

УДК 004.75

МОДИФИЦИРОВАННЫЙ МЕТОД НАЗНАЧЕНИЯ ЗАДАНИЙ В РАСПРЕДЕЛЕННОЙ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЕ ИНТЕРНЕТА ВЕЩЕЙ

Степанова М.В.

*Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет), Москва, e-mail: stepanova@bmstu.ru*

Появление концепции Интернета вещей (ИВ) и создание инфраструктур цифровых решений на основе данной концепции породило как новые возможности, так и новые вызовы для таких цифровых решений. В основе концепции и инфраструктуры Интернета вещей заложен принцип гетерогенной и распределенной инфраструктуры за счет встраивания различных вычислительных модулей в объекты окружающего мира и их взаимной интеграции, их интеграции с иными инфраструктурами. При таком принципе возникает ряд критических проблем при реализации распределенной вычислительной системы на основе объектов ИВ, прежде всего, связанных с оптимальным управлением несколькими потоками данных и команд, а также с распределением вычислительной нагрузки на узлы инфраструктуры Интернета вещей. При этом также необходимо учитывать, что состав, месторасположение узлов и состояние распределенной вычислительной системы Интернета вещей могут динамически изменяться вследствие внутренних и внешних процессов. Первоначальная идея концепции Интернета вещей заключалась в саморегулирующейся распределенной инфраструктуре, в которой каждый узел взаимодействует с остальными напрямую, без централизованного посредника. Однако на сегодняшний день в основном преобладают архитектуры построения инфраструктур Интернета вещей по централизованному типу вычислений, что делает актуальной проблему оптимальной балансировки нагрузки и самостоятельного интеллектуального взаимодействия узлов на распределенной вычислительной системе Интернета вещей. В данной статье предлагаются такие механизмы и метод решения данного вопроса на основе подхода машинного обучения с подкреплением, которые позволяют распределенной вычислительной системе Интернета вещей быстро реагировать на внутренние процессы и окружающую среду в режиме реального времени в неопределенных условиях.

Ключевые слова: Интернет вещей, машинное обучение, обучение с подкреплением, распределенные вычислительные системы, кластеризация

A MODIFIED METHOD OF ASSIGNING TASKS IN A DISTRIBUTED COMPUTING SYSTEM OF THE INTERNET OF THINGS

Stepanova M.V.

*Bauman Moscow State Technical University (National Research University),
Moscow, e-mail: stepanova@bmstu.ru*

The emergence of the concept of the Internet of Things and the creation of infrastructures for digital solutions based on this concept has generated both new opportunities and new challenges for such digital solutions. The concept and infrastructure of the Internet of Things is based on the principle of a heterogeneous and distributed infrastructure by embedding various computing modules into objects of the surrounding world and their mutual integration, and their integration with other infrastructures. With this principle, a number of critical problems arise in the implementation of a distributed computing system based on IoT objects, first of all, associated with the optimal control of several data and command streams, as well as with the distribution of the computational load on the nodes of the IoT infrastructure. It should also be borne in mind that the composition, location of nodes, and the state of the distributed computing system of the Internet of Things can dynamically change due to internal and external processes. The original idea behind the concept of the Internet of Things was a self-regulating distributed infrastructure in which each node interacts directly with the others, without a centralized intermediary. However, today the architecture of building IoT infrastructures based on the centralized type of computing is mainly dominated, which makes the issue of optimal load balancing and independent intelligent interaction of nodes on the distributed computing system of the IoT urgent. This article proposes such mechanisms and a method for solving this issue based on a reinforcement learning approach, which allows the distributed computing system of the Internet of Things to quickly respond to internal processes and the environment in real time under uncertain conditions.

Keywords: internet of things, machine learning, reinforcement learning, distributed computing systems, clustering

В основе Интернета вещей (ИВ) заложена концепция динамической автономной инфраструктуры [1], что делает ключевым аспектом самостоятельное взаимодействие устройств в инфраструктуре ИВ. Вопрос о данном взаимодействии был первоначально обозначен в 2009 г. [2], затем был повторно рассмотрен в 2015 г. [1], однако до сих пор его рассмотрение и решение является актуальными [3].

Ведущиеся исследования по созданию механизмов интеллектуального взаимодействия устройств ИВ направлены, в целом, на сокращение времени взаимодействия устройств и повышение качества обслуживания [4–6].

