УДК 004.052.42 DOI: 10.17587/prin.13.200-208

И. А. Федотов[1], аспирант, ivan.fedotov@phystech.edu,
А. С. Хританков[2], канд. физ.-мат. наук, доц., akhritankov@hse.ru,
М. Д. Обидаре[1], студент, obidarefolu1@gmail.com
[1] Московский физико-технический институт,
[2] Национальный исследовательский университет "Высшая школа экономики", Москва

Автоматическая верификация многосторонних соглашений и планирование отправки сообщений в системах распределенного реестра

Многосторонние соглашения используются в системах распределенного реестра и блокчейн-сетях для согласования изменений в системе. Если один из участников сети предлагает транзакцию на запись, то сначала ее должны подтвердить определенные участники сети. Многостороннее соглашение, или консенсус, определяет состав этих участников. На основе предыдущих ответов можно посчитать вероятность подтверждения транзакции для каждого из участников. В настоящей работе предложен метод статистической проверки моделей для определения вероятности того, что консенсус будет достигнут. Отправка запросов на подтверждение может требовать дополнительных затрат. Кроме отмеченной вероятности вычислено математическое ожидание числа сообщений, которые прошли в сети до достижения консенсуса. Приведена модель или несколько моделей консенсуса в виде марковской цепи с различными стратегиями отправки сообщений. На основе алгоритмов построения модели и спецификации разработано инструментальное средство анализа консенсуса и отправки сообщений на подтверждение.

Ключевые слова: *консенсус, блокчейн, верификация, статистическая проверка моделей, марковская цепь*

Введение

Технология распределенного реестра получила в последнее время широкое распространение. Хранилища на основе технологии блокчейн используются в различных областях: юриспруденция; отслеживание активов; финансовый сектор и ряд других [1]. Блокчейн удобен при использовании в системах, где несколько заинтересованных сторон, не доверяющих друг другу, должны достигнуть соглашения. Консенсус утверждает состав участников, которые должны подтвердить транзакцию, чтобы она считалась валидной для всех участников сети [2].

Каждый участник сети может либо подтвердить, либо отклонить транзакцию. Так как ответ каждого участника заранее неизвестен, то процесс подтверждения транзакции является вероятностным. На основе полученных ответов от каждого участника консенсуса можно вычислить вероятность того, что в следующем раунде подтверждения участник консенсуса подтвердит транзакцию. Условия консенсуса могут задаваться в виде логической формулы. Таким образом условия консенсуса задаются во фреймворке Hyperledger Fabric (HLF), который является одной из самых используемых платформ для построения блокчейн-систем [3]. Предложенные в статье алгоритмы и инструментальные средства могут использоваться для верификации правил одобрения HLF.

Уязвимости в блокчейн-системах приводили к значительным потерям [4]. В частности, уязвимости в принятых правилах одобрений HLF являлись причиной финансовых потерь [5]. Верификация многосторонних соглашений позволяет выявить уязвимости до развертывания соглашений в сети.

В исследовании, результаты которого представлены в настоящей статье, представлено построение модели консенсуса в виде марковской цепи. Условия консенсуса передаются в виде вероятностной линейной временной логики (*probabilistic linear temporal logic*, pLTL) [6]. Использован метод статистической проверки моделей [7] для определения вероятности достижения консенсуса. Использованы различные уникальные комбинации последовательностей отправки сообщений на подтверждение. В целом, результаты предлагаемого авторами статьи подхода состоят в следующем:

- предложен алгоритм построения марковской цепи и спецификации для консенсуса, заданного в виде pLTL-формулы;

- разработан модуль автоматического построения модели и проверки спецификации;
- предложен алгоритм отправки сообщений на подтверждение на основе выбранной модели марковской цепи;
- разработан модуль автоматической отправки сообщений на подтверждение;
- представлена возможность интеграции разработанного инструментального средства с фреймворком Hyperledger Fabric.

Смежные работы

Формальные методы верификация и тестирования применяются в процессе разработки и отладки блокчейн-систем [8]. Верификация с помощью метода статистической проверки моделей хорошо изучена, и для этого могут использоваться различные инструментальные средства [9]. В недавних исследованиях [10, 11] авторы продемонстрировали возможность использования статистической проверки моделей для верификации блокчейн-систем. Модель в виде марковской цепи и спецификация в виде pLTL-формулы может быть представлена для протоколов взвешенного консенсуса [12]. В работе [12] при этом не исследован вопрос модификации модели консенсуса, если вероятность принятия транзакции недостаточна для пользователя. Более того, результат верификации может быть применен для автоматического режима планирования и отправки сообщений для достижения консенсуса. Интересен вопрос интеграции инструментальных средств верификации консенсуса с существующими платформами построения блокчейн-сетей. В настоящей статье рассмотрены перечисленные открытые вопросы.

Технология распределенного реестра и метод статистической проверки моделей

В настоящем разделе представлены протоколы консенсуса, которые впоследствии будут верифицированы. Также рассмотрены методы и инструментальные средства статистической проверки моделей, которые используются для верификации многосторонних соглашений.

