

Минимизация задержек при взаимодействии граничных устройств с использованием кластеризации в сетях VANETs

 Павел Владимирович Плотников✉, pavplot@gmail.com

 Андрей Геннадьевич Владыко, vladuko@sut.ru

Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича,
Санкт-Петербург, 193232, Российская Федерация

Аннотация: В настоящей работе рассматривается подход к моделированию информационных процессов и оценке их эффективности в автомобильных самоорганизующихся сетях (VANETs). Приводятся результаты численного моделирования взаимодействия граничных устройств в традиционной конфигурации и с использованием кластерного подхода к разбиению стационарных вычислительных устройств. Показано, что в транспортной сети исследуемый подход позволяет существенно увеличить процент обслуженных устройств, тем самым свести к минимуму общую задержку вычислений, что значительно повысит эффективность работы динамической системы.

Ключевые слова: VANET, кластеризация, граничные вычисления

Источник финансирования: работа подготовлена в рамках исполнения Государственного контракта № П33-1-26/9.

Ссылка для цитирования: Плотников П.В., Владыко А.Г. Минимизация задержек при взаимодействии граничных устройств с использованием кластеризации в сетях VANETs // Труды учебных заведений связи. 2022. Т. 8. № 2. С. 6–13. DOI:10.31854/1813-324X-2022-8-2-6-13

Minimizing Delays in the Interaction of Edge Devices Using Clustering in VANETs

 Pavel Plotnikov✉, pavplot@gmail.com

 Andrei Vladyko, vladuko@sut.ru

The Bonch-Bruevich Saint Petersburg State University of Telecommunications,
St. Petersburg, 193232, Russian Federation

Abstract: The article describes an approach to modeling information processes and evaluating their effectiveness in vehicular ad-hoc networks (VANETs). The results of numerical simulation of the interaction of edge devices in the traditional configuration and using a clustering approach to the partitioning of road-side units are presented. It is shown that in the transport network, this approach can significantly increase the percentage of serviced devices and minimize the overall delay of computing, which will significantly increase the efficiency of the dynamic system.

Keywords: VANET, clustering techniques, edge computing

Funding: the work was supported under the State Contract no. П33-1-26/9.

For citation: Plotnikov P., Vladyko A. Minimizing delays in the interaction of edge devices using clustering in VANETs. *Proc. of Telecom. Universities*. 2022;8(2):6–13. (in Russ.) DOI:10.31854/1813-324X-2022-8-2-6-13

Введение

В мобильных коммуникационных средах с помощью решений для облачных вычислений и широкополосной системы связи большинство вычислительных задач может быть перенесено для удаленного выполнения в облако. Несмотря на то, что данный процесс ускоряется с каждым новым поколением сетей связи, разгрузка в удаленное облако подразумевает существенную задержку. Чтобы решить проблему для приложений чувствительных к задержкам, были представлены мобильные граничные вычисления (MEC, *аббр. от англ. Mobile Edge Computing*) [1]. На сегодняшний день MEC также стали новой парадигмой и для интеллектуальных транспортных систем (ИТС) [2–4]. Внедряя автомобильные граничные вычисления (VEC, *аббр. от англ. Vehicular Edge Computing*) в автомобильные самоорганизующиеся сети (VANETs, *аббр. от англ. Vehicular Ad-hoc Networks*), поставщики услуг могут обеспечить сервисы со сверхмалыми задержками, поскольку инструменты граничных вычислений разворачивают ресурсы на стороне, близкой к пользовательскому оборудованию.

Архитектура VEC опирается на коммуникационную инфраструктуру и услуги, предоставляемые посредством граничных устройств и технологий LTE/5G. К граничным устройствам можно отнести бортовые вычислительные устройства транспортных средств (OBU, *аббр. от англ. On-Board Unit*), придорожные вычислительные устройства (RSU, *аббр. от англ. Road-Side Unit*) и MEC-серверы. Отметим, что устройства OBU имеют небольшие вычислительные ресурсы и ресурсы хранения, вычислительная мощность устройств RSU также относительно ограничена из-за размера вычислительных задач и высокого спроса на реакцию системы в реальном времени. Когда мобильные устройства транспортных средств (ТС) генерируют задачи, они могут подключаться к придорожным устройствам, выбирая подходящее время и подходящие MEC-серверы для разгрузки задач.