Самым распространенным типом взаимодействия устройств ИВ является централизованный тип архитектуры, характеризующийся наличием узла, выделенного

в качестве центрального и обеспечивающего: консолидацию всех данных и заданий; управление данными и заданиями; управление устройствами [7, 8]. При данном типе устройство отправляет данные или задание на центральный узел, а центральный узел производит необходимую обработку и принимает дальнейшее решение по управлению. Поскольку вся логика управления и хранения данных находится на центральном узле, то это создает ряд проблем для инфраструктуры ИВ (характерных для централизованной архитектуры), таких как:

- высокая вычислительная нагрузка на центральный узел (или облако);
- управление гетерогенными устройствами и их мониторинг;
- масштабируемость при подключении новых устройств, пользователей, приложений;
- управление сетевым трафиком;
- управление устройствами при их перемещении на местности;
- постоянное изменение параметров передающей среды;
- необходимость в поддержке множества сетевых протоколов.

Для снижения влияния указанных проблем на всю инфраструктуру ИВ может быть использован децентрализованный подход, в частности предложенный в структуре граничных вычислений (Edge Computing) [8], при котором нагрузка с центрального узла переносится на удаленные или пограничные узлы сети. Таким образом, за счет распределения логики управления и хранения данных с центрального узла на пограничные узлы можно получить следующие преимущества:

- сократить сетевой трафик внутри инфраструктуры ИВ;
- снизить вычислительную нагрузку на центральном узле.

Перечисленные преимущества граничных вычислений перед централизованным типом архитектуры являются значимыми, однако не представляются в полной мере достаточными для рассмотрения граничных вычислений в качестве основного подхода в целях обеспечения децентрализованной архитектуры для ИВ, поскольку не позволяют решить остальные проблемы. Следовательно, описанные проблемы централизованного типа архитектуры полностью не решаются, в первую очередь, за счет того, что принцип взаимодействия устройств и пограничных узлов сети основан на централизованном типе архитектуры. А именно: во взаимодействие между устройством и основным центральным узлом добавляется пограничный узел сети (например, шлюз), который, в свою очередь,

является центральным узлом для устройств и одновременно устройством для основного центрального узла. Таким образом, архитектура по типу граничных вычислений, в конечном итоге, испытывает те же проблемы масштабирования, перегрузки узлов, регулирования трафика и т.д. Более того, в [9, 10] отмечаются критические проблемы граничных вычислений, особенно в отношении задержек, управляемости, надежности, выборочного получения данных и выполнения задач пользователя.

Таким образом, для распределенной вычислительной системы ИВ должны быть предусмотрены подходы и механизм, снижающие негативное влияние централизованного типа архитектуры (в том числе и подобной архитектуре граничных вычислений) и обеспечивающие ее автономность.

В данной работе представлены механизм и метод, которые не имеют цели заменить существующие архитектурные решения и подходы, но позволяют повысить надежность и масштабируемость инфраструктуры ИВ, в частности распределенной вычислительной системы на основе объектов ИВ, сделав ее более гибкой и автономной.

Материалы и методы исследования

1. Архитектура шаблона взаимодействия инфраструктуры Интернета вещей

В [11] определены четыре основных шаблона (паттерна) взаимодействия интеллектуальных объектов:

- устройство – устройство;
- устройство – облако;
- устройство – шлюз;
- обратный паттерн совместного использования данных.

Шаблоны могут быть реализованы и использованы как по отдельности, так и в комбинациях, которые приведены на рис. 1.

В зависимости от конкретной бизнес-задачи, специфики или сферы деятельности можно выделить следующие возможные комбинации шаблонов взаимодействия:

- облако – шлюз – устройство;
- центральный узел – шлюз – устройство;
- центральный узел – устройство.