Многосторонние соглашения в блокчейн-системах. Блокчейн является распределенным реестром, где каждый из участников хранит все записи, которые считаются валидными для сети [13]. Каждый участник сети может предложить транзакцию на запись. Если определенные участники, которые заданы в протоколе консенсуса, подтверждают транзакцию, то транзакция считается валидной и все участники сети ее принимают.

В блокчейн-сетях присутствуют разные протоколы консенсуса: доказательство работы; доказательство доли; доказательство веса и др. [13]. Условие консенсуса может задаваться в виде логической формулы. Значение каждой переменной формулы представляет ответ определенного участника консенсуса. В HLF условие соглашения по транзакции, которое называют правилом одобрения, можно задать логической формулой [14]. Каждого участника правила одобрения в HLF принято называть организацией. Далее по тексту будем придерживаться данной терминологии.

Из предшествующих по времени данных можно получить статистику ответов подтверждений и отказа транзакций для каждой организации. Из собранной статистики для каждого участника можно определить вероятность подтверждения следующей транзакции с помощью байесовского подхода [15, 16]. Таким образом, достижение консенсуса является вероятностным событием.

Статистическая проверка моделей. Проверка моделей является одним из формальных методов верификации. Проверка моделей состоит из перечисленных далее трех фаз [6].

- Фаза моделирования. Моделирование системы, формализация свойств для их проверки.
- Фаза запуска, а именно запуск инструментальных механизмов проверки моделей. Проверка того, что модель удовлетворяет спецификации.
- Фаза анализа, на которой можно проверять следующие далее свойства, если модель удовлетворяет спецификации. В противном случае нужно изменить модель или спецификацию и повторить процедуру проверки модели.

В классической проверке модели не учитываются вероятностные свойства модели. Метод статистической проверки моделей позволяет проверить на модели вероятностные свойства. Метод статистической проверки моделей позволяет ответить на следующие вопросы [17]:

- количественный — вероятность того, что система удовлетворит спецификации на уровне больше порогового значения;
- качественный, показывающий какова вероятность удовлетворить требованиям спецификации.

Для ответа на поставленные вопросы используется один из перечисленных далее методов.

- Метод проверки статистических гипотез. Допустим, $p = Pr(x)$ — вероятность случайного события x. Чтобы определить, что $p \geqslant Th$, где Th — некоторое пороговое значение, необходимо протестировать гипотезу H: $p \geqslant Th$ и гипотезу K: $p < Th$.
- Метод интервальной оценки статистических параметров. Используется для определения значений случайных событий с заданным распределением.

В настоящей работе представлен ответ на качественный вопрос. При этом используется метод проверки статистических гипотез, так как метод оценки статистических параметров больше подходит для эмпирических экспериментов [18]. В методе интервальной оценки статистических параметров необходимо строить оценку вероятностных величин на основе функции распределения вероятности. Из статистических данных подтверждения и отклонения транзакции не всегда можно определить функцию распределения вероятности, поэтому применяется метод проверки статистических гипотез. Предложенное решение использует инструментальное средство PRISM [19]. Оно позволяет построить модель консенсуса в виде модели марковской цепи с дискретным временем (МЦДВ, или

discrete-time Markov chain — DTMC), а также задать спецификацию в виде pLTL-формулы.

Построение и использование модели и спецификации для протоколов многостороннего соглашения

Построение модели протокола консенсуса. Пользователь может передать конфигурацию консенсуса в следующем виде:

- множество организаций;
- для каждой организации i вероятность подтверждения транзакции P_i;
- спецификация консенсуса в виде логической формулы, которая изначально может быть задана в любом виде, однако для построения модели необходимо будет задавать формулу в дизъюнктивной нормальной форме (ДНФ).

Алгоритм 1 (см. далее) описывает создание модели консенсуса. Алгоритму передается множество пар *orgs*. Каждая пара — это имя организации и вероятность подтверждения транзакции для данной организации. Модель представляет собой бинарное дерево, где каждому узлу соответствует организация. Каждый узел хранит информацию о родителе, об ответе, который дал родитель на подтверждение транзакции, а также P_i — вероятность перехода по ребру, которое соответствует подтверждению транзакции, i — порядковый номер организации. Сначала выбирается организация для корня дерева, от которого строятся два перехода. Один переход соответствует подтверждающему ответу организации, которой поставлен в соответствие корень, второй — отказу. Переход по первому ребру происходит с вероятностью P_i , по второму — с вероятностью $1 - P_i$. Далее корень удаляется из множества *orgs* и рекурсивно строится левое и правое поддерево. Порядок организаций в множестве *orgs* не важен, так как предполагается, что вероятности подтверждений разных организаций независимы.

Построение спецификации для протокола консенсуса. Из спецификации в виде формулы ДНФ можно получить множество узлов, которые соответствуют достижению консенсуса. Алгоритм 2 (см. далее) описывает процесс получение pLTL-формулы. Для каждого литерала из изначальной спецификации в виде ДНФ происходит обход дерева марковской цепи вглубь. Если организация, которая соответствует текущему узлу обхода, находится в литерале, то совершается переход по ребру, которое соответствует подтверждающему ответу. Если же организация не присутствует в литерале, происходит переход и в левое, и в правое поддерево, так как это соответствует как подтверждению транзакции, так и отклонению ее. Далее продолжается обход вглубь по двум поддеревьям. В итоге будет получено множество листьев, каждое из которых соответствует подтверждению транзакции.

