижает эффективность сети, так как большее количество транзакций теряется до завершения процесса переключения алгоритма консенсуса [74].

4.5 Методика интеграции модуля принятия решения

Интеграция модуля принятия решения адаптивного выбора блокчейн-систем представляет собой сложную задачу, направленную на повышение эффективности работы сети в условиях изменяющихся характеристик передачи данных. Основной особенностью данного подхода является способность алгоритма, являющегося основой модуля, динамически адаптировать параметры блокчейн-сети в зависимости от текущего состояния сети передачи данных, таких как пропускная способность, задержка, джиттер, загрузка пула неподтвержденных транзакций и нагрузка на центральный процессор. Это позволяет повысить производительность сети, снизить задержки при передаче данных и обеспечить устойчивость к нагрузкам.

Модуль принятия решения адаптивного выбора консенсуса блокчейн-сети предполагает использование аналитической модели, которая регулярно оценивает состояние сети, отслеживая ключевые сетевые характеристики. На основе этих данных происходит динамическая коррекция параметров, включая размер блока, частоту генерации блоков, сложность вычислений для консенсуса, а также

максимальное количество транзакций в блоке. Такая архитектура позволяет гибко реагировать на изменения в сети, например, на увеличение пользовательского трафика или временные скачки интенсивности передачи данных.

Для каждого алгоритма консенсуса адаптивный подход имеет свои особенности. В алгоритме PoW наиболее важным параметром является сложность вычислений, которая напрямую влияет на время подтверждения блока [75, 76]. В условиях перегрузки сети алгоритм может снижать сложность, сокращая время генерации блоков и уменьшая нагрузку на сеть. В PoS основное внимание уделяется механизму выбора валидатора, где адаптивный алгоритм может регулировать частоту генерации блоков и количество транзакций, входящих в блок. В алгоритме DPoS ключевой задачей является перераспределение нагрузки между делегатами, что позволяет избежать перегрузок на отдельных узлах [77 - 79]. В случае PoET адаптивный подход может влиять на длительность временных интервалов ожидания [80], регулируя общую нагрузку на систему [81, 82].

Методика интеграции модуля принятия решения.

Шаг 1. Анализ сети передачи данных и блокчейн-архитектуры.

На первом этапе необходимо провести глубокий анализ сетевых компонентов и физических устройств сети.

Данный этап включает в себя сбор статистических данных системы, а также исследование граничных значений сетевых характеристик, структуры блокчейн-сети: распределение узлов, поддерживаемые алгоритмы консенсуса и архитектурные ограничения. Результаты анализа используются для выбора параметров адаптации.

Также на этом этапе можно рассчитать теоретическую ожидаемую эффективность модуля принятия решения на произвольных периодах времени с помощью формулы (4.16).

Шаг 2. Разработка аналитической модели сети и алгоритма консенсуса.

Корректируется ранее представленная математическая модель, описывающая работу сети в зависимости от её параметров. Она должна учитывать:

потоки пользовательского и блокчейн-трафика, параметры блокчейн-сети (размер блока, частоту генерации и т. д.).

А также для каждого алгоритма консенсуса описываются специфические аспекты, которые могут подвергаться адаптации.

Шаг 3. Интеграция модуля мониторинга сетевых характеристик.

Модуль мониторинга внедряется в узлы сети с целью отслеживания данных о пропускной способности, задержке, джиттере и текущей загрузке узла (CPU, память, диск). Регулярный мониторинг заполненности пула неподтвержденных транзакций и скорости обработки транзакций также входит в обязанности модуля мониторинга.

Агрегированная информация поступает в модуль принятия решений в режиме реального времени.

Шаг 4. Интеграция адаптивного модуля принятия решений.

Основой адаптивного подхода является модуль, анализирующий собранные данные и инициирующий изменения в параметрах сети блокчейна для корректировки нагрузок на сеть передачи данных. В связи с этим следующим шагом является интеграция данного модуля в узлы сети. В свою очередь модуль использует набор правил и условий для изменения параметров сети.

Шаг 5. Калибровка параметров адаптивного алгоритма.

На основании первоначального мониторинга проводится уточнение параметров. Корректируются оптимальные пороговые значения для каждого сетевого показателя, настраиваются временные интервалы реакции модуля принятия решения на изменения сети, а также уточняется логика адаптации для различных сценариев.

Шаг 6. Оценка эффективности и корректировка на основе реальных данных.

После развертывания модуля принятия решения необходимо регулярно оценивать его эффективность. Для этого производятся расчеты и сравнение сетевых характеристик до и после внедрения адаптивного подхода.

Также в этот этап включены работы по анализу вероятных проблем, таких как избыточная реакция модуля на незначительные изменения сети. При необходимости алгоритм дорабатывается с учетом новых условий работы.

Шаг 7. Подготовка методических рекомендаций и масштабирование.

После успешной интеграции и стабилизации модуля необходимо сформировать методические рекомендации по его использованию для администраторов сети - руководство по настройке параметров адаптации.

Для дальнейшей перспективы развития модуля принятия решения и сети в целом проработать сценарии масштабирования модуля на более крупные сети или новые алгоритмы консенсуса и выработать подходы к интеграции с различными уровнями сетевой инфраструктуры, включая использование облачных технологий для повышения производительности.

Представленная методика обеспечивает комплексный подход к интеграции модуля принятия решения адаптивного выбора консенсуса блокчейн сети, что позволяет не только повысить производительность блокчейн-сети, но и сделать её более устойчивой к изменяющимся условиям передачи данных.

4.6 Методика оценки эффективности адаптивного выбора блокчейн-систем с учетом характеристик трафика в сетях связи

Шаг 1. Определение исходных параметров.

Необходимо определить общее количество узлов в сети, среднюю частоту генерации блоков и временной промежуток, на котором рассчитывается эффективность. Эти параметры являются ключевыми для последующих расчетов.

Шаг 2. Определение общего количества блоков.

Необходимо рассчитать общее количество блоков, созданных всеми узлами за определенный период. Это значение будет служить основой для последующей оценки эффективности модуля принятия решения.

Шаг 3. Определение приблизительного числа блоков неактуального алгоритма консенсуса, на определенном временном промежутке.

Для определения общего количества блоков, созданных по старому алгоритму, необходимо провести расчеты. Этот процесс включает в себя умножение числа узлов на уровень генерации блоков и вероятность осведомленности. Результат позволяет оценить, сколько блоков было создано в условиях, когда некоторые узлы не получили информацию о новой системе.

Шаг 4. Расчет эффективности модуля принятия решения адаптивного выбора консенсуса блокчейн-сети.

Эффективность модуля можно определить с помощью формулы 4.14. Это позволяет количественно оценить, насколько хорошо работает модуль по сравнению со статичным консенсусом.

Подставив соответствующие значения для времени распространения информации, можно заметить зависимость эффективности от диаметра сети, который характеризует максимальное расстояние между узлами. Частота генерации транзакций может существенно влиять на нагрузку сети и, соответственно, на время передачи сообщений между узлами.

Шаг 5. Влияние диаметра сети на эффективность.

Для графов с характеристиками «малого мира» или случайных графов можно установить зависимость диаметра от числа узлов. Это позволяет дополнительно упростить расчет эффективности модуля принятия решения а и тогда можно воспользоваться формулой 4.16.

Шаг 6. Выводы о влиянии различных факторов на эффективность.

В заключение можно сделать ряд выводов о влиянии различных факторов на эффективность предлагаемого подхода. Увеличение числа узлов приводит к логарифмическому снижению эффективности, тогда как повышение скорости передачи данных способствует ее росту. Увеличение времени генерации блоков может повысить эффективность, тогда как увеличение частоты создания транзакций может привести к ее снижению, если оно существенно влияет на скорость передачи данных.

4.7 Выводы

1) Проведена оценка скорости распространения информации по сети в рамках модельного стенда ITS.

2) Разработана модель оценки эффективности разработанного модуля принятия решения адаптивного выбора алгоритма блокчейн-систем, учитывающая ряд параметров.