Данные взаимодействия обобщены в соответствии с используемыми подходами [12, 13]. Однако для реализации механизмов и методов назначения заданий и обеспечения автономности устройств в распределенной вычислительной системе Интернета вещей, применимых к различным сферам деятельности, необходимо учитывать аспекты использования каждого шаблона взаимодействия [11]. С данной целью рассматривается обобщенный шаблон взаимодействия в инфраструктуре

Интернета вещей (рис. 1). Для его реализации необходимо определить множество условий и предположений:

– на каждом узле должно быть установлено программное обеспечение, такое как виртуальная машина (VM, VM – virtual machine), которая совместима с конкретным устройством и уменьшает влияние специфики аппаратного обеспечения от конкретного производителя. Данное условие является необходимым для реализации обобщенной архитектуры взаимодействия инфраструктуры ИВ;

На начальном этапе работы алгоритма осуществляется формирование кластеров устройств с помощью метода машинного обучения без учителя – метод нечеткой кластеризации С-средних (Fuzzy C-Means), который основан на данных о сходстве множеств и является алгоритмом мягкой кластеризации. Принцип мягкой кластеризации для ИВ подразумевает, что одно устройство может быть отнесено не к одному кластеру, а способно принадлежать к нескольким множествам кластеров одновременно, что крайне важно при оценке

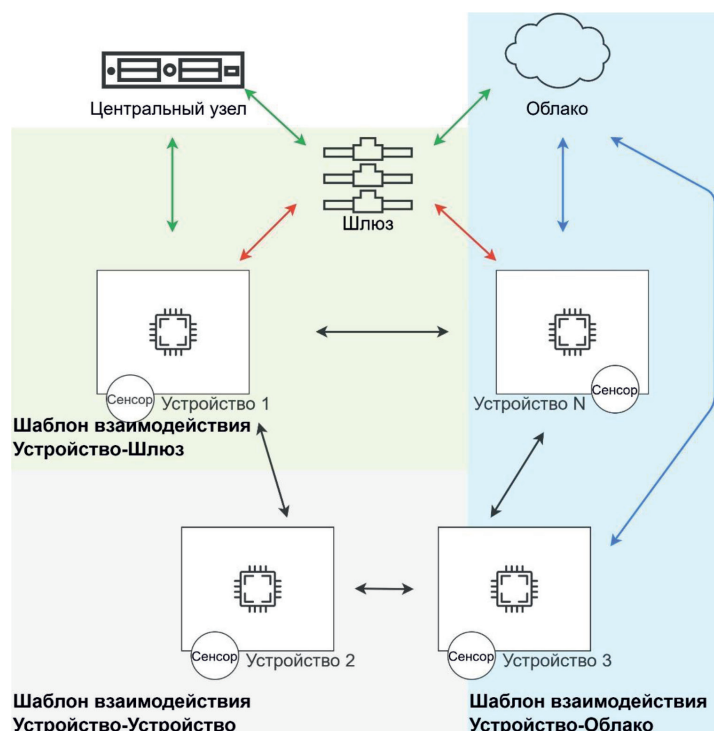


Рис. 1. Обобщенный шаблон взаимодействия в инфраструктуре Интернета вещей

– устройства должны поддерживать любую реализацию физического уровня модели OSI (Open Systems Interconnection);

– взаимодействие внутри инфраструктуры ИВ должно быть реализовано по клиент-серверной модели, причем назначения ролей сервера и клиента могут меняться;

– взаимодействие внутри инфраструктуры ИВ должно обеспечиваться посредством одного и того же протокола.

2. Кластеризация устройств Интернета вещей

В данной работе реализован модифицированный алгоритм назначения заданий в распределенной вычислительной системе Интернета вещей, функционирование которого состоит из нескольких этапов.

принятия решений, поскольку инфраструктура ИВ, как правило, подвержена динамическим изменениям, таким как: изменение местоположения устройств, изменение технологии передачи данных, изменение типа решаемого задания и т.д.

Для задачи кластеризации устройств необходимо определить следующие требования к алгоритму FCM (Fuzzy C-Means):

– количество кластеров является постоянным и конечным;

– центроид кластера рассчитывается следующим образом [14]:

$$u_{ik} = \frac{1}{\sum_{j=1}^c \left(\frac{d_{ik}}{d_{jk}} \right)^{2/(m-1)}}, \quad (1)$$

$$v_k = \frac{\sum_{i=1}^n u_{ik}^m x_i}{\sum_{i=1}^n u_{ik}^m}, \quad (2)$$

где d_{ik} – евклидово расстояние от устройства ИВ до центроида, i – номер устройства, x_i – устройство ИВ, k – номер кластера, m – индекс нечеткости. Индекс нечеткости определяет степень нечеткости: чем больше значение индекса, тем выше степень нечеткости. Обычно значение индекса принимается равным: $m = 2$ [15]. Однако величина индекса зависит от распределения величин кластера [16].