Для подтверждения транзакции необходимо достичь хотя бы один из таких листьев, поэтому строится их дизъюнкция. Так как достаточно посетить лист один раз, перед дизъюнкцией ставится временной оператор "Eventually" *F,* а также вероятностный оператор *P*.

Число узлов в модели, построенной в соответствии с алгоритмом 1, растет экспоненциально относительно числа организаций. Таким образом, алгоритм 1 имеет экспоненциальную сложность. Алгоритм 2 также имеет экспоненциальную сложность, так как в худшем случае надо обойти все поддеревья, чтобы получить листья, которые соответствуют подтверждению.

Алгоритм 1: создание модели DTMC-creation

Входные данные: множество пар <организация, вероятность> *orgs,* корень *root,* множество узлов *nodes*
Результат: Множество с узлами модели *nodes,* хранящих информацию о модели
если лист *orgs* не пустой, то:
извлечь из листа организацию *nextOrg* с вероятностью $P_{nextOrg}$
// построить поддерево с подтверждающим ответом:
создать узел $N_{nextOrg}$, родителем которого является *root*, с параметром $P_{nextOrg}$ и подтверждающим ответом;
добавить узел $N_{nextOrg}$ с подтверждающим ответом родителя в множество *nodes*;
//запустить алгоритм рекурсивно для построения поддерева:
DTMC-creation(copy(orgs), $N_{nextOrg}$*, nodes)*;
//построить поддерево с ответом отказа:
создать узел $N_{nextOrg}$, родителем которого является *root*, с параметром $N_{nextOrg}$ и с ответом отказа;
добавить узел $N_{nextOrg}$ с ответом отказа родителя в множество *nodes*;
// запустить алгоритм рекурсивно для построения поддерева:
DTMC-creation(orgs, $N_{nextOrg}$*, nodes)*;
если лист *orgs* **пустой, то:**
создать два листовых узла из корня *root* с двумя переходами, соответствующими ответам подтверждения и отказа;
добавить листья в множество *nodes*;
завершение алгоритма.

Алгоритм 2: создание спецификации для модели DTMC

Входные данные: Множество узлов модели *nodes*, спецификация в виде ДНФ *s*
Результат: спецификация в виде pLTL *s_new*
инициализировать множество листовых узлов *leafs*;
Для каждого литерала *lit* из спецификации *s* обойти дерево вглубь:
если организация, соответствующая узлу, встречается в *lit*, то совершить переход, соответствующий подтверждению транзакции;
если не встречается, то продолжить переход по двум поддеревьям: одно соответствует подтверждению, второе — отклонению транзакции;
каждый лист, который был достигнут при обходе вглубь, добавить в множество *leafs*;
s_new = дизъюнкция узлов из множества *leafs*;
добавить в начало *s_new* временной оператор *F*;
добавить в начало *s_new* вероятностный оператор *P*;
возвратить формулу *s_new*.

Использование модели и спецификации для отправки транзакции на подтверждение. После того как пользователь построил модель, можно отправлять сообщения в соответствии с этой моделью. Отправка сообщений начинается с корня дерева. Если получен ответ подтверждения, то совершается переход, соответствующий утвердительному ответу. Переход по дереву заканчивается, как только достигнут один из листьев. Если листовой узел принадлежит множеству *s_new*, которое было получено из алгоритма 2, то консенсус заканчивается принятием транзакции. При каждом переходе отправляется сообщение на подтверждение.

В некоторых моделях в случае получения ответа об отклонении транзакции может быть совершен обратный переход. Обратный переход ведет к корню дерева и инициирует процесс обхода модели заново от корня. Обратные переходы увеличивают вероятность подтверждения транзакции, но также увеличивают и число сообщений, прошедших в сети. Может быть наложено ограничение на максимальное число сообщений в модели. Более детально рассмотрим применение обратных переходов в разделе описания инструментального средства.

Инструментальные средства автоматической верификации и планирования отправки сообщений в протоколах многостороннего соглашения

В предыдущем разделе описаны алгоритмы получения модели в виде МЦДВ, а спецификации — в виде pLTL-формулы. Описан процесс отправки транзакции на подтверждение в соответствии с выбранной моделью. Таким образом, реализованы алгоритмы в инструментальные средства, которые состоят из двух описанных далее модулей.

- Consensus analyzer. Данный модуль принимает на вход конфигурацию консенсуса. В конфигурации содержится множество организаций, вероятность подтверждения транзакции для каждой организации, а также логическая формула. После этого строится модель марковской цепи и вычисляется вероятность достижения консенсуса и математическое ожидание числа сообщений. Опционально могут быть построены модели для всех возможных комбинаций обратных переходов.
- Consensus scheduler. Модуль принимает на вход модель, которая была построена модулем анализатора, а также реализацию логики отправки транзакции на подтверждение. В соответствии с моделью модуль осуществляет отправку сообщений организациям для подтверждения транзакции.