3) Проведена апробация разработанной модели оценки эффективности с применением аналитического расчета и имитационного моделирования.

4) Представлены методики интеграции модуля принятия решения и оценки его эффективности.

ЗАКЛЮЧЕНИЕ

В данной работе был разработан и протестирован модуль принятия решения по выбору блокчейн-систем для снижения числа потерянных блоков транзакций с учётом влияния сетевых характеристик. Процесс разработки включал в себя несколько ключевых этапов, каждый из которых был тщательно проанализирован и оптимизирован для достижения максимальной эффективности:

1) Проведен анализ сетевых характеристик и архитектуры блокчейн-сети. Важнейшими параметрами для адаптации стали пропускная способность, задержка, джиттер, а также другие характеристики, влияющие на стабильность работы сети. Сформирована теоретическая основа для динамического регулирования параметров работы блокчейн-сети в зависимости от изменений сетевых условий.

2) Разработана аналитическая модель сети связи с блокчейном. В процессе разработки, уточнялись основные параметры, такие как размер блока, частота генерации блоков и типы консенсуса, которые могут быть адаптированы в зависимости от текущего состояния сети. Этап создания аналитической модели позволил точно охарактеризовать поведение сети и выявить возможные слабые места, требующие улучшений.

3) Разработана имитационная модель сети связи с модулем принятия решения по выбору блокчейн-систем для снижения числа потерянных блоков транзакций. Результаты аналитического расчета и имитационного моделирования демонстрируют уровень расхождения не выше 5%.

4) Интегрирован модуль мониторинга сетевых характеристик. Модуль - ключевое звено системы, которое позволяет в реальном времени отслеживать такие параметры, как пропускная способность сети, задержка, джиттер, а также загрузка процессора и памяти узлов. Эти данные позволяют алгоритму принимать решения о необходимости переключения на более подходящий алгоритм консенсуса в зависимости от текущей нагрузки на сеть.

4) Интегрирован адаптивный модуль принятия решений, который, основываясь на собранных данных, может в режиме реального времени изменять параметры работы сети. Модуль принимает решения о переключении между алгоритмами консенсуса, что позволяет оптимизировать работу блокчейн-сети и снижать время обработки транзакций.

5) Выполнена калибровка параметров модуля принятия решения с уточнением пороговых значений для ключевых сетевых характеристик в количестве десяти единиц.

6) Разработана модель оценки эффективности модуля принятия решения адаптивного выбора консенсуса блокчейн-сети на основе реальных данных. В результате имитационного моделирования было продемонстрировано, что модуль способен значительно улучшить производительность сети, снизив потери данных и улучшив пропускную способность при повышенной нагрузке. Полученные результаты показали, что применение системы с модулем принятия решения и адаптивной сменой консенсуса сети на 10% эффективнее, чем при использовании статичного алгоритма консенсуса. Также были проанализированы возможные проблемы, такие как избыточная реакция модуля на малые изменения в сети, что позволило внести корректировки в работу системы.

7) На заключительном этапе были подготовлены методики интеграции модуля принятия решения и оценки его эффективности, которые предполагают его использование в более крупных сетях и с другими алгоритмами консенсуса. Методики также включает рекомендации по интеграции с различными уровнями сетевой инфраструктуры, что обеспечит гибкость и масштабируемость решения в будущем.

Проведенная работа продемонстрировала, что использование адаптивности при выборе и смены консенсуса в блокчейн-сетях позволяет не только повысить эффективность работы системы в условиях изменяющихся сетевых характеристик, но и значительно улучшить устойчивость и производительность сетей, обеспечивая их долговечность и масштабируемость в условиях высоких нагрузок.

СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ

БЧ	Блокчейн	
ИТС	Интеллектуальная транспортная	система
СМО	Система массового обслуживания	
ADNL	Abstract Datagram Network Layer	Сетевой уровень абстрактных датаграмм
API	Application Programming Interface	Программный интерфейс приложения
BFT	Byzantine Fault Tolerance	Византийская отказоустойчивость
BoC	Bag of Cells	Формат сериализации ячеек в массив байт
CIP	Cardano Improvement Proposal	Предложение по улучшению блокчейна Cardano
CPU	Central Processing Unit	Центральный процессор
DAG	Directed Acyclic Graph	Направленный ациклический граф
dApp	Decentralized Application	Децентрализованное приложение
DBFT	Delegated Byzantine Fault Tolerance	Делегированная византийская отказоустойчивость
DLT	Distributed Ledger Technology	Технология распределенного реестра
DPoS	Delegated Proof of Stake	Делегированное доказательство доли владения
eUTxO	Extended Unspent Transaction Output	Расширенный вывод неизрасходованных транзакций
FBA	Federated Byzantine Agreement	Федеративное византийское соглашение
Fee	Comission	Комиссия
HTTP	Hyper Text Transfer Protocol	Протокол прикладного уровня передачи данных
IP	Internet Protocol	Протокол сетевого уровня стека TCP/IP
IPFS	InterPlanetary File System	Распределенная система для хранения
ITS	Intelligent Transport System	Интеллектуальная транспортная сеть
SD-IoV	Software-defined Internet-of-Vehicles	Программно-определяемый Интернет-транспортных средств
LPoS	Leased Proof of Stake	Арендованное доказательство доли
MTU	Maximum Transmission Unit	Максимальная единица передачи
NVMe	Non-Volatile Memory Express	Интерфейс доступа к твердотельным накопителям

OBU	On-Board Unit	Бортовой блок
OSI	Open Systems Interconnection	Эталонная модель взаимодействия открытых систем
PBFT	Practical Byzantine Fault Tolerance	Практическая византийская отказоустойчивость
PoA	Proof of Activity	Доказательство активности
PoAuth	Proof of Authority	Доказательство авторитета
PoB	Proof of Burn	Доказательство сжигания
PoC	Proof of Capacity	Доказательство емкости
PoET	Proof of Elapsed Time	Доказательство истекшего времени
PoI	Proof of Importance	Доказательство важности
PoR	Proof of Reputation	Доказательство репутации
PoRes	Proof of Research	Доказательство исследования
PoS	Proof of Stake	Доказательство доли
PoW	Proof of Work	Доказательство работы
PoWeight	Proof of Weight	Доказательство веса
RAM	Random Access Memory	Оперативная память
RSU	Road Side Unit	Придорожный блок
SBFT	Simplified Byzantine Fault Tolerance	Упрощенная византийская отказоустойчивость
SSD	Solid-State Drive	Твердотельный накопитель
TCP	Transmission Control Protocol	Протокол управления передачей
TON	Telegram Open Network	Открытая сеть Telegram
TPS	Transaction Per Second	Транзакция в секунду
TTL	Time To Live	Время жизни
UDP	User Datagram Protocol	Протокол пользовательских датаграмм
VPN	Virtual Private Network	Виртуальная частная сеть
VRU	Vulnerable Road Users	Уязвимые участники дорожного движения

СПИСОК ЛИТЕРАТУРЫ

1) Z. Zheng, S. Xie, H. Dai, X. Chen and H. Wang, "An overview of blockchain technology: Architecture consensus and future trends", 2017 IEEE international congress on big data (BigData congress), pp. 557-564, 2017.

2) D. Mingxiao, M. Xiaofeng, Z. Zhe, W. Xiangwei and C. Qijun, "A review on consensus algorithm of blockchain", 2017 IEEE international conference on systems man and cybernetics (SMC), pp. 2567-2572, 2017.

3) Vladyko, A.; Spirkina, A.; Elagin, V. Towards Practical Applications in Modeling Blockchain System. Future Internet 2021, 13, 125. https://doi.org/10.3390/fi13050125.

4) Smetanin S, Ometov A, Komarov M, Masek P, Koucheryavy Y. Blockchain Evaluation Approaches: State-of-the-Art and Future Perspective. Sensors. 2020; 20(12):3358. https://doi.org/10.3390/s20123358.

5) Бахвалова Е.А., Судаков В.А. Исследование алгоритмов консенсуса для блокчейн-платформ // Препринты ИПМ им. М.В. Келдыша. 2021. № 26. 16 с. DOI:10.20948/prepr-2021-26.