Для учета динамических изменений в распределенной вычислительной системе Интернета вещей и реализации алгоритма по назначению заданий в ней задается модель устройства ИВ. Данная модель позволяет устройствам обмениваться информацией, что позволяет провести оценку возможности совместной работы и принятия решений.

3. Модель устройства Интернета вещей

Устройства ИВ выпускаются множеством производителей, которые могут предоставляться в рамках как закрытой, или проприетарной, инфраструктуры ИВ, так и открытой. Кроме того, существует множество комплексных решений ИВ, закрытых в принципе для разработки и манипуляций со стороны пользователя. Однако следует отметить, что наблюдается тенденция роста использования открытых, или свободных, решений за счет стандартизации устройств, протоколов, используемого открытого программного обеспечения и типовой аппаратуры.

Независимо от многообразия устройств ИВ и их принадлежности к открытой или закрытой инфраструктуре общими характеристиками, которыми они обладают, являются:

- информация о состоянии (State) устройства: характеристики, режимы работы, потребляемая энергия; вычислительная способность [17] и т.д.;

- информация о расположении (Location) устройства: статическое или динамическое позиционирование устройства, удаленность устройства от других элементов инфраструктуры ИВ;

- информация о состоянии сети связи (Network): скорость передачи данных, пропускная способность канала, стабильность канала.

Для реализации назначения заданий вводится дополнительный параметр Label, который указывает тип решаемого задания. Таким образом, общий вид модели устрой-

ства (D, Device) описывается в виде интегральной характеристики:

$$D = (Label, State, Location, Network). \quad (3)$$

4. Оценка устройств Интернета вещей и назначение заданий

Для определения устройства ИВ, которое наиболее подходит под назначенное ему задание и наиболее эффективно выполнит полученное задание, используется машинное обучение с подкреплением. Обучение с подкреплением – это разновидность машинного обучения, которая основывается на получении реакции от окружающей среды в режиме реального времени либо в режиме, близком к реальному времени.

Для реализации алгоритма по оценке устройств ИВ и назначения заданий с помощью обучения с подкреплением требуется выполнение следующих условий:

- необходимость хранить актуальную информацию об устройствах в реальном времени либо в режиме, близком к реальному времени. Состояние и поведение устройств ИВ могут быстро изменяться, причем не всегда это можно предсказать;

- устройства ИВ ограничены в таких ресурсах, как: память, вычислительная способность, режимы работы и т.п.

Обучение с подкреплением использует понятия агента и среды. Агент – сущность, имеющая собственное состояние и набор выполняемых действий: а среда – это все, что окружает агента и дает реакцию (вознаграждение) на действия агента.

В роли агента ранее рассматривался центральный узел, а среды – устройства ИВ [18]. Использование машинного обучения с подкреплением позволяет обеспечить взаимодействие между центральным узлом и непосредственно устройством ИВ, что дает возможность реализовать распределенные вычисления на основе инфраструктуры ИВ [18, 19].

В данной работе рассматривается модифицированная архитектура, при которой в роли агента выступает устройство ИВ, которое оценивает другие устройства ИВ; в роли окружающей среды – устройства ИВ, которые оцениваются для распределения заданий. Таким образом, каждое устройство ИВ может быть и агентом, и средой. В данном случае каждое устройство может делегировать задание другому, что обеспечивает назначение заданий в распределенной вычислительной системе Интернета вещей, повышает его автономность и степень интеллектуального взаимодействия устройств.

Модифицированная архитектура, рассматриваемая в данной работе, показана на рис. 2.

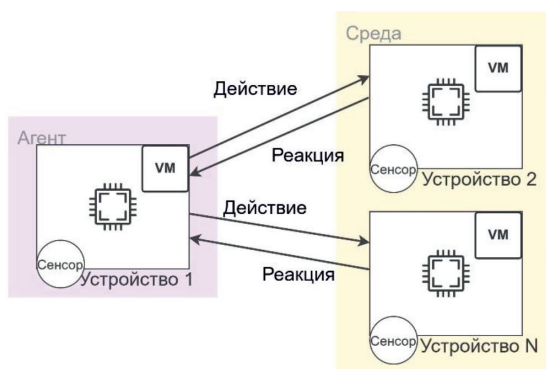


Рис. 2. Модель агент – окружающая среда

У алгоритма обучения с подкреплением есть два режима работы, которые в данном случае будут модифицированы следующим образом:

- этап исследования: устройство отправляет запросы другим устройствам в кластере. Если устройства в кластере активны, то они подготавливают и формируют обратное сообщение – сигнал вознаграждения;

- этап эксплуатации: устройство оценивает полученное вознаграждение и выби-

рает соответствующее устройство кластера для отправки задания на выполнение.