Разработанный авторами программный механизм реализован на языке Java 11. Программный код модулей можно найти в git-репозитории [20, 21]. В модуле анализатора присутствует зависимость с инструментальным средством статистической проверки моделей PRISM [19]. Модуль планировщика не имеет внешних зависимостей. На момент написания статьи модуль отправки сообщений реализован на Java, но данный модуль может быть реализован на любом языке программирования, так как он не требует внешних зависимостей. Это обеспечивает кроссплатформенность работы инструментального средства. Модель может быть построена на одной машине, где требуется Java и зависимость PRISM. Далее модель в виде текстового файла в формате .json передается на другую машину, которая подключена к блокчейн-сети. На этой машине установлен модуль отправки сообщений. В текущем разделе детально рассмотрим работу каждого из представленных выше модулей.

Consensus analyzer. Этот модуль принимает на вход конфигурацию консенсуса в формате .yaml. Конфигурация содержит имена организаций, а также вероятность принятия транзакции для каждой организации. Конфигурация содержит условие консенсуса: логическую формулу, где каждая переменная — это имя организации, участвующей в консенсусе. Для запуска модуля необходимо сохранить его исходный код и PRISM в одну и ту же директорию. Далее нужно запустить make-файл для модуля и для PRISM. После этого исходный код будет скомпилирован и можно запускать анализатор. Запускать его можно в двух режимах: первый строит одну модель без обратных переходов; второй — множество моделей со всеми уникальными комбинациями обратных переходов от узла к корню в случае, если был дан ответ

отклонения транзакции. Для каждой модели вычисляются вероятность принятия транзакции, а также математическое ожидание числа сообщений, прошедших в сети до принятия транзакции.

Модуль работает следующим образом. Сначала считывается конфигурация, а логическая формула приводится в вид ДНФ. На основе алгоритма 1 строится модель марковской цепи, а на основе алгоритма 2 — pLTL-формула. Если модуль запущен в режиме обратных переходов, то также вычисляются все возможные комбинации обратных переходов и для каждой строится модель марковской цепи и спецификация. Для каждой модели с помощью инструментальных средств PRISM вычисляется вероятность достижения консенсуса, а также математическое ожидание числа переданных сообщений до того как транзакция принята необходимым числом участников консенсуса. Результатом работы модуля является файл в формате .json, где каждой модели ставятся в соответствие вероятность и число сообщений. Модуль строит текстовое представление каждой модели, а также график зависимости вероятности подтверждения от математического ожидания числа сообщений. По графику пользователь может определить модель, которая ему наиболее подходит. Выбранную модель использует второй модуль, предназначенный для отправки сообщений.

Consensus scheduler. Данный модуль принимает на вход json-файл, который получается на выходе модуля consensus analyzer. После этого пользователю предлагается выбрать параметры алгоритма отправки сообщений, а также модель для отправки. Параметрами алгоритма отправки являются максимальное число сообщений в сети, максимальное число сообщений, которое может быть отправлено одной организации, а также время ожидания между отправками сообщений одному и тому же узлу. Модуль работает исходя из предположения, что время между отправками сообщений для одной и той же организации достаточно для того, чтобы организация изменила ответ на принятие транзакции. Порог для максимального числа сообщений может помочь в случае, если одна или несколько организаций по техническим причинам перестали отвечать. Предполагается, что достижение таймаута ответа для организации равносильно отклонению транзакции.

Модуль можно подключить как стороннюю Java-зависимость. Возможна интеграция модуля с фреймворком HLF, который имеет программный интерфейс Java, позволяющий задавать логику отправки транзакции на подтверждение. В программный код, где используется фреймворк HLF, можно подключить зависимость модуля планировщика, который примет на вход файл анализатора. Далее, используя программный интерфейс планировщика, пользователь может инициировать автоматическую отправку транзакции на подтверждение в соответствии с выбранной моделью и параметрами отправки. В следующем разделе рассмотрим ряд экспериментов с участием разработанного инструментального средства.

Экспериментальная оценка работы инструментального средства

Проиллюстрируем работу алгоритмов и программного механизма на примере построения модели консенсуса. Модель похожа на ту, которая использовалась в предыдущей работе авторов [12] по этой тематике. Однако в данном эксперименте модель строится не для взвешенного консенсуса, а для консенсуса, условие которого представлено в виде логической формулы. Эксперимент преследует следующие цели. Во-первых, следует проиллюстрировать работу предложенного программного комплекса и инструментальных механизмов статистической проверки моделей PRISM для анализа протоколов консенсуса. Во-вторых, необходимо проверить корректность работы инструментального средства и алгоритмов. В этих целях изменим параметры консенсуса и проверим, как меняется результат, который показывает программный механизм. Рассмотрим консенсус, в котором участвуют три организации с соответствующими вероятностями подтверждения транзакции, которые применялись в работе [12]:

- организация 1, имя *Org1*, вероятность подтверждения транзакции 0,93;
- организация 2, имя *Org2*, вероятность подтверждения транзакции 0,99;
- организация 3, имя *Org,* вероятность подтверждения транзакции 0,98.

Условие консенсуса выглядит следующим образом: транзакция принимается сетью, если большинство организаций, участвующих в правиле одобрения, подтвердили транзакцию. В виде логической формулы это условие можно записать следующим образом:

(*Org1* AND *Org2*) OR (*Org2* AND *Org3*)
OR (*Org1* AND *Org3*).