6) S. Zhang, L. Ni, W. Xie, G. Li and H. Sun, "Research on Self-Adaptive Consensus Method Based on Blockchain," 2022 IEEE 14th International Conference on Advanced Infocomm Technology (ICAIT), Chongqing, China, 2022, pp. 292-297, doi: 10.1109/ICAIT56197.2022.9862639.

7) Помогалова А.В., Донсков Е.А., Елагин В.С. Модель интеграции адаптивного алгоритма выбора и смены консенсуса блокчейна при граничных значениях показателей сети. Электросвязь. - 2024. - №12-2. - С. 16-24.

8) Помогалова А.В. Оценка эффективности адаптивного алгоритма блокчейн-сетей, как части мультиконсенсусной системы на сетях связи. Труды учебных

заведений связи. 2024;10(5):34-42. https://doi.org/10.31854/1813-324X-2024-10-5-46-54.

9) Дворецков К.А., Мартынюк А.А., Помогалова А.В., Блокчейн как новый уровень развития современных баз данных. //АПИНО 2023. Сборник научных статей XII Международной научно-технической и научно-методической конференции. В 4-х томах. Под редакцией С.И. Макаренк, сост. В.С. Елагин, Е.А. Аникевич. Санкт-Петербург, 2023. Том 2. С. 617-622.

10) M. Crosby, P. Pattanayak, S. Verma and V. Kalyanaraman, "Blockchain technology: Beyond bitcoin", Applied Innovation, vol. 2, no. 6–10, pp. 71, 2016.

11) A. H. Mohammed, A. A. Abdulateef and I. A. Abdulateef, "Hyperledger, Ethereum and Blockchain Technology: A Short Overview," 2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), Ankara, Turkey, 2021, pp. 1-6, doi: 10.1109/HORA52670.2021.9461294.

12) Top Blockchains rating [Электронный ресурс] URL: https://dappradar.com/rankings/chains?resultsPerPage=50&sort=priceInFiat&order=desc (Дата обращения: 03.12.2024)

13) Внедрение блокчейна продолжается с прежней силой [Электронный ресурс] URL: https://www.binance.com/ru/square/post/1632987 (Дата обращения: 03.12.2024)

14) Ethereum Validator Queue [Электронный ресурс] URL: https://www.validatorqueue.com/ (Дата обращения: 03.12.2024)

15) Блокчейн-технологии в госуправлении. Мировой опыт [Электронный ресурс] URL: https://www.forbes.ru/tehnologii/343203-blokcheyn-tehnologii-v-gosupravlenii-mirovoy-opyt (Дата обращения: 03.12.2024)

16) Банк России. Цифровой рубль [Электронный ресурс] URL: https://www.cbr.ru/fintech/dr/ (Дата обращения: 03.12.2024)

17) Кошелек в сим-карте: как сотовые операторы используют блокчейн в работе [Электронный ресурс] URL: https://www.rbc.ru/technology_and_media/24/09/2024/66eefcca9a79472d62ba1f61 (Дата обращения: 03.12.2024)

18) СБЕР Лаборатория блокчейн [Электронный ресурс] URL: https://sberlabs.com/laboratories/laboratoriya-blockchain (Дата обращения: 03.12.2024)

19) СПбГУ Цифровой метр Блокчейн-платформа [Электронный ресурс] URL: https://www.tadviser.ru/a/805028 (Дата обращения: 03.12.2024)

20) Политика государства по развитию блокчейна в России [Электронный ресурс] URL: https://www.tadviser.ru/a/554478 (Дата обращения: 03.12.2024)

21) Носиров З.А., Фомичев В.М. Анализ блокчейн-технологии: основы архитектуры, примеры использования, перспективы развития, проблемы и недостатки // Системы управления, связи и безопасности. 2021. №2. URL: https://cyberleninka.ru/article/n/analiz-blokcheyn-tehnologii-osnovy-arhitektury-primery-ispolzovaniya-perspektivy-razvitiya-problemy-i-nedostatki (дата обращения: 10.02.2024).

22) Ethereum development documentation / [Электронный ресурс]. – Режим доступа: https://ethereum.org/en/developers/docs/ (дата обращения: 15.03.2024).

23) TON Blockchain documentation / [Электронный ресурс]. – Режим доступа: https://docs.ton.org/ (дата обращения: 20.03.2024).

24) Cardano ecosystem documentation / [Электронный ресурс]. – Режим доступа: https://docs.cardano.org/ (дата обращения: 25.03.2024).

25) S. Barac, I. Botički, G. Perković, V. Radošević and I. Terzić, "Cardano - What Is It and How to Start Working with It," *2023 46th MIPRO ICT and Electronics Convention (MIPRO)*, Opatija, Croatia, 2023, pp. 1727-1732, doi: 10.23919/MIPRO57284.2023.10159944.

26) A. V. Pomogalova, A. A. Martyniuk and K. E. Yesalov, "Key Features and Formation of Transactions in the Case of Using UTxO, EUTxO and Account Based Data Storage Models," *2022 International Conference on Modern Network Technologies (MoNeTec)*, Moscow, Russian Federation, 2022, pp. 1-7, doi: 10.1109/MoNeTec55448.2022.9960753.

27) G.-T. Nguyen and K. Kim, A survey about consensus algorithms used in blockchain, Journal of Information processing systems, vol. 14, no. 1, pp. 101-128, 2018.

28) Прикладные научные исследования в области создания сетей связи 2030, включая услуги телеприсутствия с сетевой поддержкой, и экспериментальная проверка решений при подготовке отраслевых кадров. Вторая очередь / Шестаков А.В., Громова Н.Н., Кучерявый А.Е., Маколкина М.А., Парамонов А.И., Выборнова А.И., Мутханна А.С.А., Матюхин А.Ю., Дунайцев Р.А., Владимиров С.С., Горбачева Л.С., Ворожейкина О.И., Паньков Б.О., Анваржонов Б.Н.У., Есалов К.Э., Помогалова А.В., Селиванов А.Е., Куликов Е.Ю., Сербин А.А., Попонин А.С. и др. - Отчет о НИР. Шифр «Телепорт-2030» Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации. 2022. Номер государственной регистрации: 122020100465-3

29) B. H. Swathi, M. S. Meghana and P. Lokamathe, An Analysis on Blockchain Consensus Protocols for Fault Tolerance, 2021 2nd International Conference for Emerging Technology (INCET), Belagavi, India, 2021, pp. 1-4, doi: 10.1109/INCET51464.2021.9456310.

30) S. Sharma, O. Sharma and J. Arora, Consensus Mechanisms Analysis: A Remedy for the Byzantine Generals Problem, 2023 3rd International Conference on Technological Advancements in Computational Sciences (ICTACS), Tashkent, Uzbekistan, 2023, pp. 674-678, doi: 10.1109/ICTACS59847.2023.10390006.

31) J. Pan, Z. Song and W. Hao, "Development in Consensus Protocols: From PoW to PoS to DPoS," 2021 2nd International Conference on Computer Communication and

Network Security (CCNS), Xining, China, 2021, pp. 59-64, doi: 10.1109/CCNS53852.2021.00020.

32) Воронин Д. Ю., Евстигнеев В. П. Обобщенный анализ пиковых нагрузок на коммутаторах инфокоммуникационной сети //Современная наука: проблемы, идеи, инновации. – 2019. – С. 52-59.

33) A. V. Pomogalova, E. A. Donskov, V. S. Elagin and A. G. Vladyko, "Methods for Evaluating Network Characteristics on Blockchain-V2X System Nodes," 2022 Systems of Signals Generating and Processing in the Field of on Board Communications, Moscow, Russian Federation, 2022, pp. 1-6, doi: 10.1109/IEEECONF53456.2022.9744263.

34) A. V. Pomogalova, E. A. Donskov, V. S. Elagin and A. G. Vladyko, "Blockchain Technologies for Validation of Priority Vehicles in ITS," 2022 Intelligent Technologies and Electronic Devices in Vehicle and Road Transport Complex (TIRVED), Moscow, Russian Federation, 2022, pp. 1-6, doi: 10.1109/TIRVED56496.2022.9965553.

