

Генерация и исследование коммуникационных протоколов с помощью НК-автоматов и их модификаций

Е.А. Кольчугина ^{1✉}, В.А. Стежка ²

¹ Пензенский государственный университет, г. Пенза, 440026, Россия

² Научно-технический центр АО «Радиозавод», г. Пенза, 440039, Россия

Ссылка для цитирования

Кольчугина Е.А., Стежка В.А. Генерация и исследование коммуникационных протоколов с помощью НК-автоматов и их модификаций // Программные продукты и системы. 2025. Т. 38. № 4. С. 598–607. doi: 10.15827/0236-235X.152.598-607

Информация о статье

Группа специальностей ВАК: 2.3.8

Поступила в редакцию: 26.05.2025

После доработки: 09.06.2025

Принята к публикации: 20.06.2025

Аннотация. Предметом анализа являются методы и алгоритмы, которые позволяют интеллектуальным устройствам исследовать и осваивать коммуникационные протоколы распределенных инфокоммуникационных систем и сред. В качестве научного подхода был выбран эволюционный метод построения НК-автоматов и их модификаций. Цель состояла в том, чтобы получить НК-автомат, способный воспроизводить последовательность пакетов протокола TCP через последовательность своих состояний. Эта последовательность должна быть корректной с точки зрения готового общедоступного эхо-сервера TCP и подходящей для передачи данных в ходе TCP-сессии. Создание такого автомата означает, что интеллектуальное устройство обучилось протоколу TCP. Для достижения этой цели были разработаны модификация классического НК-автомата, TCP-НК-автомат, а также ПО для имитации его работы. В отличие от других модификаций предложенная версия моделирует не активную программную структуру, а пассивную структуру данных – TCP-пакет. В ходе экспериментов с созданным ПО удалось получить экземпляры автоматов, которые генерировали правильные последовательности пакетов протокола TCP, имитируя TCP-сессию, что свидетельствует об успешном обучении. Полученные результаты способны обеспечить совместимость различных устройств через их обучение новым протоколам без использования нейросетей, позволяют обойтись без использования брокеров, представляют возможность исследования и обратного инжиниринга ранее неизвестных протоколов.

Ключевые слова: исследование коммуникационных протоколов, НК-автоматы, вычислительные среды, генерация, эволюционное обучение устройств

Введение. Повсеместное внедрение интеллектуальных вычислительных устройств, развитие сетевых технологий и, в частности, концепции Интернета вещей актуализирует проблемы и задачи, связанные с обеспечением слаженного взаимодействия или, в минимальном варианте, бесконфликтного сосуществования множества устройств различной сложности и степени интеллектуальности, созданных разными производителями в разные годы и в соответствии с отличными стандартами.

Подразумевается, что современные интеллектуальные устройства должны оснащаться сенсорами и средствами коммуникации, что позволяет им функционировать не изолированно, а в составе некой экосистемы, включающей в себя и другие подобные им устройства, с которыми необходимо взаимодействовать. То есть упомянутые устройства изначально ориентированы на общение как с реальным внешним миром, так и с виртуальным, при этом их языки общения различаются. Для обеспечения слаженной безопасной работы интеллектуальных устройств необходимо наделить их спо-

собностями к взаимодействию и согласованию своих действий, что подразумевает способность договариваться. Данное свойство важно для обеспечения обратной совместимости устройств и продления срока их службы: развитие языков и протоколов общения не должно исключать из работы уже функционирующие устройства, использующие более старые языки и протоколы.

В современной науке эта проблематика соответствует направлению M2M (*Machine-to-Machine Communications*) [1, 2] как ответвлению ИИ. Одним из наиболее известных решений в этой области является использование протокола MQTT [3, 4], предполагающего организацию взаимодействия устройств с помощью брокеров. Такая концепция имеет следующие недостатки:

– возрастание сложности алгоритмов брокера в связи с увеличением количества и разнообразия устройств, при этом придется учитывать все большее количество возможных ситуаций и частных случаев, поэтому для реализации брокеров потребуются высокопроизводитель-

ные вычислительные системы, хранилища данных существенного объема и средства связи с высокой пропускной способностью;

– уязвимость к отказам – недостатком любых систем с централизованным управлением и логикой является наличие уязвимой точки в виде центра управления.

Другим, диаметрально противоположным подходом к решению указанной проблемы обеспечения взаимодействия множества коммуницирующих интеллектуальных устройств может стать отказ от централизации и наделение устройств способностью обучаться путем освоения языков и протоколов других устройств. Способность к обучению как к генерации новых логических структур прямо связана с самоорганизацией. Наиболее популярными примерами самоорганизующихся систем в информатике служат нейронные сети и мультиагентные системы. В случае последних самоорганизация осуществляется через постепенную подстройку параметров, регулирующих поведение роботов-агентов, как правило, реализованных программно. Вследствие чего формируется слаженное коллективное (социальное) поведение агентов вплоть до формирования и использования агентами эмпатии [5]. Подробный обзор и анализ имеющихся платформ для мультиагентных систем приведен в [6]. Нейронные сети также демонстрируют формирование слаженных вычислительных структур, которые все шире используются для решения практически важных задач благодаря сонстройке весовых коэффициентов [7–9]. Однако перечень моделей самоорганизующихся систем не ограничивается только моделями нейросетей и коллективного поведения агентов. Существуют и другие, которые демонстрируют способность к самоорганизации и, в частности, к образованию вычислительных структур с упорядоченным поведением. Эти модели менее сложные, а потому более подходящие для реализации в составе ПО небольших вычислительных устройств, поскольку не перегружают их функционально. К таким моделям можно отнести модель НК-автоматов Кауффмана [10] и ее модификации.