5. Модифицированный алгоритм взаимодействия инфраструктуры ИВ

Разработка модифицированного алгоритма осуществлялась на основе ранее проведенных исследований [18].

Модифицированный метод взаимодействия элементов инфраструктуры ИВ показан на рис. 3.

Указанный подход сокращает количество транзакций между устройством и главным узлом, в отличие от централизованного типа архитектуры ИВ, когда все транзакции должны быть адресованы главному узлу.

Следующий алгоритм описывает взаимодействие в инфраструктуре ИВ с учетом предложенной модификации.

Шаг 0. Установка начальных значений состояния элементов инфраструктуры ИВ. Главный узел посылает запросы всем устройствам для получения информации о них в соответствии с (1). Главный узел сохраняет список устройств и связанную с ними информацию. Главный узел вычисляет кластеры устройств на основе близости их интегральных характеристик.

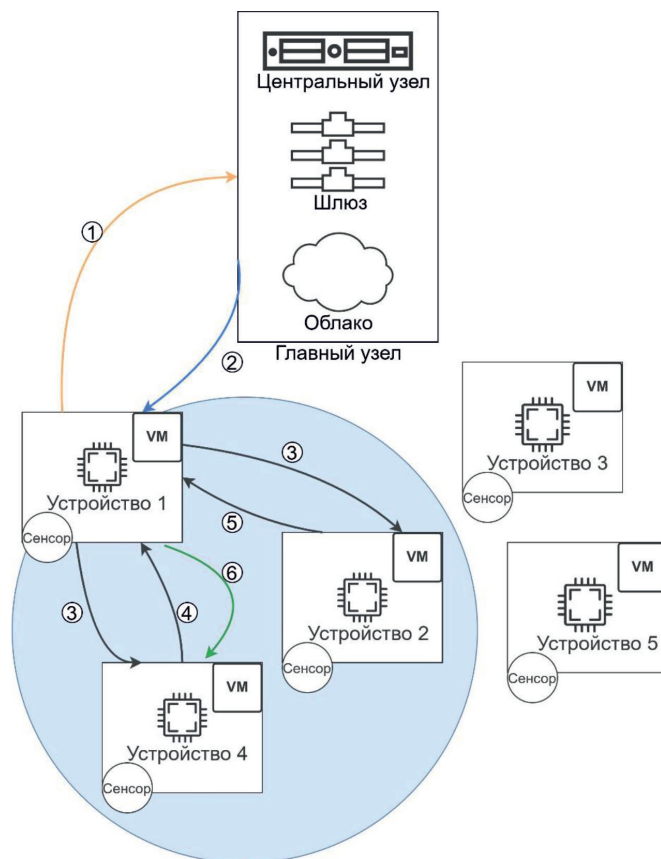


Рис. 3. Взаимодействие в инфраструктуре ИВ

Шаг 1. Каждое устройство посылает запрос на главный узел для получения информации о своей принадлежности к кластеру и о других устройствах того же кластера.

Шаг 2. Главный узел отвечает устройствам сообщением, в котором передается принадлежность к кластеру. В случае изменения местоположения устройство возвращается на Шаг 1.

Шаг 3. Устройство, которому необходимо выполнить задание, посылает запросы всем устройствам в том же кластере, если одно из них способно выполнить задание требуемого типа.

Шаг 4. Запрошенные устройства отправляют ответное сообщение в соответствии с информацией (3).

Шаг 5. Выполняется оценка полученных сообщений от устройств тем устройством, которое имеет задание на исполнение.

Шаг 6. Задание отправляется на соответствующее устройство для обработки. Информация о назначенном устройстве обновляется.

Шаг 7. В случае возникновения проблем (например, задача не может быть соответствующим образом обработана устройствами из кластера) распределяющее устройство сообщает об этом главному узлу и посылает запрос на переквалификацию к другому устройству или на включение в новый кластер – Шаг 1.