35) A. V. Pomogalova, E. A. Donskov, V. S. Elagin and A. G. Vladyko, "Aspects of Data Transfer and Synchronization for Vulnerable Road Users Emergency Scenarios Based on Blockchain Technology in ITS," 2024 Systems of Signal Synchronization, Generating and Processing in Telecommunications (SYNCHROINFO), Vyborg, Russian Federation, 2024, pp. 1-6, doi: 10.1109/SYNCHROINFO61835.2024.10617888.

36) M. Laskin; A. Svistunova; A. Talavirya, “Evaluation of the daily intensity of traffic at the road exit toll plaza of the intraurban toll road,” System analysis in design and management. Part 2: Proceedings of the XXIV International Scientific and Educational-Practical Conference, 2020, doi:10.18720/SPBPU/2/id20-175.

37) Помогалова А.В., Разработка распределенной системы хранения, индексирования и выдачи цифровых документов.//В сборнике: Актуальные проблемы инфотелекоммуникаций в науке и образовании. сборник научных статей: в 4х томах. Санкт-Петербургский государственный университет

телекоммуникаций им. проф. М.А. Бонч-Бруевича. Санкт-Петербург, 2021. С. 624-628.

38) F. Yang, S. Wang, J. Li, Z. Liu and Q. Sun, "An overview of Internet of Vehicles," in China Communications, vol. 11, no. 10, pp. 1-15, Oct. 2014, doi: 10.1109/CC.2014.6969789.

39) G. Tripathi, M. Abdul Ahad and M. Sathiyanarayanan, "The Role of Blockchain in Internet of Vehicles (IoV): Issues, Challenges and Opportunities," 2019 International Conference on contemporary Computing and Informatics (IC3I), 2019, pp. 26-31, doi: 10.1109/IC3I46837.2019.9055613.

40) J. Alotaibi and L. Alazzawi, "SaFIoV: A Secure and Fast Communication in Fog-based Internet-of-Vehicles using SDN and Blockchain," 2021 IEEE International Midwest Symposium on Circuits and Systems (MWSCAS), 2021, pp. 334-339, doi: 10.1109/MWSCAS47672.2021.9531857.

41) K. Mershad and B. Said, "A Blockchain Model for Secure Communications in Internet of Vehicles," 2020 IEEE/ACS 17th International Conference on Computer Systems and Applications (AICCSA), 2020, pp. 1-6, doi: 10.1109/AICCSA50499.2020.9316498.

42) M. B. Mollah et al., "Blockchain for the Internet of Vehicles Towards Intelligent Transportation Systems: A Survey," in IEEE Internet of Things Journal, vol. 8, no. 6, pp. 4157-4185, 15 March15, 2021, doi: 10.1109/JIOT.2020.3028368.

43) Елагин В.С., Спиркина А.В., Владыко А.Г., Иванов Е.И., Помогалова А.В., Аптриева Е.А. Основные сетевые характеристики blockchain трафика и подходы к моделированию // T-Comm: Телекоммуникации и транспорт. 2020. Т. 14. № 4. С. 39–45. DOI:10.36724/2072-8735-2020-14-4-39-45.

44) L. Lei, P. Ma, C. Lan and L. Lin, "Continuous Distributed Key Generation on Blockchain Based on BFT Consensus," 2020 3rd International Conference on Hot

Information-Centric Networking (HotICN), Hefei, China, 2020, pp. 8-17, doi: 10.1109/HotICN50779.2020.9350834.

45) Плескунов, Михаил Александрович. Теория массового обслуживания : учебное пособие / М. А. Пле скунов ; М-во науки и высшего образования РФ, Урал. федер. ун-т. — Екатеринбург : Изд-во Урал. ун-та, 2022. — 264 с.

46) Черушева, Татьяна Вячеславовна. Теория массового обслуживания : учеб. пособие / Т. В. Черушева, Н. В. Зверовщикова. – Пенза : Изд-во ПГУ, 2021. – 224 с.

47) Кремер Н. Ш. Теория вероятностей и математическая статистика : учебник для вузов. 2-е изд., перераб. и доп. М. : ЮНИТИДАНА, 2004. 573 с. Алиев Т. И. Основы моделирования дискретных систем. СПб. : СПбГУ ИТМО, 2009. 363 с.

48) Есипов, Б. А. Методы исследования операций : учеб. пособие / Б. А. Есипов. — Санкт-Петербург : Лань, 2013. — 304 с. — ISBN 978-58114-0917-4.

49) Матюшенко Сергей Иванович Анализ двухканальной системы массового обслуживания ограниченной ёмкости с буфером переупорядочивания и с распределениями фазового типа // Discrete and Continuous Models and Applied Computational Science. 2010. №4. URL: https://cyberleninka.ru/article/n/analiz-dvuhkanalnoy-sistemy-massovogo-obsluzhivaniya-ogranichennoy-yomkosti-s-buferom-pereuporyadochivaniya-i-s-raspredeleniyami (дата обращения: 07.04.2024).

50) Плужников В. Л., Домников А. С. Преобразование Лапласа-Стилтьеса времени соединения двух таблиц в параллельной системе баз данных // Машиностроение и компьютерные технологии. 2012. №06. URL: https://cyberleninka.ru/article/n/preobrazovanie-laplasa-stiltiesa-vremeni-soedineniya-dvuh-tablits-v-parallelnoy-sisteme-baz-dannyh (дата обращения: 15.02.2024).

51) Тарасов, В. Н. Исследование систем массового обслуживания с гиперэкспоненциальными входными распределениями / В. Н. Тарасов // Проблемы передачи информации. — 2016. — № 1. — С. 16-26.

52) Агапова Ирина Степановна Использование теории неоднородных марковских процессов при исследовании вероятностных характеристик некоторых технологических процессов // Радиоэлектроника и информатика. 2001. №4 (17). URL: https://cyberleninka.ru/article/n/ispolzovanie-teorii-neodnorodnyh-markovskih-protsessov-pri-issledovanii-veroyatnostnyh-harakteristik-nekotoryh-tehnologicheskih (дата обращения: 17.05.2024).

53) Timofeev G.A., Bolshakov V.N., Anisimov A.R., Skomorokhina E.R., Chikenev S.D. Analysis of Blockchain Technology Possibilities in Data Management. Innovatsii i investitsii. 2023;5:174–178. (in Russ.) EDN:XVVEU

54) Garipov R.I., Maximova N.N. Analysis of Methodical Approaches to Evaluating Blockchain Efficiency. Upravlenie v sovremennykh sistemakh. 2020;1(25):13–17. (in Russ.) EDN:URBTFO

55) Гарипов Р.И., Максимова Н.Н. Анализ методических подходов к оценке эффективности блокчейна // Управление в современных системах. 2020. № 1(25). С. 13–17. URL: EDN:URBTFO

56) Владимиров С.С., Владыко А.Г., Караваев Д.А., Помогалова А.В., Степанов А.Б. Испытательный стенд для исследования сети SD-IOV с технологией LORA / В сборнике: Модернизация информационной инфраструктуры для сетей 5G/IMT 2020 и для других перспективных технологий в интересах трансформации регионов РОСИНФОКОМ-2019. Сборник научных статей. 2019. С. 21-30

57) Донсков Е.А., Котенко И.В., Помогалова А.В. Анализ отказоустойчивости узла блокчейн-сети при моделировании суточной нагрузки на узел транспортной развязки. / В книге: Региональная информатика (РИ-2022). Юбилейная XVIII Санкт-Петербургская международная конференция. Материалы конференции. Санкт-Петербург, 2022. С. 153-155.

58) ITU-T H.550-2017 Architecture and functional entities of vehicle gateway platforms.

59) ITU-T F.749.4-2021 Use cases and requirements for multimedia communication enabled vehicle systems using artificial intelligence.