В данной работе ставится цель разработки модификации НК-автомата, которая позволяет одной из сторон клиент-серверного взаимодействия обучаться, анализируя сообщения, генерируемые противоположной стороной. Таким образом, фактически происходит обучение новому протоколу. В качестве примера для демонстрации возможностей НК-автоматной модели был взят широко известный протокол TCP с не-

сложной логической и процедурной характеристиками. Возможности, которые предоставляет предлагаемый в настоящей работе метод обучения на базе модификации НК-автоматной модели, можно охарактеризовать как способности к генерации и выполнению обратного инжиниринга протоколов.

Предыдущие исследования в области НК-автоматов и их модификаций

Модель НК-автомата была предложена С.А. Кауффманом [10] с целью исследования генной экспрессии и взаимовлияния генов друг на друга. Она включает в себя N элементов, каждый из которых соединен в среднем с другими K элементами. Возможное среднее число связей определяется как $1 \leq K \leq N$. Каждый из элементов реализует какую-либо несложную булеву функцию, таким образом, значений у каждой функции может быть только два: 0 или 1. НК-автомат рассматривается как динамическая система (или модель динамической системы), чьи состояния изменяются во времени. Его состояние описывается перечислением значений его выходов; время рассматривается как дискретное; N и K являются двумя параметрами, которые полностью определяют характер поведения автомата в пространстве состояний, вне зависимости от того, какие именно функции реализуются элементами.

Поскольку входы и выходы НК-автомата замкнуты друг на друга, его состояния со временем начинают повторяться. Последовательности состояний, начинающиеся и завершающиеся повторяющимися состояниями, представляют собой циклы. Если циклы повторения состояний устойчивы, это можно рассматривать как выход на аттрактор.

Особыми, критическими значениями с точки зрения смены характера поведения НК-автомата являются значения $K = 1, 2, 5, N$. Для случая $K = 1$ характерно большое количество циклов малой длины, поведение автомата характеризуется как упорядоченное. При $K = N$ длина циклов будет настолько большой, что поведение автомата можно рассматривать как максимально неупорядоченное, хаотическое, при этом возможное количество циклов будет небольшим. В случае $K = 5$ ($N \leq K \leq 5$) поведение автомата также близко к стохастическому, и чем большим становится значение K , тем больше длины циклов и меньше их самих.

Наиболее интересным является поведение при $K = 2$, которое можно определить как пове-

дение на границе порядка и хаоса, сопровождающееся выходом на аттрактор. При $K = 2$ медиана длины цикла и количество циклов определяются как \sqrt{N} . Это поведение рассматривается как фазовый переход от хаоса к порядку или как самоорганизация поведения автомата.

Известно, что НК-автоматы представляют собой несложные по построению модели, демонстрирующие самоорганизацию. Их исследования нашли свое продолжение и в виде многочисленных модификаций исходного НК-автомата, например, [11, 12]. С обзором таких модификаций можно ознакомиться в [13]. Остановимся на двух модификациях, введенных в [14–16], так как они предназначены для использования именно в компьютерных науках.

Изначально НК-автоматы были предложены как инструмент для исследований в области генетики с целью моделирования эпистаза и регуляции экспрессии генов. Однако сейчас эти модели применяются и в компьютерных науках, например, в области искусственной жизни. Работа [14] также относится к этой сфере, но по существу затрагивает исследование, доказательство и проектирование свойств ПО. В работе [14] была предложена модификация НК-автомата, получившая название НК-подобного. Этот автомат моделирует структуру самоорганизующегося программного агента – цифрового организма – и отличается от классического следующими свойствами:

- основное множество элементов автомата разбивается на два непересекающихся подмножества, одно из которых моделирует множество семантически целостных фрагментов программного кода, а другое – глобальные переменные, входящие в состав программы, модифицируемые в ходе выполнения фрагментов кода;

- в структуре автомата выделяется также управляющий элемент, задающий программу работы автомата (порядок активации элементов).

Было показано, что такой автомат соответствует классическому НК-автомату с $K = 2,33(3)$ [14], то есть он функционирует на границе порядка и хаоса. Это моделирует самоорганизацию поведения, то есть формирование алгоритма поведения программного агента (цифрового организма).

В работах [15, 16] предложена и исследована модификация НК-автомата Кауффмана, предназначенная для исследования поведения мультиагентной системы (в свою очередь имитирующей систему управления) в аспектах перехода к хаосу и функционирования в его усло-

виях. Отличительным свойством такой модификации является наличие параметра U , который определяет процент сходства состояний, позволяющий рассматривать их как неразличимые (идентичные).

Обе модификации НК-автомата из [14–16] ориентированы на моделирование активно действующих сущностей: отдельных программных агентов или мультиагентных сообществ. В отличие от этих модификаций в предлагаемом исследовании с помощью НК-автоматов моделируются пассивные сущности: структуры данных, представляющие пакеты ТСП. В настоящей работе продолжено развитие идей, изложенных в [17].

Метод исследования

Будем полагать, что имеется два абстрактных независимых устройства А и В, взаимодействующих посредством Интернета. Устройство А может взаимодействовать по протоколу ТСП, но не способно к обучению и изменению алгоритма взаимодействия. Устройство В, напротив, изначально не может взаимодействовать по протоколу ТСП, но способно к обучению как освоению нового протокола. Цель исследования состоит в том, чтобы предложить механизм, позволяющий устройству В освоить протокол взаимодействия ТСП. При этом в качестве материала и критерия обучения выступают ТСП-пакеты, генерируемые устройством А в ответ на транзакции устройства В.