6. Результаты исследований

Для решаемой в данной работе задачи выбрано значение индекса нечеткости $m = 4$.

Однако определение оптимального числа кластеров само по себе является открытой темой исследований [20]. В данной работе значение количества кластеров фиксировано и равно 5. На рис. 4 показаны результаты процесса кластеризации алгоритмом FCM, реализованным на Шаге 0.

Левая часть рис. 4 показывает поле распределения точек данных характеристик исследуемых устройств ИВ. Каждое устройство ИВ сформировало свои характеристики в соответствии с (1). Для каждой точки данных, описывающей каждую характеристику устройства ИВ, рассчитывается близость к центроидам кластера по евклидову расстоянию. Значения центроидов итеративно пересчитываются. На правой части рис. 4 показаны результаты кластеризации устройств ИВ, где видно, что устройства с близкими характеристиками образуют кластер.

Для анализа поведения модифицированного алгоритма распределения заданий необходимо сделать следующие допущения:

- количество устройств ИВ и узлов в инфраструктуре ИВ постоянно;
- значение параметра ϵ постоянное и имеет значение 0,1;
- значение вознаграждения изменяется в диапазоне [0; 1];
- количество узлов в каждом кластере переменное.

Данные допущения введены согласно проведенным ранее исследованиям [18].

На рис. 5 показано изменение значений вознаграждений в пределах одного кластера в зависимости от изменения количества устройств ИВ в кластере. Устройства сохраняют свои значения вознаграждений до нового запроса устройства, которое назначает задание.

После ряда временных шагов устройство отправляет запросы ко всем устройствам в кластере и выполняет обновление сигналов вознаграждений. Распределяющее устройство проводит оценку и отправляет задание на то устройство, которое выполнит его наилучшим способом.

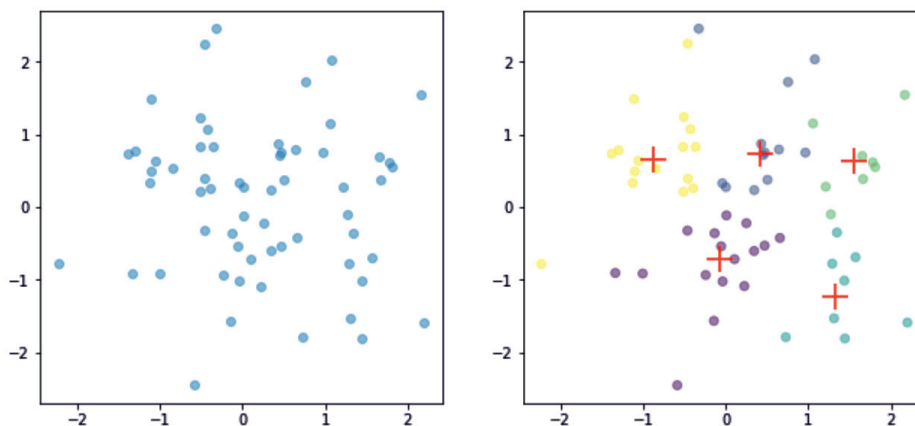


Рис. 4. Кластер устройств ИВ

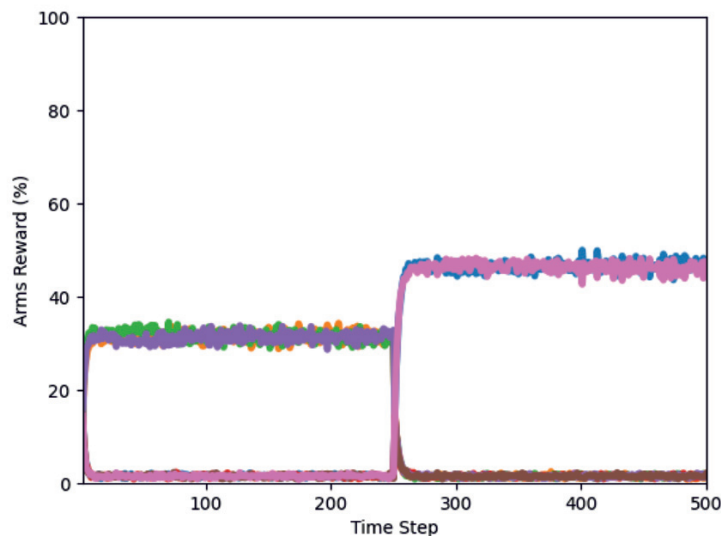


Рис. 5. Выбор наилучшего устройства (от 5 до 7 устройств ИВ, 2000 симуляций)