60) M. Bachmann, L. H. Acosta, J. Götz, D. Reinhardt and K. David, "Collision Avoidance for Vulnerable Road Users: Privacy versus Survival?," NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium, Budapest, Hungary, 2022, pp. 1-6, doi: 10.1109/NOMS54207.2022.9789710.

61) Pomogalova A., Sazonov D., Donskov E., Borodin A., Kirichek R. (2021) Identification Method for Endpoint Devices on Low-Power Wide-Area Networks Using Digital Object Architecture with Blockchain Technology Integration. In: Vishnevskiy V.M., Samouylov K.E., Kozyrev D.V. (eds) Distributed Computer and Communication Networks: Control, Computation, Communications. DCCN 2021. Lecture Notes in Computer Science, vol 13144. Springer, Cham. https://doi.org/10.1007/978-3-030-92507-9_10

62) S. Pandey, G. Ojha, B. Shrestha and R. Kumar, "BlockSIM: A practical simulation tool for optimal network design, stability and planning," 2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), Seoul, Korea (South), 2019, pp. 133-137, doi: 10.1109/BLOC.2019.8751320.

63) C. Faria and M. Correia, "BlockSim: Blockchain Simulator," 2019 IEEE International Conference on Blockchain (Blockchain), Atlanta, GA, USA, 2019, pp. 439-446, doi: 10.1109/Blockchain.2019.00067.

64) I. Rashdan and S. Sand, "Link-Level Performance of Vehicle-to-Vulnerable Road Users Communication Using Realistic Channel Models," 2024 18th European Conference on Antennas and Propagation (EuCAP), Glasgow, United Kingdom, 2024, pp. 1-5, doi: 10.23919/EuCAP60739.2024.10501335.

65) Помогалова А.В., Сазонов Д.Д., Бородин А.С., Киричёк Р.В. Идентификация устройств узкополосных беспроводных сетей связи дальнего действия на основе

архитектуры цифровых объектов с применением технологии Blockchain / Электросвязь. 2021. № 12. С. 21-26.

66) Andrianova E.G., Golovin O.L. Conceptual Aspects of Buiding a Trusted Heterogeneous Blockchain-Environments of a New Technological Structure. IT-Standard. 2017;3:1–6. (in Russ.)

67) M. Rupp and L. Wischhof, "Evaluation of the Effectiveness of Vulnerable Road User Clustering in C-V2X Systems," 2023 IEEE International Conference on Omni-layer Intelligent Systems (COINS), Berlin, Germany, 2023, pp. 1-5, doi: 10.1109/COINS57856.2023.10189204.

68) Кучерявый Андрей Евгеньевич, Окунева Дарина Владимировна, Парамонов Александр Иванович, Хоанг Фыок Ньян Методы распределения трафика в гетерогенной сети Интернета вещей высокой плотности // Труды учебных заведений связи. 2024. №2. URL: https://cyberleninka.ru/article/n/metody-raspredeleniya-trafika-v-geterogennoy-seti-interneta-veschey-vysokoy-plotnosti (дата обращения: 10.09.2024).

69) Райгородский А. М. Модели случайных графов и их применения // Труды МФТИ. 2010. №4. URL: https://cyberleninka.ru/article/n/modeli-sluchaynyh-grafov-i-ih-primeneniya (дата обращения: 02.05.2024).

70) Берновски М. М., Кузюрин Н. Н. Случайные графы, модели и генераторы безмасштабных графов // Труды ИСП РАН. 2012. №. URL: https://cyberleninka.ru/article/n/sluchaynye-grafy-modeli-i-generatory-bezmasshtabnyh-grafov (дата обращения: 20.06.2024).

71) Прикладные научные исследования в области создания сетей связи 2030, включая услуги телеприсутствия с сетевой поддержкой, и экспериментальная проверка решений при подготовке отраслевых кадров / Брусиловский С.А., Нестеров А.А., Кучерявый А.Е., Федоров С.Л., Громова Н.Н., Аникевич Е.А., Копылов Д.А., Барбанель Е.С., Швидкий А.А., Федоров А.С., Парамонов А.И.,

Мутханна А.С.А., Волков А.Н., Елагин В.С., Ушаков И.А., Прасолов А.А., Антипин Б.М., Березкин А.А., Вивчарь Р.М., Кукунин Д.С., Мендельсон М.А., Помогалова А.В., Егоров В.А., Бухинник А.Ю., Ворожейкина О.И., Пупцев Р.И., Макарова Е.В., Виноградов Е.М., Гаврилова А.Н., Рощинский Р.С., Ишутина О.Ю., Бойков М.С., Точилов В.Н., Григорьев С.В., Александров И.В., Шаброва Е.В., Рожков М.А., Аникевич У.М., Дмитриева Ю.С. Отчет о НИР. Шифр «Технология-2030». 2023. Номер государственной регистрации: 123060900012-6.

72) Свидетельство о государственной регистрации программы для ЭВМ 2024662552. Модуль менеджера задач для платформы коммуникаций на базе приватного EVM блокчейна / Шаляпин С.О., Бакатов В.Н., Ибрагимов Р.Р., Искра И.И., Мурашкин Н.А., Новиков С.С., Помогалова А.В., Фроловнин А.В. - Заявка №2024618829. Дата поступления 18.04.2024. Зарегистрировано в Реестре программ для ЭВМ 29.05.2024. Правообладатель: Общество с ограниченной ответственностью «Естественный Интеллект».

73) Свидетельство о государственной регистрации программы для ЭВМ 2022684461. Программный модуль заключения цифровых сделок для применения в мессенджерах мгновенных сообщений / Бакатов В.Н., Мартынюк А.А., Помогалова А.В., Есалов К.Э., Новиков С.С., Мурашкин Н.А., Искра И.И., Сербин А.А. - Заявка № 2022684047. Дата поступления 07.12.2022. Зарегистрировано в Реестре программ для ЭВМ 14.12.2022. Правообладатель: Федеральное государственное бюджетное образовательное учреждение высшего образования «Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича» (СПбГУТ).

74) Свидетельство о государственной регистрации программы для ЭВМ 2022663086. Система фиксации, хранения и индексации данных о документах с возможностью генерации цифровой версии документа на базе технологии блокчейн / Помогалова А.В., Помогалов В.А., Донсков Е.А. - Заявка № 2022613929. Дата поступления 14.03.2022. Зарегистрировано в Реестре программ для ЭВМ 11.07.2022. Правообладатель: Помогалова А.В.

75) P. R. Nair and D. R. Dorai, "Evaluation of Performance and Security of Proof of Work and Proof of Stake using Blockchain," 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), Tirunelveli, India, 2021, pp. 279-283, doi: 10.1109/ICICV50876.2021.9388487.

76) S. Yan, "Analysis on Blockchain Consensus Mechanism Based on Proof of Work and Proof of Stake," 2022 International Conference on Data Analytics, Computing and Artificial Intelligence (ICDACAI), Zakopane, Poland, 2022, pp. 464-467, doi: 10.1109/ICDACAI57211.2022.00098.

77) M. A. Majumdar, M. Monim and M. M. Shahriyer, "Blockchain based Land Registry with Delegated Proof of Stake (DPoS) Consensus in Bangladesh," 2020 IEEE Region 10 Symposium (TENSYMP), Dhaka, Bangladesh, 2020, pp. 1756-1759, doi: 10.1109/TENSYMP50017.2020.9230612.

78) J. Mišić, V. B. Mišić and X. Chang, "Delegated Proof of Stake Consensus with Mobile Voters and Multiple Entry PBFT Voting," GLOBECOM 2022 - 2022 IEEE Global Communications Conference, Rio de Janeiro, Brazil, 2022, pp. 6253-6258, doi: 10.1109/GLOBECOM48099.2022.10001051.

79) S. M. S. Saad, R. Z. R. M. Radzi and S. H. Othman, "Comparative Analysis of the Blockchain Consensus Algorithm Between Proof of Stake and Delegated Proof of Stake," 2021 International Conference on Data Science and Its Applications (ICoDSA), Bandung, Indonesia, 2021, pp. 175-180, doi: 10.1109/ICoDSA53588.2021.9617549.