Модель, построенная для трех организаций по алгоритму 1, представлена на рис. 1. Разберем подробно процесс построения. Сначала из множества организаций выбираем первую организацию, которую поставим в корень, это *Org1.* Далее, строим от нее два ребра: одно соответствует подтверждению транзакции; второе — отклонению. Соответственно, переход по ребру, которое соответствует подтверждению, случается с вероятностью 0,93, а по второму ребру, которое соответствует отклонению, с вероятностью отклонению 0,07. Далее выбираем вторую организацию из множества, это *Org2*, и рекурсивно применяем алгоритм 1 к левому и правому поддеревьям с новым корнем *Org2*. Потом аналогично строим поддеревья для организации *Org3,* и от каждого узла, который соответствует *Org3,* строим два листовых узла, так как это последняя организация из множества. Модель марковской цепи для алгоритма консенсуса построена, далее рассмотрим построение спецификации для модели.

В соответствии с алгоритмом 2 для построения спецификации проходим дерево марковской цепи

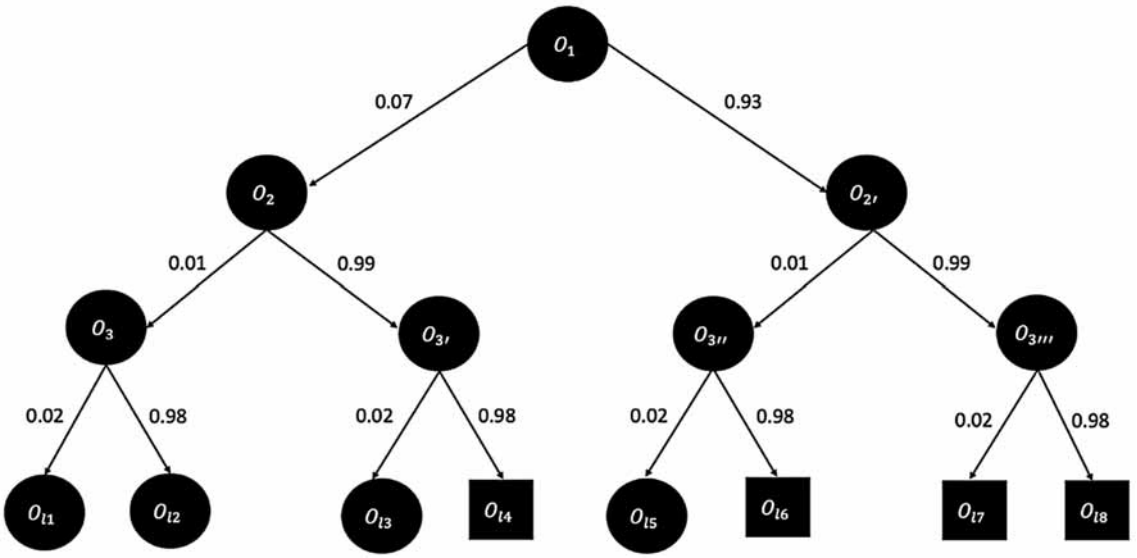

Рис. 1. Модель Марковской цепи для протокола консенсуса

вглубь для каждого литерала. Рассмотрим на примере литерала (*Org1* AND *Org2*). От узла O_1, который соответствует организации *Org1*, переходим по правому ребру, так как *Org1* присутствует в литерале, а правое ребро соответствует принятию транзакции. Аналогично для узла O_2, также переходим по правому ребру. Организация *Org3* не присутствует в литерале, поэтому совершаем переход как по правому, так и по левому ребрам. В итоге получаются узлы O_{l7}, O_{l8}, которые соответствуют принятию транзакции.

Были рассмотрены узлы только для первого литерала. Аналогично можно получить узлы, которые соответствуют принятию транзакции для второго и третьего литералов. На рис. 1 эти узлы обозначены квадратами. Для достижения соглашения по принятию транзакции необходимо посетить хотя бы один из узлов, поэтому строим дизъюнкцию узлов, которые соответствуют принятию транзакции. Так как необходимо посетить узел один раз, то ставим перед дизъюнкцией временной оператор "Eventually". В соответствии с алгоритмом 2 спецификация для марковской цепи выглядит так:

$$P = ?[F(O_{l4} \text{ OR } O_{l6} \text{ OR } O_{l7} \text{ OR } O_{l8})].$$

Таким образом, детально разобрано построение модели и спецификации, которые автоматически строятся с помощью программного модуля consensus analyzer. Далее запустим инструментальные средства и получим вероятность принятия транзакции. Спецификацию протокола консенсуса можно найти в git-репозитории [20].

Сначала запускаем анализатор без обратных переходов. Вероятность подтверждения транзакции равна 0,997728, среднее число сообщений до достижения консенсуса — 3. Этот результат соответствует ожиданиям. Так как вероятность подтверждения каждой организации по отдельности высока, то и итоговая вероятность принятия транзакции будет высокая. Принимаем во внимание всего один раунд подтверждения и три организации — ожидаемое число сообщений также равно трем.