Тем не менее, каким образом и в каком порядке эффективно разгружать вычислительные мощности, остается не полностью решенной задачей. Поскольку MEC-серверы работают на границе сети радиодоступа, подключаясь к RSU и выполняя с их помощью задачи по передаче, их зона обслуживания может быть ограничена радиопокрытием RSU. Из-за высокой мобильности движущиеся ТС могут проходить через несколько RSU и MEC-серверов и разгружать свои вычислительные задачи на любой сервер, к которому у них есть доступ, при этом нагрузка на них сильно меняется во времени из-за неравномерности количества подключаемых пользователей.

Таким образом, ключевой проблемой является то, как ТС определяет решение о разгрузке с помощью подходящего граничного устройства, чтобы

свести к минимуму общую задержку вычислений. Дополнительным узким местом остается соединение и кэширование на уровне граничных устройств OBU – RSU [5].

В ответ на эту проблему в данной работе мы предлагаем математическую модель и ее численное решение для системы взаимодействия граничных устройств в традиционной конфигурации размещения RSU. Кроме этого, проведено улучшение работы модели с применением кластерного подхода. Оценка эффективности строится на основе вычисления процента необработанных запросов, сформированных ТС при их мобильности. Проведено сравнение традиционной и кластеризованной моделей. Результаты показывают, что каждая из предложенных моделей может быть эффективно реализована в мобильных узлах и позволит значительно сократить общее ожидаемое время обработки задач по совершенствованию организации и алгоритмического обеспечения функционирования VANETs.

Постановка задачи

Предположим, что модель города задана Манхэттенской моделью мобильности в прямоугольной метрике (L_1 -метрика). Расстояние между двумя точками с координатами $A(x_1; y_1)$ и $B(x_2; y_2)$ в данной модели вычисляется по формуле:

$$d(A, B) = |x_2 - x_1| + |y_2 - y_1|.$$

Система городских автомобильных дорог образует сетку с горизонтальными и вертикальными элементами, при этом на каждой улице организовано двустороннее движение. Набор из V ТС движется по городу, где каждое ТС выбирает кратчайший маршрут для перемещения от точки старта до определенного заранее пункта назначения, при этом перед стартом каждое ТС генерирует свой индивидуальный маршрут.

Рассмотрим набор RSU S , расположенных на середине каждого отрезка между двумя перекрестками. Каждое RSU $s \in S$ оснащено ограниченным кэшем размера Z_s , который используется для кэширования обрабатываемых данных. Будем считать, что каждое RSU имеет зону покрытия в виде круга диаметром L_s .

Скорости движения ТС представляют собой поток независимых, одинаково распределенных по усеченному нормальному закону случайных величин. При этом будем считать, что скорость u каждого ТС на участке дороги (внутри области действия одного RSU) лежит в диапазоне от наперед заданных минимальной до максимальной скорости, т. е.:

$$u_{\min} \leq u \leq u_{\max}.$$

Функция плотности распределения скорости u со средним значением μ и дисперсией σ^2 задается формулой [6]:

$$\varphi(u) = \frac{2}{\sigma\sqrt{2\pi}} e^{-\frac{(u-\mu)^2}{2\sigma^2}} \cdot \frac{1}{\operatorname{erf}\left(\frac{u_{\max}-\mu}{\sigma\sqrt{2}}\right) - \operatorname{erf}\left(\frac{u_{\min}-\mu}{\sigma\sqrt{2}}\right)},$$

где $\operatorname{erf}()$ – функция распределения нормального закона или функция Лапласа.