80) A. Pal and K. Kant, "DC-PoET: Proof-of-Elapsed-Time Consensus with Distributed Coordination for Blockchain Networks," 2021 IFIP Networking Conference (IFIP Networking), Espoo and Helsinki, Finland, 2021, pp. 1-9, doi: 10.23919/IFIPNetworking52078.2021.9472787.

81) Корепанов Р.С., Помогалова А.В. Разработка платформы анализа данных блокчейн-сети Ethereum. // В сборнике: Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2023). Сборник научных

статей XII Международной научно-технической и научно-методической конференции. В 4-х томах. Под редакцией С.И. Макаренк, сост. В.С. Елагин, Е.А. Аникевич. Санкт-Петербург, 2023. С. 738-741.

82) Бакатов В.Н., Исхаков Э.Э., Помогалова А.В. Подходы к оптимизации индексаторов - ключевого элемента будущего систем распределенного реестра. //В сборнике: Актуальные проблемы инфотелекоммуникаций в науке и образовании. Сборник научных статей XIII Международной научно-технической и научно-методической конференции в 4 т.. Санкт-Петербург, 2024. С. 84-86.

ПРИЛОЖЕНИЕ А

Программный код модуля мониторинга на языке Python3

```
import time

from scapy.all import sniff

import statistics

import threading

import queue


logging.basicConfig(filename='network_traffic.log', level=logging.INFO)


# Настройки

capture_time = 30  # Время для анализа в секундах

report_interval = 1  # Интервал отчетов в секундах


# Хранение данных

packet_queue = queue.Queue()  # Очередь для хранения размеров пакетов и
временных меток

latency_data = queue.Queue()  # Очередь для хранения задержек

start_time = time.time()  # Время начала захвата

last_packet_time = None  # Время получения последнего пакета

total_transactions = 0  # Общее количество транзакций


# Функция для обработки пакетов

def packet_callback(packet):

    global last_packet_time, total_transactions


    # Получаем размер пакета

    packet_size = len(packet)
```

```
    current_time = time.time()

    # Добавляем размер пакета в очередь
    packet_queue.put((packet_size, current_time))

    # Увеличиваем счетчик транзакций (предполагаем, что каждый пакет - это транзакция)
    total_transactions += 1

    # Вычисляем задержку, если это не первый пакет
    if last_packet_time is not None:
        latency = current_time - last_packet_time
        latency_data.put(latency)

    # Обновляем время получения последнего пакета
    last_packet_time = current_time

# Функция для вычисления и вывода метрик
def report_metrics():
    global packet_queue, latency_data, total_transactions
    total_size = 0  # Общий размер пакетов для расчета пропускной способности
    total_packets = 0  # Общее количество пакетов

    while True:
        time.sleep(report_interval)

        # Проверяем, прошло ли время захвата
        elapsed_time = time.time() - start_time

        # Копируем данные из очереди
```

```
        packet_sizes = []
        packet_timestamps = []
        while not packet_queue.empty():
            size, timestamp = packet_queue.get()
            packet_sizes.append(size)
            packet_timestamps.append(timestamp)

            # Суммируем размеры пакетов для пропускной способности
            total_size += size
            total_packets += 1

        # Удаляем старые данные (старше 30 секунд)
        current_time = time.time()
        valid_indices = [i for i, ts in enumerate(packet_timestamps) if
current_time - ts <= 30]
        packet_sizes = [packet_sizes[i] for i in valid_indices]
        latency_data_list = list(latency_data.queue)  # Копируем данные
задержек

        # Пропускная способность (размер пакетов за интервал)
        throughput = (total_size * 8) / (report_interval * 1024 * 1024)  #
Мбит/с

        # Средний размер пакета
        avg_size = statistics.mean(packet_sizes) if packet_sizes else 0
        avg_size_mb = avg_size / (1024 * 1024)  # Перевод в Мбит

        # Задержка и джиттер
        avg_latency = statistics.mean(latency_data_list) if
latency_data_list else 0
        avg_jitter = statistics.pstdev(latency_data_list) if
len(latency_data_list) > 1 else 0
```

```
        # Определение характеристик блокчейна
        transactions_per_second = total_transactions / elapsed_time if
elapsed_time > 0 else 0
        avg_confirmation_time = avg_latency  # Время подтверждения может
быть связано со средней задержкой


        logging.info(f"Средний размер пакета: {avg_size_mb:.2f} Мбит")
        logging.info(f"Пропускная способность: {throughput:.2f} Мбит/с")
        logging.info(f"Средняя задержка: {avg_latency * 1000:.2f} мс")  #
Перевод в мс
        logging.info(f"Джиттер: {avg_jitter * 1000:.2f} мс")  # Перевод в
мс
        logging.info(f"Общее количество транзакций: {total_transactions}")
        logging.info(f"Транзакций в секунду:
{transactions_per_second:.2f}")
        logging.info(f"Среднее время подтверждения: {avg_confirmation_time
* 1000:.2f} мс")  # Перевод в мс
        logging.info("-------------------------------------------------")


# Запуск захвата пакетов и отчетов
if __name__ == "__main__":
    # Запуск потока для отчетов
    report_thread = threading.Thread(target=report_metrics)
    report_thread.start()

    # Захват пакетов
    sniff(prn=packet_callback)
```

Программный код модуля принятия решения на языке Python3

```
import pandas as pd
```

```
import logging
import time
import os
from web3 import Web3

# Настройки
LOG_FILE = 'network_traffic.log'
DECISION_TABLE_FILE = 'table_decision.csv'
CURRENT_ALGORITHM = "Алгоритм_1"

# Подключение к ноде Ethereum
WEB3_PROVIDER = "http://localhost:8545"  # Укажите адрес вашей ноды
web3 = Web3(Web3.HTTPProvider(WEB3_PROVIDER))

# Функция для получения заполненности memory pool через Web3
def get_memory_pool_size():
    try:
        # Получаем информацию о транзакциях в памяти (pending transactions)
        pending_txs = web3.eth.getBlock('pending')['transactions']
        return len(pending_txs)  # Возвращаем количество транзакций в
памяти
    except Exception as e:
        logging.error(f"Ошибка при получении размера memory pool: {e}")
        return 0  # Возвращаем 0 в случае ошибки

# Функция для чтения лог-файла и извлечения метрик
def read_metrics_from_log(log_file):
    metrics = {
        'avg_packet_size': None,
        'throughput': None,
```

```
        'avg_latency': None,
        'avg_jitter': None,
        'total_transactions': None,
        'transactions_per_second': None,
        'avg_confirmation_time': None,
        'memory_pool': None
    }

    with open(log_file, 'r') as file:
        for line in file:
            if "Средний размер пакета" in line:
                metrics['avg_packet_size'] =
float(line.split(":")[1].strip().split()[0])
            elif "Пропускная способность" in line:
                metrics['throughput'] =
float(line.split(":")[1].strip().split()[0])
            elif "Средняя задержка" in line:
                metrics['avg_latency'] =
float(line.split(":")[1].strip().split()[0]) / 1000
            elif "Джиттер" in line:
                metrics['avg_jitter'] =
float(line.split(":")[1].strip().split()[0]) / 1000
            elif "Общее количество транзакций" in line:
                metrics['total_transactions'] =
int(line.split(":")[1].strip())
            elif "Транзакций в секунду" in line:
                metrics['transactions_per_second'] =
float(line.split(":")[1].strip().split()[0])
            elif "Среднее время подтверждения" in line:
                metrics['avg_confirmation_time'] =
float(line.split(":")[1].strip().split()[0]) / 1000

    return metrics
```