Протокол ТСП был выбран за свою простоту, элегантность и эффективность используемых им решений. Аналогичные рассуждения могли быть выполнены и для других протоколов. Любой сетевой протокол включает в себя две составляющие: логическую и процедурную характеристики. Логическая характеристика представляет собой набор форматов структур данных (кадров, пакетов, сообщений, команд и т.п.), распознаваемых и генерируемых протоколом. Процедурная характеристика задает алгоритмическую, поведенческую составляющую как алгоритм реакции на входящие структуры данных и генерации собственных в ответ. Если проводить аналогии с естественными языками, то логическая характеристика определяет словарный состав языка и его синтаксис, а процедурная – речевой этикет и семантику.

Логическая и процедурная характеристики проявляются во взаимной связи: алгоритм, реализуемый процедурной характеристикой, определяет генерируемые исходящие структуры

данных, в то же время входящие структуры данных инициируют переход алгоритма на следующий шаг или этап.

Таким образом, воздействуя на формат и содержимое передаваемых пакетов на одной стороне TCP-соединения, можно управлять выполнением алгоритма TCP-сеанса на другой. Большую роль в управлении ходом TCP-сеанса играют управляющие биты, входящие в состав TCP-пакета. Их комбинации позволяют подтверждать получение данных, инициировать начало нового сеанса, анонсировать разъединение.

Краеугольный стандарт протокола TCP, а именно RFC-793 (<https://datatracker.ietf.org/doc/html/rfc793>), определяет в составе пакета TCP шесть управляющих битов (RST, FIN, SYN, PSH, ACK, URG). В настоящее время их уже девять (с течением времени были добавлены еще три: CWR, ECE, NS). Значения управляющих битов определяют как интерпретацию пакета, так и текущую фазу сеанса TCP: установление соединения, обмен данными (прием и передача данных), закрытие соединения. Таким образом, управление ходом сеанса TCP может быть реализовано через манипулирование значениями управляющих битов.

Поскольку каждый бит имеет только два значения, его можно рассматривать как выход логического элемента или результат вычисления логической функции. Это может быть одна из следующих случайно выбираемых таблично заданных функций: И, ИЛИ, исключающее ИЛИ, И-НЕ, ИЛИ-НЕ, НЕ-ИЛИ. Значения управляющих битов в TCP задаются не произвольно, а во взаимосвязи со значениями других логических битов, образуя совместно с ними семантически значимую последовательность, характеризующую фазу и текущее состояние сеанса TCP. Стратегия соединения логических элементов такова: каждый элемент сопрягается с K случайными элементами, кроме самого себя. Так мы получаем систему из N логических элемен-

тов, связанных друг с другом в среднем K связями, то есть НК-автомат. Его текущее состояние будет представлять собой значимый фрагмент пакета TCP, который должен быть адекватен по отношению к текущему состоянию сеанса и восприниматься таковым противоположной стороной TCP-взаимодействия.

Следовательно, обучение устройства В протоколу, реализуемому устройством А, сводится к формированию одного или нескольких НК-автоматов, способных генерировать структуры данных или TCP-пакеты, образующие адекватные семантически значимые последовательности, распознаваемые в качестве таковых устройством А. Сами структуры данных или их значимые фрагменты кодируются битовыми последовательностями, представляющими собой текущие состояния НК-автомата или набора таких автоматов.

На рисунке 1 приведена структура пакета TCP как битовой последовательности. В незакрашенных ячейках размещены биты, представленные в составе формирующегося НК-автомата, серым цветом выделены поля, значения которых должны быть однозначно определены, в противном случае пакет не будет принят противоположной стороной.

Каждый пакет TCP включает следующее:

- заголовок, содержащий набор полей определяемых протоколом, в зависимости от которых происходит смена состояния сеанса TCP (установка соединения, передача данных, завершение соединения);
- опции, которые являются необязательной частью пакета и служат для расширения возможностей протокола, например, чтобы определить максимальный размер пакета или масштабирование размера окна до величины, большей чем 2^{16} ;
- данные, содержащие полезную передаваемую информацию (для передачи данных соединение должно быть уже установлено).

Для целей исследования блок опций не представляет интереса и будет игнорироваться.

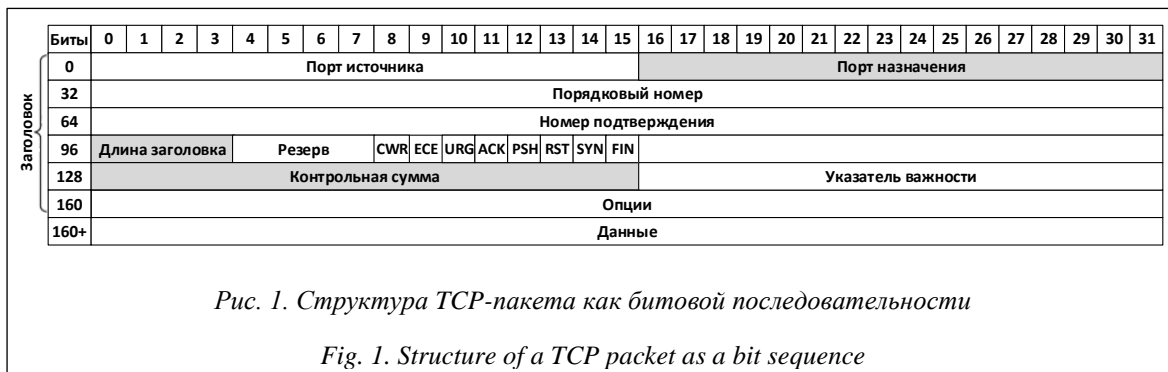


Рис. 1. Структура TCP-пакета как битовой последовательности

Fig. 1. Structure of a TCP packet as a bit sequence

Блок с данными может содержать контрольную фразу, которая в случае успешной передачи будет возвращена TCP-эхо-сервером, что будет являться маркером осуществления информационного обмена. Блок с заголовком представляет наибольший интерес, именно он и будет генерироваться. А наличие блока с данными в сегменте TCP будет определяться управляющим флагом PSH.