Взаимодействие ТС с RSU осуществляется посредством обмена блоками данных, например, информацией о загрузенности участка дороги на пути следования. Пусть каждое ТС и каждое RSU формируют информацию в виде вектора M некоррелированных элементов данных $M = \{1, 2, \dots, m\}$. Предположим, что все элементы данных имеют одинаковый размер C байтов. Каждое RSU обслуживает подключенное ТС со скоростью α_s байт в секунду.

Отметим, что, так как местоположение и количество ТС в рассматриваемой модели динамически меняются во времени, с некоторой периодичностью необходимо обновлять данные в RSU, запрашивая их с МЕС-сервера. Пусть время обработки запроса от конкретного ТС по блоку данных i составляет $t_i^{(0)}$, если данные по этой ячейке памяти актуальны и $t_i^{(0)} + t_i^{(1)}$, где $t_i^{(1)}$ – время, требующееся на обновление устаревших данных, если эти данные неактуальны и требуется их актуализация.

Будем считать, что каждое RSU может обслуживать ТС последовательно, при этом, если ТС покидает область покрытия RSU, взаимодействие прекращается, а ТС может подключиться к следующему RSU, в случае если оно свободно. Можно считать, что RSU по окончании работы с каким-либо ТС может начать обслуживание следующего мгновенно, т. е. за нулевое время, без задержек. При этом те блоки данных, которые не были полностью обработаны в ходе информационном взаимодействии с предыдущим ТС, теряются (удаляются из кэша).

Таким образом, если считать скорость ТС на участке дороги, обслуживаемом RSU s равной u м/с, а зона покрытия этого участка имеет длину L_s метров, то максимальное число запросов, которые может обработать RSU от конкретного ТС, вычисляется по формуле:

$$k = \left\lfloor \frac{L_s}{u \cdot \alpha_s} \right\rfloor.$$

Если часть запрашиваемой информации требует обновления, общее время взаимодействия составит:

$$\sum_{i=1}^k (t_i^{(0)} + \gamma \cdot t_i^{(1)}) \leq \frac{L_s}{u},$$

где $\gamma = 1$, если данные устарели, и $\gamma = 0$, если данные не требуют актуализации, для каждой пары ТС – RSU.

Общее описание модели

Рассмотрим движение ТС в транспортной городской сети как динамическую систему. Пусть городская сеть представляет собой систему пересекающихся под прямым углом дорог, как показано на рисунке 1. RSU расположены в середине каждого участка дороги между двумя соседними перекрестками (на рисунке для RSU использовано графическое обозначение , при этом будем считать, что область покрытия каждого RSU – это весь участок дороги между двумя перекрестками.

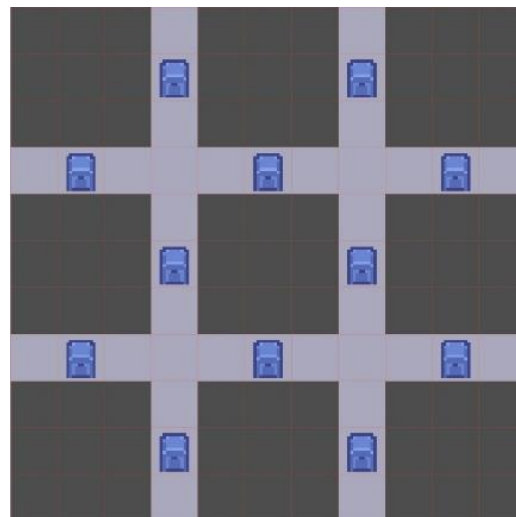


Рис. 1. Модель городской сети

Fig. 1. Scheme of Road Network

В ходе моделирования осуществляется генерация случайного потока ТС, который обрабатывается системой. Если RSU свободно, то ТС начинает взаимодействие с ним. В случае отсутствия контакта (RSU занято или не функционирует), ТС движется в соответствии с ранее намеченным маршрутом с постоянной скоростью на этом участке дороги и производит поиск свободного RSU. После нахождения свободного RSU ТС начинает взаимодействие с ним. Прежде чем оптимизировать сеть: выбирать оптимальное расположение RSU, динамически перестраивать маршруты следования ТС в соответствии с дорожной обстановкой, менять скорость ТС в зависимости от скорости движения потока и т. д., необходимо в первую очередь определить, может ли созданная система в данных условиях работать эффективно, т. е. справляться с потоком запросов, создаваемым ТС.