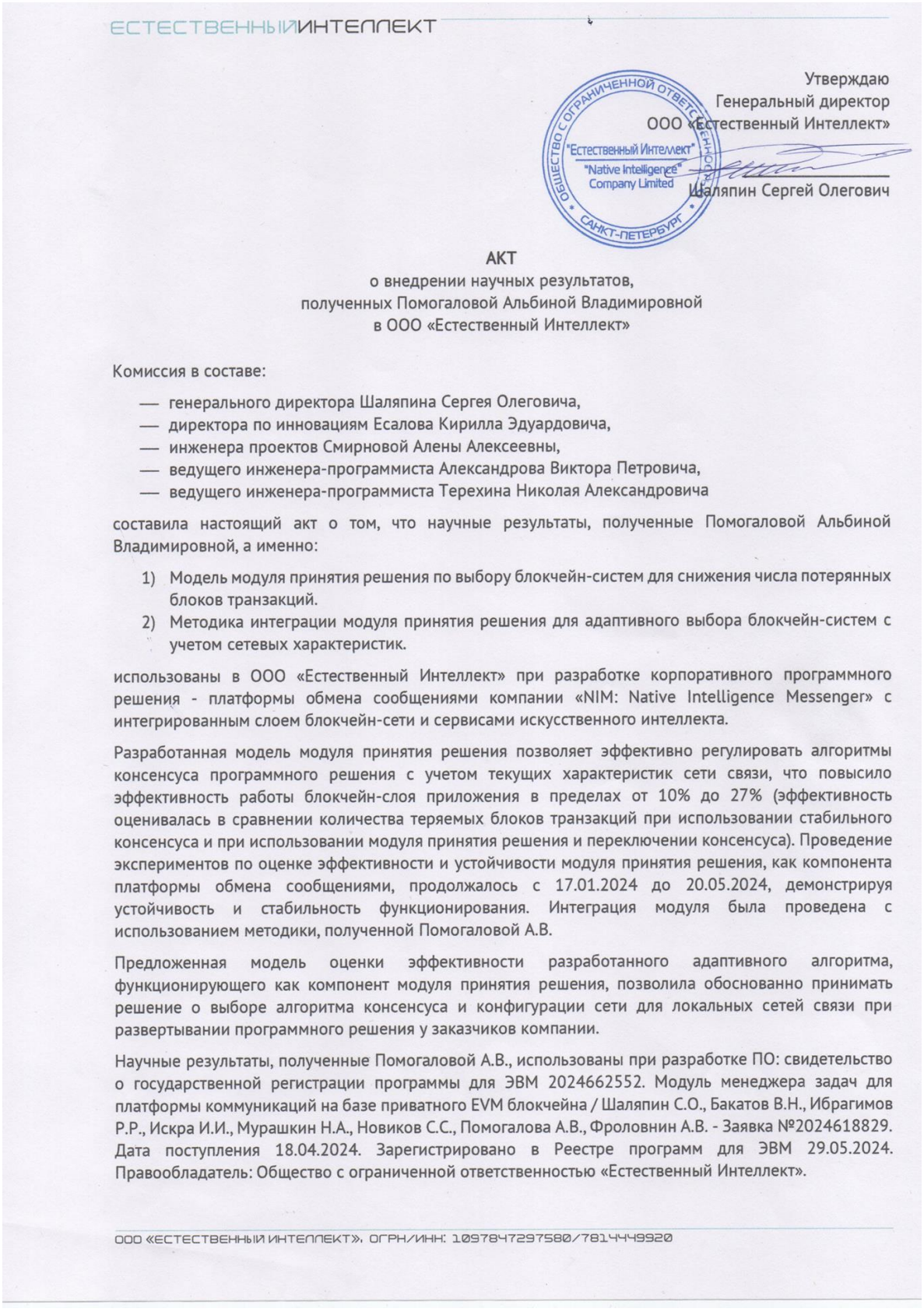
```
# Функция для чтения таблицы принятия решений
def read_decision_table(decision_table_file):
    decision_table = pd.read_csv(decision_table_file)
    return decision_table

# Функция для определения необходимости переключения алгоритма
def should_switch_algorithm(metrics, decision_table):
    for index, row in decision_table.iterrows():
        if (metrics['avg_packet_size'] >= row['avg_packet_size_min'] and
            metrics['avg_packet_size'] <= row['avg_packet_size_max'] and
            metrics['throughput'] >= row['throughput_min'] and
            metrics['throughput'] <= row['throughput_max'] and
            metrics['avg_latency'] <= row['avg_latency_max'] and
            metrics['avg_jitter'] <= row['avg_jitter_max'] and
            metrics['transactions_per_second'] >=
row['transactions_per_second_min'] and
            metrics['memory_pool'] >= row['memory_pool_min'] and
            metrics['memory_pool'] <= row['memory_pool_max']):
            return True, row['algorithm_name']
    return False, None

# Функция для поиска более подходящего алгоритма
def find_best_algorithm(metrics, decision_table):
    best_algorithm = None
    best_match_count = 0

    for index, row in decision_table.iterrows():
        match_count = 0

        if (metrics['avg_packet_size'] >= row['avg_packet_size_min'] and
```

```
            metrics['avg_packet_size'] <= row['avg_packet_size_max']):
                match_count += 1
        if (metrics['throughput'] >= row['throughput_min'] and
            metrics['throughput'] <= row['throughput_max']):
                match_count += 1
        if (metrics['avg_latency'] <= row['avg_latency_max']):
                match_count += 1
        if (metrics['avg_jitter'] <= row['avg_jitter_max']):
                match_count += 1
        if (metrics['transactions_per_second'] >=
row['transactions_per_second_min']):
                match_count += 1
        if (metrics['memory_pool'] >= row['memory_pool_min'] and
            metrics['memory_pool'] <= row['memory_pool_max']):
                match_count += 1

        if match_count > best_match_count:
                best_match_count = match_count
                best_algorithm = row['algorithm_name']

    return best_algorithm

# Функция для переключения алгоритма консенсуса
def switch_consensus_algorithm(new_algorithm):
    global CURRENT_ALGORITHM
    logging.info(f"Переключение алгоритма консенсуса с {CURRENT_ALGORITHM}
на {new_algorithm}...")
    CURRENT_ALGORITHM = new_algorithm

# Основная логика
if __name__ == "__main__":
```

```
    logging.basicConfig(level=logging.INFO)


    while True:
        # Получаем размер memory pool
        memory_pool_size = get_memory_pool_size()
        logging.info(f"Размер memory pool: {memory_pool_size}")


        # Чтение метрик из лог-файла
        metrics = read_metrics_from_log(LOG_FILE)
        metrics['memory_pool'] = memory_pool_size  # Обновляем метрики с
размером memory pool
        logging.info(f"Извлеченные метрики: {metrics}")


        # Чтение таблицы принятия решений
        decision_table = read_decision_table(DECISION_TABLE_FILE)


        # Проверка необходимости переключения алгоритма
        is_current_valid, current_algorithm =
should_switch_algorithm(metrics, decision_table)


        if not is_current_valid:
            new_algorithm = find_best_algorithm(metrics, decision_table)
            if new_algorithm and new_algorithm != CURRENT_ALGORITHM:
                switch_consensus_algorithm(new_algorithm)
            else:
                logging.info("Текущий алгоритм консенсуса все еще
актуален.")
        else:
            logging.info("Текущий алгоритм консенсуса все еще актуален.")


        time.sleep(1)  # Ждем 1 секунду перед следующим чтением
```

ПРИЛОЖЕНИЕ Б

Акт о внедрении от компании ООО «Естественный Интеллект»



ЕСТЕСТВЕННЫЙИНТЕЛЛЕКТ

Утверждаю
Генеральный директор
ООО «Естественный Интеллект»
Шаляпин Сергей Олегович

АКТ
о внедрении научных результатов,
полученных Помогаловой Альбиной Владимировной
в ООО «Естественный Интеллект»

Комиссия в составе:

- генерального директора Шаляпина Сергея Олеговича,
- директора по инновациям Есалова Кирилла Эдуардовича,
- инженера проектов Смирновой Алены Алексеевны,
- ведущего инженера-программиста Александрова Виктора Петровича,
- ведущего инженера-программиста Терехина Николая Александровича

составила настоящий акт о том, что научные результаты, полученные Помогаловой Альбиной Владимировной, а именно:

1) Модель модуля принятия решения по выбору блокчейн-систем для снижения числа потерянных блоков транзакций.
2) Методика интеграции модуля принятия решения для адаптивного выбора блокчейн-систем с учетом сетевых характеристик.