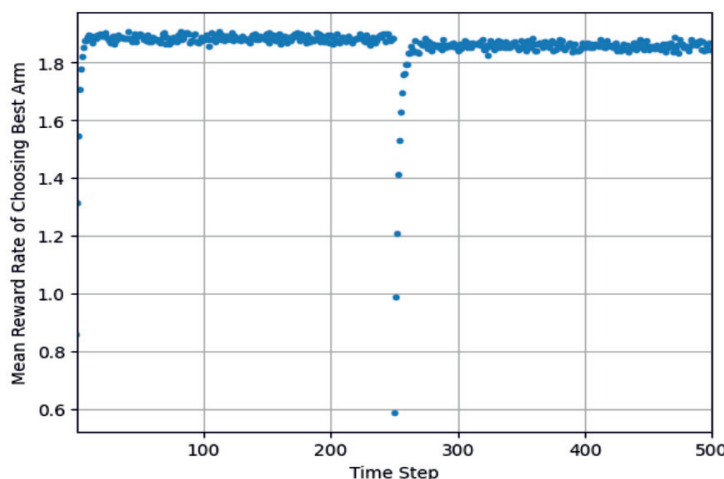


Рис. 6. Среднее значение вознаграждения при выборе лучшего устройства ИВ (от 5 до 7 устройств ИВ, 2000 симуляций)

На рис. 6 показано среднее значение вознаграждения для случаев в 7 и 7 устройств (левая и правая части рисунка). При меньшем количестве устройств ИВ наибольшее среднее значение вознаграждения достигается быстрее, чем с большим количеством устройств. Более того, среднее значение вознаграждения выше, если сами значения вознаграждений выше, что определяется поведением алгоритма [21].

Заключение

Принципы автономности и саморегулирования объектов Интернета вещей, которые изначально заложены в концепции

Интернета вещей, остаются открытыми вопросами для проведения исследований. Существующие подходы взаимодействия элементов инфраструктуры основаны на централизации управления. Однако простота такого решения приводит к возникновению проблем масштабирования и перегрузки каналов связи и объектов Интернета вещей. Более того, архитектура инфраструктуры Интернета вещей, реализованная по централизованному принципу, не позволяет достичь первоначальной идеи Интернета вещей: самоуправление объектов. Таким образом, предложенный новый подход к реализации объектов Интернета вещей

позволяет устранить ряд существующих проблем взаимодействия. В работе предложены обобщенный шаблон взаимодействия устройств Интернета вещей, модель устройства и инфраструктуры Интернета вещей, а также модифицированный метод назначения заданий по устройствам ИВ на основе машинного обучения с подкреплением с использованием алгоритма Fuzzy C-Means. Такой подход позволяет динамически контролировать узлы инфраструктуры ИВ и управлять ими, снижает сетевой трафик и вычислительную нагрузку центрального узла, позволяет учитывать специфику узлов Интернета вещей и дает возможность поддерживать гибкость системы в реальном времени при неопределенных условиях среды.