Сделаем некоторые дополнительные упрощения, позволяющие построить работоспособную модель. Пусть каждое ТС v_i запрашивает заранее определенное количество блоков информации $k^{v_i} \leq m$ в диапазоне от $k_1^{v_i}$ до $k_2^{v_i}$, обозначаемое $M_{v_i} = \{m_1^{v_i}, m_2^{v_i}, \dots, m_{k^{v_i}}^{v_i}\}$. Если при движении по маршруту запрошенный блок данных обработан, ответ от RSU получен, то считаем, что число необработанных блоков информации уменьшается на

один. Такой подход в дальнейшем позволит интерпретировать полученные данные как информацию о дорожной обстановке на ближайшем перекрестке, с целью выбора оптимальной скорости движения ТС или перестройки текущего маршрута для уменьшения интенсивности дорожного трафика на загруженном перекрестке.

При анализе пропускной способности сети будем полагать, что размер кэш-памяти Z_s каждого RSU совпадает с максимальным обрабатываемым объемом информации St . Это позволит не следить за переполнением памяти и выборочным удалением данных, уже хранящихся на RSU. При этом стоит обратить внимание на следующий важный аспект: если n ТС последовательно обращаются к одному блоку данных, то, интерпретируя информацию как дорожную ситуацию на перекрестке, OBU этих ТС направят на этот перекресток все ТС, образуя затор. Поэтому необходимо ввести некоторые ограничения F на количество обращений l к каждому блоку информации до ее принудительного обновления, будем считать $l \leq F$. При этом введем дополнительно ограничение T на «время жизни» t_{m_i} данных m_i , положим $t_{m_i} < T$.

В качестве оценки эффективности работы системы будем вычислять процент необработанных запросов, сформированных на начальном этапе ТС.

Результаты моделирования

Численное моделирование проводилось с использованием языка программирования Python. Генерация случайного потока ТС с заданным распределением осуществлялась с использованием кроссплатформенного механизма Godot Engine [7]. Рассматривается дорожная сеть, представленная на рисунке 2. Полагаем при моделировании, что начальная позиция, с которой стартуют все ТС, находится в центре карты (на рисунке это место обозначено знаком ) , такое положение позволяет наглядно отследить движение ТС и изменение количества отправляемых запросов. RSU обозначены на схеме синими блоками (как на рисунке 1 – ). Конечная точка маршрута каждого ТС генерируется отдельно и расположена на границе приведенной на рисунке 2 области.

Для проведения численных расчетов примем:

$$V = 250, m = 10, L_s = 200, k_1^{v_i} = 4, k_2^{v_i} = 10, \\ t_i^{(0)} = 0,1 \text{ с}, t_i^{(1)} = 1 \text{ с}, F = 3, T = 10 \text{ с}.$$

Результаты моделирования представлены в таблице 1. Частичное представление результатов обусловлено тем обстоятельством, что полный вариант данной таблицы включает 250 строк. В нашем случае, без потери общности, ограничимся приведением лишь 10 % сгенерированных данных.