использованы в ООО «Естественный Интеллект» при разработке корпоративного программного решения - платформы обмена сообщениями компании «NIM: Native Intelligence Messenger» с интегрированным слоем блокчейн-сети и сервисами искусственного интеллекта.

Разработанная модель модуля принятия решения позволяет эффективно регулировать алгоритмы консенсуса программного решения с учетом текущих характеристик сети связи, что повысило эффективность работы блокчейн-слоя приложения в пределах от 10% до 27% (эффективность оценивалась в сравнении количества теряемых блоков транзакций при использовании стабильного консенсуса и при использовании модуля принятия решения и переключении консенсуса). Проведение экспериментов по оценке эффективности и устойчивости модуля принятия решения, как компонента платформы обмена сообщениями, продолжалось с 17.01.2024 до 20.05.2024, демонстрируя устойчивость и стабильность функционирования. Интеграция модуля была проведена с использованием методики, полученной Помогаловой А.В.

Предложенная модель оценки эффективности разработанного адаптивного алгоритма, функционирующего как компонент модуля принятия решения, позволила обоснованно принимать решение о выборе алгоритма консенсуса и конфигурации сети для локальных сетей связи при развертывании программного решения у заказчиков компании.

Научные результаты, полученные Помогаловой А.В., использованы при разработке ПО: свидетельство о государственной регистрации программы для ЭВМ 2024662552. Модуль менеджера задач для платформы коммуникаций на базе приватного EVM блокчейна / Шаляпин С.О., Бакатов В.Н., Ибрагимов Р.Р., Искра И.И., Мурашкин Н.А., Новиков С.С., Помогалова А.В., Фроловнин А.В. - Заявка №2024618829. Дата поступления 18.04.2024. Зарегистрировано в Реестре программ для ЭВМ 29.05.2024. Правообладатель: Общество с ограниченной ответственностью «Естественный Интеллект».

ООО «ЕСТЕСТВЕННЫЙ ИНТЕЛЛЕКТ», ОГРН/ИНН: 1097847297580/7814449920

Председатель комиссии:

Генеральный директор _____ С.О. Шаляпин 20.11.24

Члены комиссии:

Директор по инновациям _____ К.Э. Есалов 20.11.24

Инженер проектов _____ А.А. Смирнова 20.11.24

Ведущий инженер-программист _____ В.П. Александров 20.11.24

Ведущий инженер-программист _____ Н.А. Терехин 20.11.24

Акт о внедрении от компании ООО «ЮбиТел»

ООО «ЮбиТел»
197046, Россия, Санкт-Петербург,
ул.Чапаева, д. 9, лит А, пом. 1-Н, пом.25
тел. (812) 600-7920, www.ubitel.ru

УТВЕРЖДАЮ
Директор ООО «ЮбиТел»

_____ Д.В. Осипов

«28» ноября 2024 г.

АКТ
о внедрении научных результатов,
полученных Помогаловой Альбиной Владимировной,
в ООО «ЮбиТел»

Комиссия в составе:

Председатель комиссии – технический директор, Афонин Иван Григорьевич

Члены комиссии: -

- заместитель технического директора, Кучерявый Евгений Андреевич, доктор технических наук;
- инженер -программист 1 категории, Тучкевич Артем Никитович

настоящим актом подтверждает, что научные результаты, полученные Помогаловой Альбиной Владимировной, применены в ООО «ЮбиТел» при проектировании программного обеспечения «Моделирование сетей 3GPP/IEEE системного уровня с интегрированными библиотеками для построения систем искусственного интеллекта».

При разработке названного программного обеспечения использованы следующие новые научные результаты, полученные Помогаловой А.В.:

- Набор пороговых значений сетевых и аппаратных характеристик для эффективного оперирования информационными потоками блокчейн, демонстрирующие диапазоны наибольшей эффективности алгоритмов консенсуса.

Внедрение подтвердило, что число сетевых характеристик для эффективного выбора блокчейн-систем в сетях связи достаточно и не превышает 10 штук с учетом реальных диапазонов их пороговых значений.

Применение результатов работ позволило расширить функциональные возможности системы с учетом сетевых характеристик и значений аппаратного обеспечения, используемого для исполнения проектируемого ПО

Председатель комиссии:

Технический директор И.Г. Афонин

Члены комиссии -

Заместитель технического директора Е.А. Кучерявый

Инженер -программист 1 категории А.Н.Тучкевич

Общество с ограниченной ответственностью «ЮбиТел» «UbiTel» Co. Ltd. Санкт-Петербург

Акт о внедрении от Федерального государственного бюджетного образовательного учреждения высшего образования «Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича»

МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ,
СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ
ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»
(СПбГУТ)

Юридический адрес: набережная реки Мойки,
д. 61, литера А, Санкт-Петербург, 191186

Почтовый адрес: пр. Большевиков, д. 22, корп. 1,
Санкт-Петербург, 193232
Тел.(812) 3263156, Факс: (812) 3263159
http://sut.ru
E-mail: rector@sut.ru
ОКПО 01179934 ОГРН 1027809197635
ИНН 7808004760 КПП 784001001
ОКТМО 40909000

25.11.2024 № 10/54
на №_____ от _____

Утверждаю

И.о. проректора по
научной работе

Рабин
Алексей Владимирович

Акт

о внедрении научных результатов,

полученных Помогаловой Альбиной Владимировной

Комиссия в составе Елагина В.С., и.о. декана факультета инфокоммуникационных сетей и систем, Дунайцева Р.А., доцента кафедры сетей связи и передачи данных, Гольдштейна А.Б., доцента кафедры инфокоммуникационных сетей и Ворожейкиной О.И., заведующей лабораторией кафедры сетей связи и передачи данных составила настоящий акт о том, что научные результаты, полученные Помогаловой Альбиной Владимировной, использованы:

1) При чтении лекций и проведении практических занятий по курсу «Сети связи и системы коммутации» (Рабочая Программа № 24.05/14-Д, утверждена Первым проректором-проректором по учебной работе А.В. Абилов 02.04.2024), разделы программы:
   - Раздел 11. Понятие о качестве обслуживания.
   - Раздел 16. Система с повторными вызовами.
   - Раздел 18. Оценка надежности сетевых элементов.
2) При выполнении ПНИ и отчета о НИР на тему «Прикладные научные исследования в области создания, развития и нормативного регулирования сетей связи, цифровых и перспективных технологий, подготовки отраслевых кадров на период до 2030 года с учетом импортозамещения и необходимости преодоления санкционных ограничений», шифр «Цифры-2030», дата начала 01.01.2024, дата окончания 31.12.2024.
3) При выполнении ПНИ и отчета о НИР на тему «Прикладные научные исследования в области создания сетей связи 2030, включая услуги телеприсутствия с сетевой поддержкой, и экспериментальная проверка решений при подготовке отраслевых кадров», шифр «Технология-2030», дата

начала 19.01.2023, дата окончания 29.12.2023, номер государственной регистрации: 123060900012-6.

В рамках выполнения ПНИ и в указанных дисциплинах используются следующие новые научные результаты, полученные Помогаловой Альбиной Владимировной в диссертационной работе:

– модель нового адаптивного алгоритма для совершенствования управления информационными потоками блокчейн на сетях связи, что позволяет выбирать наиболее эффективный алгоритм консенсуса с учетом ряда сетевых параметров,

– модель оценки эффективности нового адаптивного алгоритма на сети связи с учетом конфигурации сети и среднего времени генерации новых блоков,

– границы пороговых значений сетевых характеристик для эффективного оперирования алгоритмов консенсуса.

И.о. декана факультета ИКСС,
канд. техн. наук, доцент — В.С. Елагин

Доцент кафедры ССиПД,
канд. техн. наук — Р.А. Дунайцев

Профессор кафедры ИКС,
док. техн. наук — А.Б. Гольдштейн

Зав. лабораторией кафедры ССиПД — О.И. Ворожейкина