В заголовке TCP-пакета содержатся поля, значения которых на протяжении всего сеанса могут принимать лишь единственное верное значение, жестко задаваемое или вычисляемое по строго определенному алгоритму. Это порт назначения, длина заголовка, а также контрольная сумма, вычисляемая как 16-битное дополнение к сумме всех 16-битных слов заголовка. Остальные поля – порт источника, порядковый номер, номер подтверждения, набор флагов, размер окна, указатель важности и резерв – зависят от состояния сеанса TCP (от значений управляющих флагов), поэтому их будет генерировать НК-автомат. Особую важность имеет комбинация значений управляющих флагов, определяющая фазу сеанса и порядок взаимодействия клиента и сервера.

Минимальный размер заголовка TCP-пакета составляет 160 бит (20 байт), размер полей, которые не будут генерироваться с помощью НК-автомата, составляет 36 бит. Таким образом, НК-автомат будет генерировать 124 бита, из которых сформируется пакет TCP. Однако если будет установлен флаг PSH, к сформированному с помощью автомата заголовку будут добавлены и контрольные данные.

Формирование НК-автомата, способного генерировать пакеты TCP в требуемой последовательности, будет происходить эволюционным путем. При этом при выходе траектории смены состояний автомата на аттрактор случайным образом будет изменяться функция произвольно выбранного узла, как это было предложено в [15, 16].

В дальнейшем будем обозначать НК-автомат, генерирующий пакеты TCP и воспроизводящий логику TCP-сеанса, как TCP-НК-автомат.

Описание экспериментов

В процессе экспериментальной проверки предложенного метода исследования разработана программа protogen (генератор протоколов). Это клиентская программа на языке C++, автоматически генерирующая TCP-пакеты с помощью TCP-НК-автомата с целью передачи

данных на сервер. В роли сервера, в паре с которым должна функционировать программа protogen, использовался tcprbin.com – публичный эхо-сервер TCP, подходящий для целей тестирования. Таким образом, программа protogen моделирует работу устройства В, в то время как сервер tcprbin.com моделирует работу устройства А.

Задача программы protogen состояла в том, чтобы экспериментально, через формирование подходящего TCP-НК-автомата, построить правильную с точки зрения протокола TCP последовательность пакетов, необходимую для передачи полезных данных с получением подтверждения от сервера. Трудность этой задачи заключается в том, что для TCP требуется предварительная установка соединения, так называемая процедура тройного рукопожатия: сначала отправляется пакет с управляющим флагом SYN, а потом пакет с флагом ACK, после чего сеанс переходит в режим установленного соединения, и открывается возможность передавать данные пользователя.

Отправляя очередной пакет в адрес сервера, утилита protogen в течение 10 секунд ждет ответа от сервера. При его получении отправленный пакет TCP рассматривается как правильный и добавляется в цепочку пакетов. В дальнейшем при добавлении в цепочку нового TCP-пакета только предшествующие ему в цепочке правильные пакеты будут предварительно отправляться в адрес сервера. Все построенные цепочки хранятся в виде ориентированных графов, описывающих последовательности логически связанных пакетов. Будем называть их графами последовательностей. Каждый такой граф представляет структуру, хранящую сгенерированные автоматами 124-битные блоки данных, на основе которых могут быть построены уже полные варианты пакетов.

В начале каждого эксперимента граф пуст, в дальнейшем каждый найденный правильный пакет помещается в структуру, представляющую граф, в качестве его вершины. Постепенно граф последовательностей накапливает цепочки пакетов в целях формирования адекватной последовательности заголовков, позволяющей установить соединение и передать данные. В соответствии с процедурной составляющей протокола TCP, для успешной передачи полезных данных необходимо передать как минимум два пакета. В одном из них должен содержаться установленный бит SYN, а в другом – PSH и ACK. Таким образом, граф

последовательностей накапливает цепочки заголовков пакетов, стремясь найти их правильную комбинацию. Фактически он представляет собой дерево, каждый узел которого может иметь любое неограниченно большое количество потомков.

Примеры графов последовательностей приведены на рисунке 2, где для каждой вершины каждого графа представлены только значения управляющих флагов, так как они имеют наибольшее влияние на результат. Вершины с усиленным контуром соответствуют пакетам, на которые приходили ответы от сервера. В исходном состоянии каждый граф последовательностей содержит только один корневой элемент. С помощью усиления контура и заливки выделен один из графов последовательностей, описывающий корректное начало TCP-сеанса, которое продолжается фазой передачи данных.

Алгоритм работы программы protogen состоит из следующих шагов.

Шаг 1: выбор цепочки пакетов. Случайным образом выбирается узел из графа последовательностей. Если граф пуст, шаг пропускается.

Шаг 2: генерация блока данных. На каждом шаге своей работы TCP-НК-автомат генерирует 124 бита данных, из которых может быть сформирован заголовок TCP-пакета. Обязательное условие для завершения этого шага состоит в уникальности полученного варианта заполнения заголовка. Если такой вариант уже имеется в графе последовательностей, текущий шаг повторяется. Если происходит достижение аттрактора, то есть присутствует цикл повтора пакетов, выполняется мутация автомата.

Шаг 3: обмен пакетами с сервером. Из выбранной цепочки графа последовательностей отправляются все правильные пакеты, уже присутствовавшие в нем ранее, а в завершение отправляется пакет, сгенерированный на предыдущем шаге.