Список литературы

1. Eleftherakis G., Pappas D., Lagkas T., Rousis K., Pournovski O. Architecting the IoT Paradigm: A Middleware for Autonomous Distributed Sensor Networks. Hindawi Publishing Corporation. International Journal of Distributed Sensor Networks. 2015. DOI: 10.1155/2015/139735.
2. Rozik A.S., Tolba A.S., El-Dosuky M.A. Design and Implementation of the Sense Egypt Platform for Real-Time Analysis of IoT Data Streams. Advances in Internet of Things. 2016. Vol. 6. No. 4.
3. Zaidan A.A., Zaidan B.B., Qahtan M.Y., Albahri O.S., Albahri A.S., Alaa M., Jumaah F.M., Talal M., Tan K.L., Shir W.L., Lim C.K. A survey on communication components for IoT-based technologies in smart homes. Telecommun Syst 69. 2018. P. 1–25. DOI: 10.1007/s11235-018-0430-8.
4. Tanganelli G., Vallati C., Mingozzi E. Ensuring Quality of Service in the Internet of Things. 2018. DOI: 10.1007/978-3-319-58190-3_9.
5. Bigi P., Tansu A., Jiong J., Marimuthu P. Distributed Real-Time IoT for Autonomous Vehicles. IEEE Transactions on Industrial Informatics. 2018. P. 1–1. DOI: 10.1109/TII.2018.2877217.
6. Vermesan O., Bahr R., Ottella M., Serrano M., Karlsen T., Wahlstrom T., Erik Sand H., Ashwathnarayan M., Micaela Troglia M. Internet of Robotic Things Intelligent Connectivity and Platforms. 2020. DOI: 10.3389/frobt.2020.00104.
7. Celic L., Magjarevic R. Seamless connectivity architecture and methods for IoT and wearable devices. Automatika. 2020. Vol. 61. P. 21–34. DOI: 10.1080/00051144.2019.1660036.
8. Mocnej J., Seah W.K.G., Pekar A., Zolotova I. Decentralised IoT Architecture for Efficient Resources Utilisation. IFAC-PapersOnLine. 2018. Vol. 51. P. 168–173. DOI: 10.1016/j.ifacol.2018.07.148.
9. Parikh S., Dave D., Patel R., Doshi N. Security and Privacy Issues in Cloud, Fog and Edge Computing. Procedia Computer Science. 2019. Vol. 160. P. 734–739. DOI: 10.1016/j.procs.2019.11.018.
10. Mostafavi S., Dawlatnazar M., Paydar F. Edge Computing for IoT: Challenges and Solutions. Journal of Communications Technology, Electronics and Computer Science. 2019.
11. Tschofenig H., ARM Ltd., Arkko J., Thaler D., McPherson D. Architectural Considerations in Smart Object Networking. [Electronic resource]. URL: <https://tools.ietf.org/html/rfc7452> (date of access: 03.08.2021).
12. Field and cloud edge gateways. [Electronic resource]. URL: <https://docs.microsoft.com/en-us/azure/architecture/example-scenario/iot/field-cloud-edge-gateways> (date of access: 03.08.2021).
13. Securing IoT applications. [Electronic resource]. URL: <https://developer.ibm.com/articles/iot-trs-secure-iot-solutions3/> (date of access: 03.08.2021).
14. Ren M., Liu P., Wang Z., Jing Y. A Self-Adaptive Fuzzy c-Means Algorithm for Determining the Optimal Number of Clusters. Computational Intelligence and Neuroscience. 2016. Vol. 2016. DOI: 10.1155/2016/2647389.
15. Hammah R.E., Curran J.H. Fuzzy cluster algorithm for the automatic identification of joint sets. International Journal of Rock Mechanics and Mining Sciences. 1998. Vol. 35. P. 889–905.
16. Zhou K., Yang S., Fuzzifier Selection in Fuzzy C-Means from Cluster Size Distribution Perspective. Informatica 2019. Vol. 30. No. 3. P. 613–628. DOI: 10.15388/Informatica.2019.221.
17. Uskenbayeva R.K., Kuandykov A.A., Cho Y.r., Kozhamzharova D.K., Baimuratov O.A.S. Main Principles of Task Distribution in Multi-Agent Systems and Defining Basic Parameters. 14th International Conference on Control, Automation and Systems (ICCAS 2014). 2014. P. 1471–1474. DOI: 10.1109/ICCAS.2014.6987793.
18. Степанова М.В., Еремин О.Ю. Назначение заданий узлам распределенной системы платформы Интернета вещей на основе машинного обучения с подкреплением // Автоматизация процессов управления. 2021. № 1 (63). С. 27–33. DOI: 10.35752/1991-2927-2021-1-63-27-33.
19. Еремин О.Ю., Степанова М.В. Распределение заданий по узлам вычислительной системы на платформе Интернета вещей на основе машинного обучения // Динамика сложных систем. 2020. Т. 14. № 2. С. 84–92.
20. Verma C., Illés Z., Stoffová V. Towards Real-Time Homogeneity and Heterogeneity in Student's Beliefs. International Conference on Decision Aid Sciences and Application (DASA). 2020. P. 99–103. DOI: 10.1109/DASA51403.2020.9317040.
21. Caelen O., Bontempi G. Improving the exploration strategy in bandit algorithms. In: Learning and Intelligent Optimization. Number 5313 in LNCS, Springer. 2008. P. 56–68.