Далее запускаем алгоритм анализа с обратными переходами. Одна из моделей с обратными переходами проиллюстрирована на рис. 2.

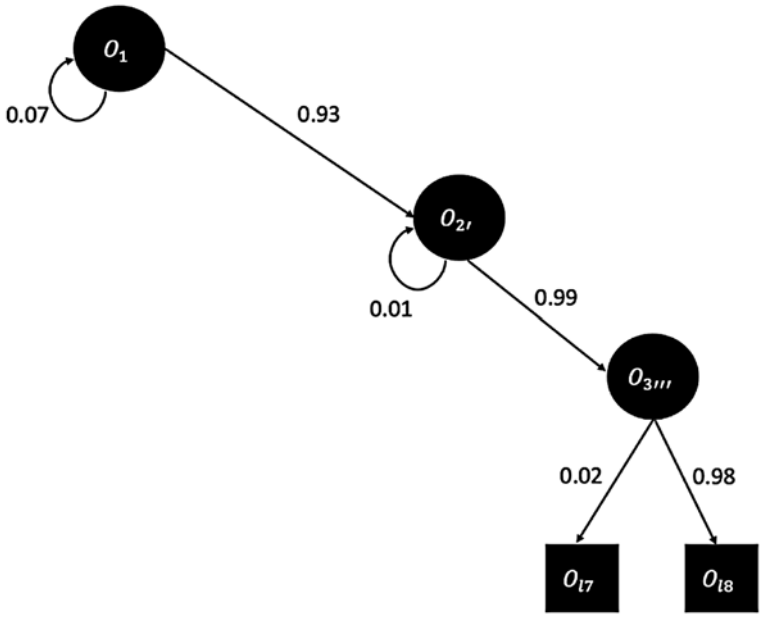

Рис. 2. Модель Марковской цепи с обратными переходами

Выбрана модель с обратными переходами, которая соответствует наибольшей вероятности достижения консенсуса при наименьшем числе сообщений. Вероятность принятия транзакции равна 1, математическое ожидание числа сообщений 3,2212. Как видно на рис. 2, все переходы ведут к листьям, которые соответствуют принятию транзакции O_{l7}, O_{l8}, что дает стопроцентную вероятность подтверждения транзакции.

Теперь рассмотрим, как меняется результат анализа при изменении параметров консенсуса. Поднимем вероятность подтверждения транзакции для первой и второй организаций до 1. Программный механизм показывает, что вероятность принятия транзакции в этом случае также равна 1. Это ожидаемо, так как в спецификации вероятность, что первый литерал будет верен, равна 1.

Проведем противоположный эксперимент: снизим вероятность принятия транзакции для организации 3 до нуля. В этом случае анализатор показывает, что вероятность принятия транзакции равна 0,9207. Это также соответствует ожиданиям, так как это число равно произведению вероятностей первой и второй организаций: $0{,}93 \times 0{,}99 = 0{,}9207$.

Итак, показано, что результаты экспериментов, полученные в ходе работы инструментального средства, совпадают с теоретическими расчетами. Далее одну из моделей можно передать на вход модулю consensus scheduler. Модуль consensus scheduler позволяет выбрать модель с наименьшим математическим ожиданием числа сообщений в сети, или модель с наивысшей вероятностью подтверждения, либо любую другую модель, построенную модулем consensus analyzer. Выбор модели остается на усмотрение пользователя.

Заключение

В настоящей работе представлены разработанные авторами алгоритмы построения модели и спецификации для протоколов консенсуса, задаваемой в виде логических формул. Продемонстрировано, как на основе выбранной модели пользователь может автоматически отправлять транзакции на подтверждение. Разработанный программный механизм реализует предложенные алгоритмы. Он состоит из двух программных модулей. Модуль для анализа консенсуса consensus analyzer определяет вероятность принятия транзакции, а также математическое ожидание числа сообщений, прошедших в сети до подтверждения транзакции. После этого результат анализатора может быть передан модулю consensus scheduler, который инициирует отправку сообщений в соответствии с выбранной моделью. Разработанный программный механизм может быть интегрирован с фреймворком Hyperledger Fabric. Перспективным направлением является его использование в промышленных системах, а также проведение испытаний с помощью метода ab-test [22] с целью выявить изменение числа подтверждения транзакций с работой инструментальных средств и без них.

Список литературы

1. **Jaoude J. A., Saade R. G.** Blockchain Applications — Usage in Different Domains // IEEE Access. 2019. Vol. 7. P. 45360—45381.

2. **Bach L., Branko M., Zagar M.** Comparative analysis of blockchain consensus algorithms // MIPRO. 2018. P. 1545—1550.

3. **Androulaki E., Barger A., Bortnikov V., Cachin C., Christidis K., De Caro A., Enyeart D., Ferris C., Laventman G., Manevich Y.** Hyperledger fabric: a distributed operating system for permissioned blockchains // Proceedings of the Thirteenth EuroSys Conference. 2018. P. 1—15.

4. **Saad M., Spaulding J., Njilla L., Kamhoua C., Shetty S., Nyang D., Mohaisen A.** Exploring the attack surface of blockchain: A systematic overview // IEEE Communications Surveys & Tutorials. 2020. Vol. 22, No. 3. P. 1977—2008.