Шаг 4: обработка ответа. Для каждого отправленного пакета производится ожидание

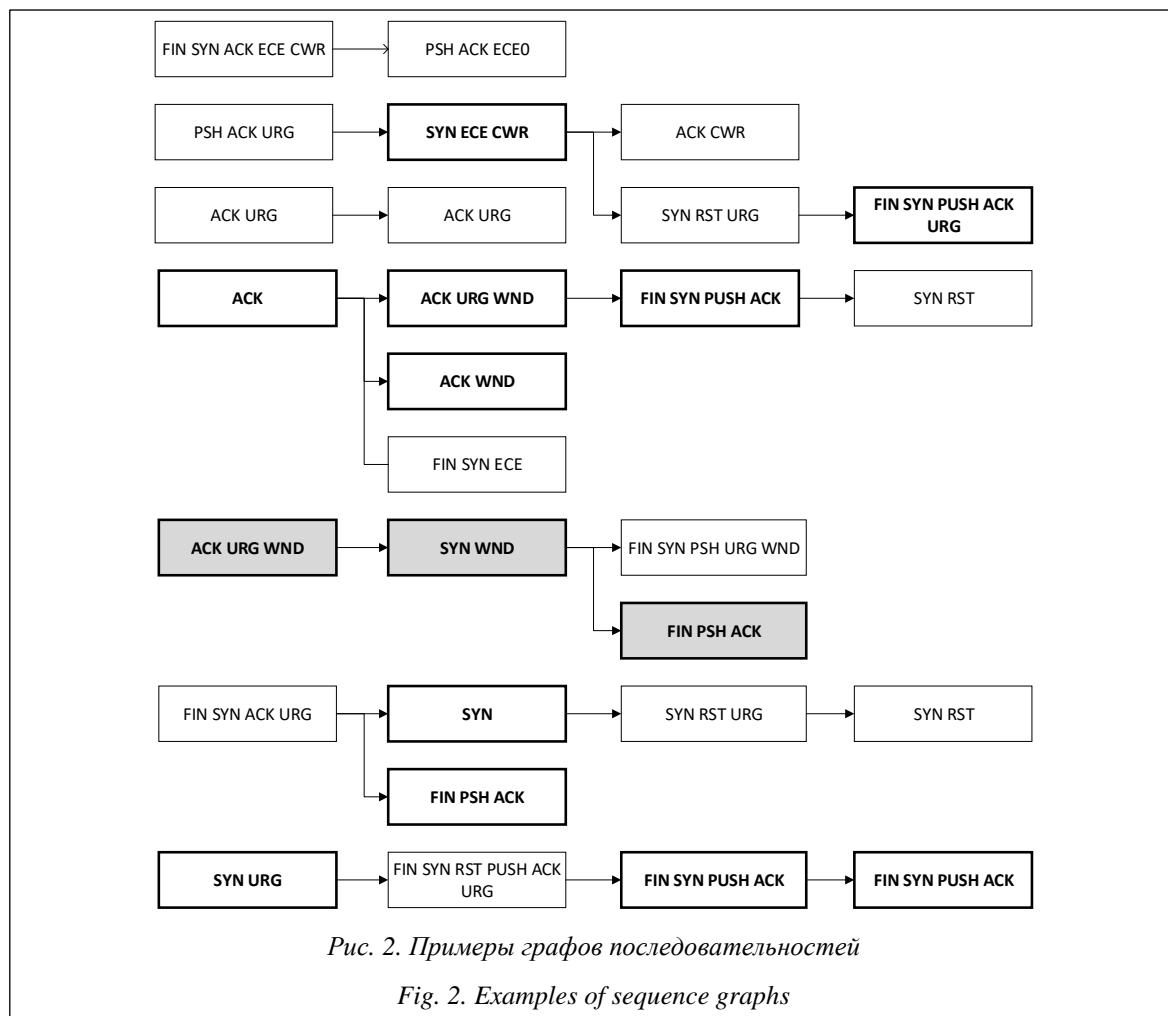


Рис. 2. Примеры графов последовательностей

Fig. 2. Examples of sequence graphs

ответа. Если ответ от сервера приходит, то пакет помечается как правильный, и в граф последовательностей добавляется соответствующий ему узел. Дополнительно в составе ответа проверяется наличие полезных данных и выполняется их анализ. Если данные соответствуют контрольным данным, которые требовалось передать на сервер, это означает, что решение найдено, цель достигнута: TCP-NK-автомат смог воспроизвести фазы установки соединения и обмена данными протокола TCP. В этом случае выполнение алгоритма завершается. Во всех других случаях переходим к шагу 1.

После того как последовательность пакетов информационного обмена выявлена, обучающаяся программа получает возможность выполнять передачу другой полезной информации.

Результаты исследований и их обсуждение

Реализованный алгоритм был задействован при проведении серии экспериментов с использованием программы protogen и сервера tcpbin.com. Каждый эксперимент в серии продолжался в течение 24 часов. При этом утилита protogen искала правильные последовательности TCP-пакетов для передачи данных на сервер при различных значениях $K = 1, 2, 3, 4, 5, 6, 7, 10, 25, 50, 75, 100, 110, 123$. Если в ходе

экспериментов за 30 минут не удавалось найти решение, то есть подобрать корректную последовательность пакетов, граф последовательностей очищался, генерировался новый TCP-NK-автомат, и поиск начинался заново. Итоги экспериментов сведены в таблицу, в скобках указано количество найденных успешных решений.

На основе таблицы был построен график (рис. 3), который иллюстрирует зависимость количества найденных успешных решений от значения величины K .

К результатам экспериментов также относятся и множество построенных графов последовательностей, приведенных на рисунке 2, в том числе и случай графа, описывающего корректный TCP-сеанс.