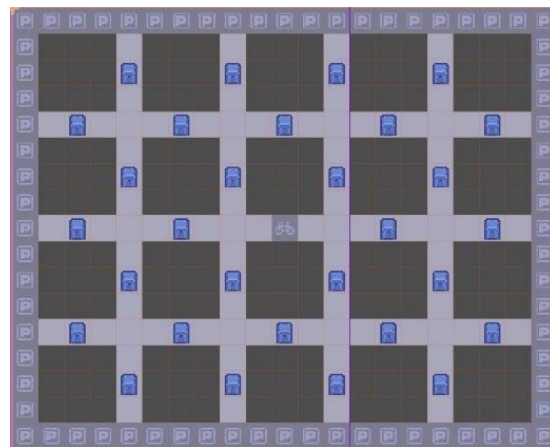


Рис. 2. Моделируемая дорожная сеть

Fig. 2. Model of Road Network

ТАБЛИЦА 1. Результаты численного моделирования потока ТС в дорожной сети

TABLE 1. The Results of Numerical Simulation of Vehicle Flow in the Road Network

VehicleID	Speed	TotalRequests	UnsolvedRequests
18738	38,940022	5	5
17778	42,629593	5	0
17918	35,35902	4	0
18693	38,399227	6	6
18308	34,725357	4	4
18768	38,740234	4	0
18243	33,849865	4	4
17963	30,485712	6	0
17713	42,053638	6	0
18803	39,415653	4	4
18883	32,336693	5	0
18368	27,50292	7	0
18603	35,260983	7	0
18908	36,077919	6	6
18618	34,487629	6	5
18813	31,438875	7	7
18343	32,947659	4	0
18673	27,594978	4	0
18858	33,758209	6	6
20013	35,346455	5	1
20198	35,17841	4	4
20378	34,48555	6	3
20513	34,510269	5	5
18678	32,479774	6	0
20658	34,893787	7	3

Условные обозначения:

VehicleID – ID транспортного средства;

Speed – средняя скорость движения автомобиля на протяжении всего маршрута;

TotalRequests – количество запросов, созданных ТС;

UnsolvedRequests – количество созданных, но не обработанных запросов ТС.

Для получения более точной оценки процента необработанных запросов проведем моделирование 10 раз и найдем среднее значение процента необработанных запросов. Результаты численного моделирования показывают, что при генерировании потока ТС 11 и более раз, процент необработанных запросов, поступающих от ТС, меняется незначительно. 10-ти моделирований достаточно, для получения необходимой точности при оценке эффективности рассматриваемого подхода. Результаты представлены в таблице 2.

ТАБЛИЦА 2. Результаты численного моделирования потока ТС в традиционной конфигурации размещения RSU

TABLE 2. The Numerical Simulation Results of Vehicle Flow in the Traditional Configuration of Edge Devices

Iteration	TotalRequests	UnsolvedRequests	Percent
1	1381	518	37,5
2	1361	545	40,0
3	1381	473	34,3
4	1345	481	35,6
5	1386	531	38,0
6	1390	533	38,3
7	1367	526	38,5
8	1358	490	36,1
9	1344	529	39,4
10	1374	508	37,0

Условные обозначения:

Iteration – номер итерации;

TotalRequests – общее количество запросов, созданных ТС;

UnsolvedRequests – совокупное количество не обработанных RSU-запросов;

Percent – процент необработанных RSU-запросов.

Средний процент необработанных запросов (расчет выполнялся по формуле среднего арифметического) составляет 37,5 %. Процент необработанных запросов достаточно большой, следовательно, значительная часть ТС движется бесконтрольно, не корректируя свой маршрут в зависимости от дорожной обстановки. Это потенциально приводит к снижению эффективности организации дорожного движения. Для снижения количества необработанных запросов применим кластерный подход к решению рассматриваемой задачи.

Схема кластерного кэширования

Подробный обзор применения кластерного подхода к решению задач в сетях VANETs приведен в работах [8–10]. Отдельные решения по кластеризации получены в работах [11–13]. Так, в исследовании [11] применен алгоритм кластеризации FOREL с целью минимизации задержки доставки данных. Рассматривается составляющая задержки, характеризующаяся временем распространения электромагнитного или оптического сигнала в динамической системе. В качестве среды распространения