5. **Dabholkar A., Saraswat V.** Ripping the fabric: Attacks and mitigations on hyperledger fabric // Applications and Techniques in Information Security. ATIS 2019. Communications in Computer and Information Science. 2019. Vol. 1116. P. 300—311.

6. **Baier C., Katoen J.** Principles of model checking. MIT press, 2008. 984 p.

7. **Legay A., Delahaye B., Bensalem S.** Statistical model checking: An overview // Runtime Verification. RV 2010. Lecture Notes in Computer Science. 2010. vol. 6418. P. 122—135.

8. **Федотов И. А., Хританков А. С.** Систематический обзор исследований в области автоматической верификации кода смарт-контрактов // Программная инженерия. 2020. Том 11, № 1. С. 3—13.

9. **Agha G., Palmskog K.** A survey of statistical model checking // *ACM Transactions on Modeling and Computer Simulation (TOMACS)*. 2018. Vol. 28, No. 1. P. 1—39.

10. **Fedotov I., Khritankov A.** Statistical Model Checking of Common Attack Scenarios on Blockchain // Proceedings of the 9th International Symposium on Symbolic Computation in Software Science. 2021. P. 65—77.

11. **Imeri A., Agoulmine N., Khadraoui D.** Smart Contract modeling and verification techniques: A survey // 8th International Workshop on ADVANCEs in ICT Infrastructures and Services. 2020. P. 1—8.

12. **Fedotov I., Khritankov A., Barger A.** Towards automated verification of multi-party consensus protocols. arXiv preprint. 2021.

13. **Zheng Z., Xie S., Dai H., Chen X., Wang H.** An overview of blockchain technology: Architecture, consensus, and future trends // 2017 IEEE International Congress on Big Data. 2017. P. 557—564.

14. **Cachin C.** Architecture of the hyperledger blockchain fabric // Workshop on distributed cryptocurrencies and consensus ledgers. 2016. Vol. 310, No. 4. P. 1—4.

15. **Duda J.** Exploiting statistical dependencies of time series with hierarchical correlation reconstruction// CoRR. 2018. P. 11—24.

16. **Corani G., Benavoli A.** A bayesian approach for comparing cross-validated algorithms on multiple data sets // Machine Learning. 2015. Vol. 100, No. 2-3. P. 285—304.

17. **Legay A.** Statistical model checking // Computing and Software Science. Lecture Notes in Computer Science. 2016. Vol. 10000. P. 478—504.

18. **Younes H., Simmons R.** Probabilistic verification of discrete event systems using acceptance sampling // Computer Aided Verification. CAV 2002. Lecture Notes in Computer Science. 2002. Vol. 2404. P. 223—235.

19. **Kwiatkowska M., Norman G., Parker D.** PRISM 4.0: Verification of probabilistic real-time systems // Computer Aided Verification. CAV 2011. Lecture Notes in Computer Science, 2011. Vol. 6806. P. 585—591.

20. **Fedotov I., Morounfoluwa D. O.** Consensus Analyzer. URL: https://github.com/1vanan/consensus-analyzer

21. **Fedotov I., Morounfoluwa D. O.** Consensus Scheduler. URL: https://github.com/1vanan/consensus-scheduler

22. **Gronau Q., Akash Raj N., Wagenmakers E.** Informed Bayesian Inference for the A/B Test. arXiv preprint. 2019.

Automated Verification of Multi-Party Agreements and Scheduling of Sending Messages in Distributed Ledger Systems

I. A. Fedotov[1], ivan.fedotov@phystech.edu, **A. S. Khritankov**[2], anton.khritankov@acm.org,
M. D. Obidare[1], obidarefolu1@gmail.com,
[1] MIPT, Dolgoprudny, Moscow Region, 141701, Russian Federation,
[2]HSE, Moscow, 115432, Russian Federation

Corresponding author:

Fedotov Ivan A., Postgraduate Student, MIPT, Dolgoprudny, Moscow Region, 141701, Russian Federation
E-mail: ivan.fedotov@phystech.edu

Received on February 28, 2022
Accepted on March 06, 2022

One can use a multi-party agreement in distributed ledger systems and blockchain networks to reach an agreement on changes of the state of the system. If one of the network members proposes a transaction, then certain network participants shall confirm it. After that the whole network can consider transaction as a valid one. A multi-party agreement or consensus determines the composition of these participants. Based on the historical data set, one can calculate the probability of confirming a transaction for each of the participants. In this paper, we use a statistical model checking approach to determine the likelihood that the network accepts a transaction. Sending confirmation requests may require an additional fee. We calculate the probability, and the mathematical expectation of the number of messages before reaching a consensus. Further, consensus models are built in the form of a Markov chain with various strategies for sending messages. Based on the proposed methods, we design a tool that automatically builds models for various strategies of sending messages and verifies the model using a statistical model verification approach. After choosing the optimal model, one can send confirmation messages using the scheduler module of developed tool.

Keywords: *consensus, blockchain, verification, statistical model checking*

For citation:

Fedotov I. A., **Khritankov A. S.**, **Obidare M. D**. Automated Verification of Multi-Party Agreements and Scheduling of Sending Messages in Distributed Ledger Systems, *Programmnaya Ingeneria,* 2022, vol. 13, no. 4, pp. 200—208.