Анализ приведенных результатов позволяет утверждать следующее:

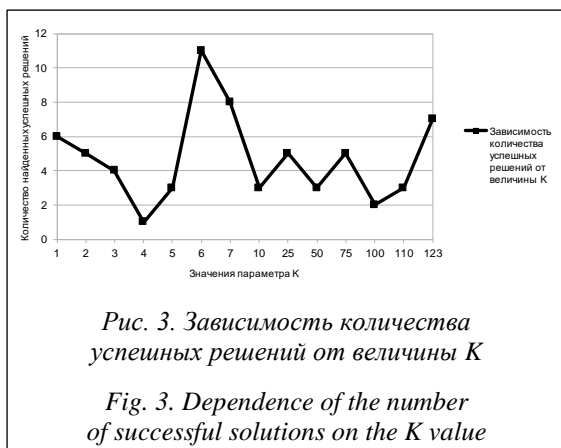
- динамическая эволюционная генерация TCP-NK-автоматов, обеспечивающих генерацию цепочек пакетов, корректных с точки зрения протокола TCP, действительно происходит, то есть исходное предположение, выдвинутое в данной статье, подтвердилось, и обучение устройств новым протоколам с нуля возможно при использовании NK-автоматов и их модификаций;

- наибольший процент успеха наблюдается в случае, когда $K = 6$, хотя, как отмечалось выше, для классических NK-автоматов значе-

Результаты экспериментов

Experimental results

K	Найденные успешные решения от общего числа сгенерированных вариантов, %	Количество случаев достижений аттрактора/ мутаций	Длина аттрактора		Отношение числа правильных пакетов к общему числу	Среднее время нахождения успешного решения, сек.
			средняя	максимальная		
1	11,1 (6)	7 879	13,9	168	9 140 / 69 919	549
2	9,4 (5)	5 717	19,1	700	5 981 / 66 199	605
3	7,7 (4)	130	578,3	15 619	9 887 / 72 539	488
4	2,0 (1)	65	1 096,9	17 523	7 561 / 70 244	623
5	5,9 (3)	1	58 280	58 280	7 664 / 73 562	274
6	18,6 (11)	2	22 783,5	45 443	5 066 / 66 138	421
7	14,3 (8)	-	-	-	8 044 / 73 077	323
10	5,9 (3)	-	-	-	9 994 / 67 946	600
25	9,4 (5)	-	-	-	9 763 / 72 006	592
50	5,9 (3)	-	-	-	5 114 / 68 202	470
75	9,4 (5)	-	-	-	9 686 / 66 711	972
100	4,0 (2)	-	-	-	6 697 / 73 601	301
110	5,9 (3)	-	-	-	6 259 / 69 586	320
123	12,7 (7)	-	-	-	6 448 / 70 944	554



ние $K = 5$ является одним из критических, при нем поведение автоматов начинает соответствовать хаотическому.

Второй вывод наводит на размышления о причине наблюдаемого эффекта. Изначально НК-автоматы использовались именно для исследования переходов от порядка к хаосу и явлений на границе между ними. В случае критического значения $K = 2$ поведение системы, очевидно, более тяготеет к упорядоченному, но это ограничивает способность автомата к генерации новых структур. С другой стороны, при $K \geq 5$ поведение автомата тяготеет к полностью стохастическому и неупорядоченному, но новые структуры генерируются чаще. Отметим также, что согласно таблице при $K = 5$ на поиск решения затрачивается наименьшее время.

Вероятно, что именно на границе полностью хаотического поведения при $K = 6$ способность TCP-НК-автоматов к генерации новых структур оказывается ярко выражена, но при этом еще сохраняется возможность получения упорядоченных, семантически значимых решений.

Возможно также, что при $K = 6$ удастся выйти из ловушки локального оптимума и достичь глобального максимума, чего не происходит при $K = 5$ и меньших значениях.

Эти предположения требуют дальнейших исследований и обоснований.

Выводы и направление дальнейших исследований

В настоящей работе было показано, что для обеспечения слаженной работы интеллектуальных устройств, взаимодействующих друг с другом в составе некоторой единой системы, не обязательно использовать централизованный подход. Следовательно, появляется альтернатива применению брокеров и становится

возможным избежать недостатков, связанных с их использованием: все возрастающей сложности, ресурсоемкости, уязвимости в отношении отказов.

Слаженность работы может достигаться и при децентрализованном подходе при условии, что устройства, входящие в состав системы, способны взаимодействовать друг с другом и приспосабливаться друг к другу. Такая сонстройка возможна в случае взаимного обучения, когда устройства способны обучаться протоколам взаимодействия друг с другом. При этом в ходе процессов самоорганизации должны формироваться поведенческие (алгоритмические) структуры, обеспечивающие требуемое поведение обучающихся устройств. Вопрос о выявлении семантики, стоящей за данными алгоритмическими структурами, остается открытым, и не обсуждался в данном исследовании: принято достаточным, что выявленные структуры рассматриваются устройством А как адекватные и семантически значимые.

В данном исследовании показано, что для обеспечения такого обучения несложных интеллектуальных устройств необязательно опираться на нейросети. Для этого можно использовать другие модели, также обеспечивающие самоорганизацию, а именно НК-автоматы и их модификации. Этот подход проще нейросетей, так как не требует специального подбора типа взаимодействующих элементов и топологии связывающей их сети: и то и другое в случае НК-автоматов универсально и может динамически перенастраиваться в процессе функционирования автоматов.

В работе в качестве среды взаимодействия использовался Интернет, а в роли моделируемого протокола – TCP. Однако это могут быть и другие среды и протоколы и, соответственно, среды и языки взаимодействия устройств. Например, различные прикладные протоколы Интернета, в частности почтовые, или протоколы сетей, отличные от TCP/IP.