рассматриваются оптические кабельные линии связи. Однако данная архитектура не позволяет учесть проблемы, связанные с тем, что кабельные линии прокладываются вдоль существующих дорог, которые не всегда могут быть заданы отрезком прямой линии. В работе [12] авторы предложили использовать спектральную кластеризацию. Объектами изучения являются связи ТС – инфраструктура (V2I, аббр. от англ. Vehicle-to-Infrastructure) и ТС – ТС (V2V, аббр. от англ. Vehicle-to-Vehicle) в сценарии многополосной автомагистрали, где покрытие обеспечивается сетью RSU. Предлагается механизм оптимального выбора ТС, имеющих качественную связь с RSU, что позволяет разгрузить ТС с низким показателем сигнал-шум. Приведены численные результаты моделирования, демонстрирующие значительное улучшение общей производительности динамической системы. Авторами работы [13] была изучена модель города, в которой основное внимание уделялось влиянию упреждающего кэширования на некластеризованные и кластеризованные схемы. Показано, что кластерная схема кэширования является тем более эффективной, чем больше RSU входит в кластер. В этой связи является важным проверить, позволит ли кластеризация свести к минимуму общую задержку вычислений на граничных устройствах.

В настоящей работе в схеме кластерного кэширования предполагается, что каждая группа RSU может составлять кластер со свойством совместного использования памяти. Другими словами, элементы кластера взаимодействуют друг с другом, чтобы обслуживать любое подключенное ТС в зоне покрытия этого кластера с временной задержкой меньше, чем у отдельного элемента. В качестве примера рассмотрим кластер RSU $\{s_1, s_2, s_3\}$, как показано на рисунке 3.

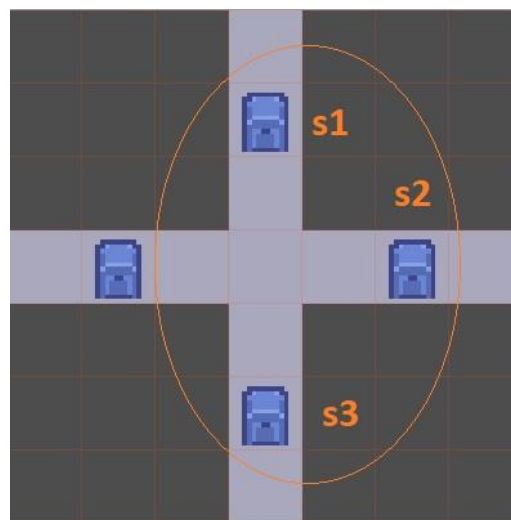


Рис. 3. Схема кластеризации RSU

Fig. 3. Clustering Scheme for RSU

Когда ТС v_i , подключенное к RSU s_1 , запрашивает элемент данных m_j , RSU s_1 проверяет свой локальный кэш. Если запрошенный элемент данных найден, то он доставляется через $t_i^{(0)}$ секунд. В противном случае RSU s_1 проверяет кэш-память других элементов кластера, т. е. s_2 и s_3 , чтобы доставить запрошенный элемент данных за τ секунд. Если запрашиваемый файл не найден ни на одном из элементов кластера, то данные доставляются на ТС за $t_i^{(1)}$ секунд.

Проведем кластеризацию, объединяя RSU двумя способами: по парам (рисунок 4) и по четверкам (рисунок 5).