DOI: 10.17587/prin.13.200-208

References

1. **Jaoude J. A., Saade R. G.** Blockchain Applications — Usage in Different Domains, *IEEE Access,* 2019, vol. 7, pp. 45360—45381.

2. **Bach L., Branko M., Zagar M.** Comparative analysis of blockchain consensus algorithms, *MIPRO*, 2018, pp. 1545—1550.

3. **Androulaki E., Barger A., Bortnikov V., Cachin C., Christidis K., De Caro A., Enyeart D., Ferris C., Laventman G., Manevich Y.** Hyperledger fabric: a distributed operating system for permissioned blockchains, *Proceedings of the Thirteenth EuroSys Conference,* 2018, pp. 1—15.

4. **Saad M., Spaulding J., Njilla L., Kamhoua C., Shetty S., Nyang D., Mohaisen A.** Exploring the attack surface of blockchain: A systematic overview, *IEEE Communications Surveys & Tutorials*, 2020, vol. 22, no. 3, pp. 1977—2008.

5. **Dabholkar A., Saraswat V.** Ripping the fabric: Attacks and mitigations on hyperledger fabric, *Applications and Techniques in*

Information Security. ATIS 2019. Communications in Computer and Information Science, 2019, vol. 1116, pp. 300—311.

6. **Baier C., Katoen J.** *Principles of model checking,* MIT press, 2008. 984 p.

7. **Legay A., Delahaye B., Bensalem S.** Statistical model checking: An overview, *Runtime Verification. RV 2010. Lecture Notes in Computer Science*, 2010, vol. 6418. pp. 122—135.

8. **Fedotov I. A., Khritankov A. S.** Systematic Review of Automatic Verification of Smart-Contracts, *Programmnaya Ingeneria,* 2020, vol. 11, no. 1, pp. 3—13 (in Russian).

9. **Agha G., Palmskog K.** A survey of statistical model checking, *ACM Transactions on Modeling and Computer Simulation (TOMACS),* 2018, vol. 28, no. 1, pp. 1—39.

10. **Fedotov I., Khritankov A.** Statistical Model Checking of Common Attack Scenarios on Blockchain, *Proceedings of the 9th International Symposium on Symbolic Computation in Software Science*, 2021, pp. 65—77.

11. **Imeri A., Agoulmine N., Khadraoui D.** Smart Contract modeling and verification techniques: A survey, *8th International Workshop on ADVANCEs in ICT Infrastructures and Services,* 2020, pp. 1—8.

12. **Fedotov I., Khritankov A., Barger A.** Towards automated verification of multi-party consensus protocols, arXiv preprint. 2021.

13. **Zheng Z., Xie S., Dai H., Chen X., Wang H.** An overview of blockchain technology: Architecture, consensus, and future trends, *2017 IEEE International Congress on Big Data,* 2017, pp. 557—564.

14. **Cachin C.** Architecture of the hyperledger blockchain fabric, *Workshop on distributed cryptocurrencies and consensus ledgers,* 2016, vol. 310, no. 4, pp. 1—4.

15. **Duda J.** Exploiting statistical dependencies of time series with hierarchical correlation reconstruction, *CoRR,* 2018, pp. 11—24.

16. **Corani G., Benavoli A.** A bayesian approach for comparing cross-validated algorithms on multiple data sets, *Machine Learning,* 2015, vol. 100, no. 2-3. pp. 285—304.

17. **Legay A.** Statistical model checking, *Computing and Software Science. Lecture Notes in Computer Scienc*e, 2016, vol. 10000, pp. 478—504.

18. **Younes H., Simmons R.** Probabilistic verification of discrete event systems using acceptance sampling, *Computer Aided Verification. CAV 2002. Lecture Notes in Computer Science,* 2002, vol. 2404, pp. 223—235.

19. **Kwiatkowska M., Norman G., Parker D.** PRISM 4.0: Verification of probabilistic real-time systems, *Computer Aided Verification. CAV 2011. Lecture Notes in Computer Science,* 2011, vol. 6806, pp. 585—591.

20. **Fedotov I., Morounfoluwa D. O.** Consensus Analyzer, available at: https://github.com/1vanan/consensus-analyzer

21. **Fedotov I., Morounfoluwa D. O.** Consensus Schefuler, available at: https://github.com/1vanan/consensus-scheduler

22. **Gronau Q., Akash Raj N., Wagenmakers E.** Informed Bayesian Inference for the A/B Test. arXiv preprint. 2019.

ООО "Издательство "Новые технологии". 107076, Москва, ул. Матросская Тишина, д. 23, стр. 2
Технический редактор *Е. М. Патрушева*. Корректор *А. В. Чугунова*.

Сдано в набор 09.03.2022 г. Подписано в печать 05.04.2022 г. Формат 60×88 1/8. Заказ PI421
Цена свободная.

Оригинал-макет ООО "Адвансед солюшнз". Отпечатано в ООО "Адвансед солюшнз".
119071, г. Москва, Ленинский пр-т, д. 19, стр. 1. Сайт: **www.aov.ru**