Применение полученных результатов позволит легко интегрировать устройства в состав различных коммуникационных сред, так как устройства смогут осваивать новые протоколы. По этой же причине может быть снижена острота проблемы версионной совместимости устройств, выпущенных в разные годы различными производителями. Наконец появляется возможность выполнять обратный инжиниринг протоколов взаимодействия даже в тех случаях, когда исследуемый протокол взаимодействия заранее полностью неизвестен.

То есть спектр применения полученных результатов на практике достаточно широк: их можно использовать для валидации, исследования и генерации сетевых протоколов в различных областях.

Дальнейшие исследования будут направлены на определение влияния величины K и эпистатичности на скорость и эффективность

обучения, то есть на процент удачных решений и длину последовательности состояний перед выходом на аттрактор. Заслуживают исследования и случаи, когда требуемое количество генерируемых бит, то есть количество элементов в модификации НК-автомата, заранее неизвестно и должно быть выявлено в процессе функционирования самого автомата.

Список литературы

1. Song L., Niyato D., Han Z., Hossain E. Machine-to-machine (M2M) communications. In: Wireless Device-to-Device Communications and Networks. Cambridge University Press, Cambridge, 2015, pp. 338–368. doi: 10.1017/SBO9781107478732.012.
2. Oluwatosin A.A., Mohamed O. Machine-to-machine communication: An overview of opportunities. Comput. Networks, 2018, vol. 145, pp. 255–276. doi: 10.1016/j.comnet.2018.09.001.
3. Longo E., Redondi A.E.C. Design and implementation of an advanced MQTT broker for distributed pub/sub scenarios. Comput. Networks, 2023, vol. 224, art. 109601. doi: 10.1016/j.comnet.2023.109601.
4. Patti G., Leonardi L., Testa G., Lo Bello L. PrioMQTT: A prioritized version of the MQTT protocol. Comput. Communications, 2024, vol. 220, pp. 43–51. doi: 10.1016/j.comcom.2024.03.018.
5. Matsumura T., Esaki K., Yang S. et al. Active inference with empathy mechanism for socially behaved artificial agents in diverse situations. Artificial Life, 2024, vol. 30, no. 2, pp. 277–297. doi: 10.1162/artl_a_00416.
6. Wrona Z., Buchwald W., Ganzha M. et al. Overview of software agent platforms available in 2023. Information, 2023, vol. 14, no. 6, art. 348. doi: 10.3390/info14060348.
7. Maragatharajan M., Sureshkumar A., Dhanaraj R.K. et al. Hybrid feed forward neural networks and particle swarm optimization for intelligent Self-Organization in the industrial communication networks. IEEE Open J. Communications Society, 2025, vol. 6, pp. 3816–3833. doi: 10.1109/OJCOMS.2025.3559172.
8. Zhang Y., Soydaner D., Behrad F. et al. Finding closure: A closer look at the gestalt law of closure in convolutional neural networks. Computational Brain & Behavior, 2025, vol. 8. doi: 10.1007/s42113-025-00251-x.
9. Zhang M., Luo X., Wu J. et al. Toward building human-like sequential memory using brain-inspired spiking neural models. IEEE Transactions on Neural Networks and Learning Syst., 2025, vol. 36, no. 6, pp. 10143–10155. doi: 10.1109/TNNLS.2025.3543673.
10. Kauffman S.A. The Origin of Order: Self-Organization and Selection in Evolution. Oxford University Press, 1993, 710 p. doi: 10.1142/9789814415743_0003.
11. Bull L. Nonbinary representations in the NK and NKCS models. Complex Systems, 2022, vol. 31, no. 1, pp. 87–101. doi: 10.25088/ComplexSystems.31.1.87.
12. Bull L. Evolving multi-valued regulatory networks on tunable fitness landscapes. Complex Syst., 2023, vol. 32, no. 3, pp. 289–307. doi: 10.25088/ComplexSystems.32.3.289.
13. Кольчугина Е.А., Стежка В.А. Классификация НК-подобных автоматов // Вестн. ПензГУ. 2024. № 4. С. 70–74.
14. Кольчугина Е.А. Исследование свойств цифровых организмов с помощью НК-подобных автоматов // Вестн. компьютерных и информационных технологий. 2011. № 11. С. 20–24.
15. Бабич М.Ю., Бабич А.М. Аппарат моделирования особого периода функционирования многоагентных специализированных систем // Изв. высших учебных заведений. Поволжский регион. Технич. науки. 2018. № 4. С. 17–27. doi: 10.21685/2072-3059-2018-4-2.
16. Бабич М.Ю., Бабич А.М. Возможности модифицированного НК-автомата Кауфмана при имитации особого периода функционирования систем // XXI век: итоги прошлого и проблемы настоящего плюс. 2018. Т. 7. № 4. С. 110–114.
17. Стежка В.А. Валидация протокола TCP с использованием НК-подобного автомата // Вестн. ПензГУ. 2024. № 4. С. 126–130.

Generation and analysis of communication protocols using NK-automata and their modifications

Elena A. Kolchugina ^{1✉}, Vladimir A. Stezhka ²

¹ Penza State University, Penza, 440026, Russian Federation

² Scientific and Technical Center of JSC “Radiozavod”, Penza, 440039, Russian Federation

For citation

Kolchugina, E.A., Stezhka, V.A. (2025) ‘Generation and analysis of communication protocols using NK-automata and their modifications’, *Software & Systems*, 38(4), pp. 598–607 (in Russ.). doi: 10.15827/0236-235X.152.598-607

Article info

Received: 26.05.2025

After revision: 09.06.2025

Accepted: 20.06.2025

Abstract. This study focuses on methods and algorithms that enable intelligent devices to explore and learn communication protocols in distributed infocommunication systems and environments. The research adopts an evolutionary approach to

constructing NK-automata and their modifications. The objective is to obtain an NK-automaton capable of reproducing TCP protocol packet sequences through its state transitions. These sequence must demonstrate correctness when interacting with publicly available TCP echo servers and remain suitable for data transmission throughout TCP communication sessions. Creating such an automaton signifies that the intelligent device has learned the TCP protocol. To achieve this goal, the authors developed a modification of the classical NK-automaton called TCP-NK-automaton, along with software to simulate its operation. Unlike other NK-automaton variants, the proposed modification models a passive TCP packet data structure rather than an active software architecture. Experiments with the developed software successfully produced automaton instances that generated correct TCP protocol packet sequences, simulating TCP sessions and demonstrating successful learning. The obtained results enable device compatibility through protocol learning without neural networks, eliminate the need for brokers, and enable the investigation and reverse engineering of previously unknown protocols.