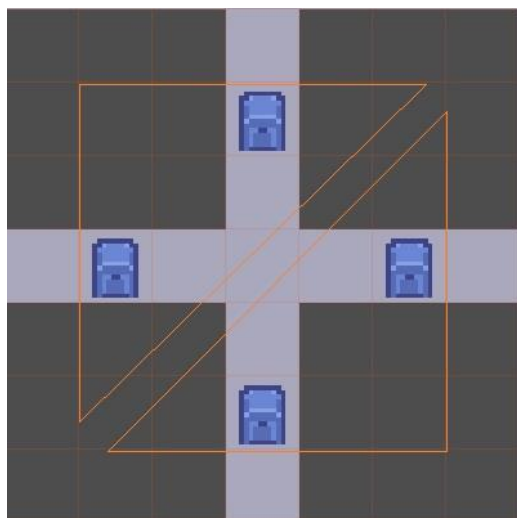


Рис. 4. Схема кластеризации по 2 RSU в кластере

Fig. 4. Clustering Scheme for RSU. Two Road-Side Units Combined in one Cluster

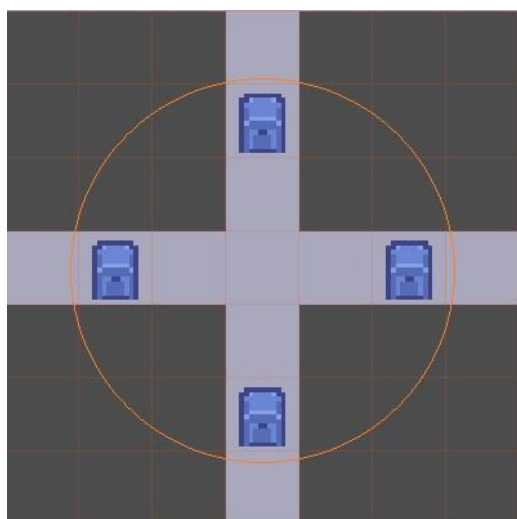


Рис. 5. Схема кластеризации по 4 RSU в кластере

Fig. 5. Clustering Scheme for RSU. Four Road-Side Units Combined in one Cluster

Выбор данных элементов кластера обусловлен техническими характеристиками RSU, с силу ограничений на область покрытия.

Результаты моделирования с использованием кластерного подхода

Численные результаты моделирования будем получать с теми же начальными данными:

$$V = 250, m = 10, L_s = 200, k_1^{v_i} = 4, k_2^{v_i} = 10, \\ t_i^{(0)} = 0,1 \text{ с}, \tau = 0,5 \text{ с}, t_i^{(1)} = 1 \text{ с}, F = 3, T = 10 \text{ с}.$$

Как и ранее, проведем моделирование 10 раз для каждого вида кластеризации (по 2 и по 4 RSU в кластере). В каждом из случаев найдем среднее значение процента необработанных запросов. Результаты расчетов представлены в таблицах 3 и 4. Средний процент необработанных запросов в случае использования кластерного подхода составляет 7,5 % и 1,9 %, соответственно, что указывает на значительное сокращение общего времени обработки задач на граничных устройствах в сравнении с традиционной моделью, рассмотренной ранее.

ТАБЛИЦА 3. Результаты численного моделирования потока ТС в кластеризованной конфигурации размещения RSU, объединенных в блоки по 2 RSU

TABLE 3. The Numerical Simulation Results of Vehicle Flow with Clustering Technique. Two Road-Side Units Combined in One Cluster

Iteration	TotalRequests	UnsolvedRequests	Percent
1	1372	87	6,3
2	1374	72	5,2
3	1375	153	11,1
4	1374	117	8,5
5	1334	80	6,0
6	1372	112	8,2
7	1362	94	6,9
8	1397	86	6,2
9	1387	111	8,0
10	1346	116	8,6

ТАБЛИЦА 4. Результаты численного моделирования потока ТС в кластеризованной конфигурации размещения RSU, объединенных в блоки по 4 RSU

TABLE 4. The Numerical Simulation Results of Vehicle Flow with Clustering Technique. Four Road-Side Units Combined in One Cluster

Iteration	TotalRequests	UnsolvedRequests	Percent
1	1369	21	1,5
2	1380	20	1,6
3	1388	28	2,0
4	1397	7	0,5
5	1371	19	1,4
6	1381	59	4,2
7	1362	12	0,9
8	1369	30	2,2
9	1373	43	3,1
10	1388	24	1,7

Выводы

Распространение современных технологий на автомобильном транспорте, таких как подключенные транспортные средства (CV, аббр. от англ. Connected Vehicles), автономные транспортные средства (AV, аббр. от англ. Autonomous Vehicles), требует решения ряда задач по повышению эффективности процесса взаимодействия ТС с элементами стационарной инфраструктуры. Проблему составляет то, что при высокой интенсивности движения пропускная способность системы обработки и передачи данных не всегда позволяет обеспечить безопасность дорожного движения.