Keywords: communication protocol analysis, NK-automata, computational environments, generation, evolutionary device learning

References

1. Song, L., Niyato, D., Han, Z., Hossain, E. (2015) 'Machine-to-machine (M2M) communications', in *Wireless Device-to-Device Communications and Networks*. Cambridge: Cambridge University Press, pp. 338–368. doi: 10.1017/CBO9781107478732.012.
2. Oluwatosin, A.A., Mohamed, O. (2018) 'Machine-to-machine communication: An overview of opportunities', *Comput. Networks*, 145, pp. 255–276. doi: 10.1016/j.comnet.2018.09.001.
3. Longo, E., Redondi, A.E.C (2023) 'Design and implementation of an advanced MQTT broker for distributed pub/sub scenarios', *Comput. Networks*, 224, art. 109601. doi: 10.1016/j.comnet.2023.109601.
4. Patti, G., Leonardi, L., Testa, G., Lo Bello, L. (2024) 'PrioMQTT: A prioritized version of the MQTT protocol', *Comput. Communications*, 220, pp. 43–51. doi: 10.1016/j.comcom.2024.03.018.
5. Matsumura, T., Esaki, K., Yang, S. et al. (2024) 'Active inference with empathy mechanism for socially behaved artificial agents in diverse situations', *Artificial Life*, 30(2), pp. 277–297. doi: 10.1162/artl_a_00416.
6. Wrona, Z., Buchwald, W., Ganzha, M. et al. (2023) 'Overview of software agent platforms available in 2023', *Information*, 14(6), art. 348. doi: 10.3390/info14060348.
7. Maragatharajan, M., Sureshkumar, A., Dhanaraj, R.K. et al. (2025) 'Hybrid feed forward neural networks and particle swarm optimization for intelligent Self-Organization in the industrial communication networks', *IEEE Open J. Communications Society*, 6, pp. 3816–3833. doi: 10.1109/OJCOMS.2025.3559172.
8. Zhang, Y., Soydaner, D., Behrad, F. et al. (2025) 'Finding closure: A closer look at the gestalt law of closure in convolutional neural networks', *Computational Brain & Behavior*, 8. doi: 10.1007/s42113-025-00251-x.
9. Zhang, M., Luo, X., Wu, J. et al. (2025) 'Toward building human-like sequential memory using brain-inspired spiking neural models', *IEEE Transactions on Neural Networks and Learning Syst.*, 36(6), pp. 10143–10155. doi: 10.1109/TNNLS.2025.3543673.
10. Kauffman, S.A. (1993) *The Origin of Order: Self-Organization and Selection in Evolution*. Oxford University Press, 710 p. doi: 10.1142/9789814415743_0003.
11. Bull, L. (2022) 'Nonbinary representations in the NK and NKCS models', *Complex Systems*, 31(1), pp. 87–101. doi: 10.25088/ComplexSystems.31.1.87.
12. Bull, L. (2023) 'Evolving multi-valued regulatory networks on tunable fitness landscapes', *Complex Systems*, 32(3), pp. 289–307. doi: 10.25088/ComplexSystems.32.3.289.
13. Kolchugina, E.A., Stezhka, V.A. (2024) 'Classification of NK-like automata', *Vestn. of PSU*, (4), pp. 70–74 (in Russ.).
14. Kolchugina, E.A. (2011) 'Researching of digital organisms' properties by means of *NK*-like automata', *Vestn. of Comput. and Inform. Tech.*, (11), pp. 20–24 (in Russ.).
15. Babich, M.Yu., Babich, A.M. (2018) 'A modeling apparatus of a special period of Multi-Agent specialized systems functioning', *University Proc. Volga Region. Eng. Sci.*, (4), pp. 17–27 (in Russ.). doi: 10.21685/2072-3059-2018-4-2.
16. Babich, M.Yu., Babich, A.M. (2018) 'Opportunities of the modified Kauffman's NK automaton in imitation of system functioning special period', *XXI Century: Resumes of the Past and Challenges of the Present plus*, 7(4), pp. 110–114 (in Russ.).
17. Stezhka, V.A. (2024) 'Validation of the TCP protocol using an NK-like automaton', *Vestn. of PSU*, (4), pp. 126–130 (in Russ.).

Авторы

Кольчугина Елена Анатольевна¹, д.т.н.,
доцент, профессор, kea_sci@list.ru
Стежка Владимир Андреевич²,
начальник отдела, adevilgio@gmail.com

Authors

Elena A. Kolchugina¹, Dr.Sci. (Engineering),
Associate Professor, Professor, kea_sci@list.ru
Vladimir A. Stezhka²,
Head of Department, adevilgio@gmail.com

¹ Пензенский государственный университет,
г. Пенза, 440026, Россия

² Научно-технический центр АО «Радиозавод»,
г. Пенза, 440039

¹ Penza State University,
Penza, 440026, Russian Federation

² Scientific and Technical Center of JSC "Radiozavod",
Penza, 440039, Russian Federation