Решение указанной проблемы осуществляется двумя основными способами. Во-первых, улучшением технических характеристик используемого оборудования, в частности построением сетей, основанных на новом стандарте мобильной связи 5G. Во-вторых, совершенствованием организации и алгоритмического обеспечения функционирования VANETs. В данной работе рассмотрен подход к повышению эффективности процессов взаимодействия с использованием второго способа.

В частности, определена постановка задачи, сформирована модель и выполнено численное моделирование взаимодействия граничных устройств

в традиционной конфигурации размещения элементов стационарной инфраструктуры VANETs, в том числе с применением кластерного подхода. Предложено объединять отдельные стационарные вычислительные устройства в кластеры.

Доказано, что кластеризация существенно повышает эффективность работы модели движения: если при традиционной организации работы RSU среднее количество необработанных запросов от ТС составляло 37,5 %, то при применении кластерной схемы это количество снизилось до 1,9–7,5 %. Следовательно, кластеризация позволяет повысить эффективность работ модели и существенно сократит задержку в вычислениях при взаимодействии граничной системы OBU-RSU.

Направлением дальнейших исследований является анализ оптимального размещения RSU с учетом дорожной обстановки, динамическая перестройка маршрута следования ТС на основе информации, полученной от RSU, анализ оптимального разбиения RSU на кластеры для максимизации пропускной способности транспортной сети с учетом технических ограничений, накладываемых на сетевое оборудование.

Список источников

1. Abbas N., Zhang Y., Taherkordi A., Skeie T. Mobile Edge Computing: A Survey // IEEE Internet of Things Journal. 2018. Vol. 5. Iss. 1. PP. 450–465. DOI:10.1109/JIOT.2017.2750180
2. Brehon-Grataloup L., Kacimi R., Beylot A.L. Mobile edge computing for V2X architectures and applications: A survey // Computer Networks. 2022. Vol. 206. ID 108797. DOI:10.1016/j.comnet.2022.108797
3. Vladiko A., Khakimov A., Muthanna A., Ateya A.A., Koucheryavy A. Distributed Edge Computing to Assist Ultra-Low-Latency VANET Applications // Future Internet. 2019. Vol. 11. Iss. 6. ID 128. DOI:10.3390/fi11060128
4. Vladiko A., Elagin V., Spirikina A., Muthanna A., Ateya A.A. Distributed Edge Computing with Blockchain Technology to Enable Ultra-Reliable Low-Latency V2X Communications // Electronics. 2022. Vol. 11. Iss. 2. ID 173. DOI:10.3390/electronics11020173
5. Dziyauddin R.A., Niyato D., Luong N.C., Izhar M.A.M., Hadhari M., Daud S. Computation Offloading and Content Caching Delivery in Vehicular Edge Computing: A Survey // arXiv preprint. 2019. DOI:10.48550/arXiv.1912.07803
6. Abuelenin S.M., Abul-Magd A.Y., Empirical study of traffic velocity distribution and its effect on VANETs connectivity // Proceedings of the International Conference on Connected Vehicles and Expo (ICCVE, Vienna, Austria, 03–07 November 2014). IEEE, 2014. PP. 391–395. DOI:10.1109/ICCVE.2014.7297577
7. Godot Engine // Godot. 2022. URL: <https://godotengine.org> (дата обращения 30.06.2022)
8. Cooper C., Franklin D., Ros M., Safaei F., Abolhasan M. A Comparative Survey of VANET Clustering Techniques // IEEE Communications Surveys and Tutorials. 2017. Vol. 19. Iss. 1. PP. 657–681. DOI:10.1109/COMST.2016.2611524
9. Bali R.S., Kumar N., Rodrigues J.J. Clustering in vehicular ad hoc networks: taxonomy, challenges and solutions // Vehicular Communications. 2014. Vol. 1. Iss. 3. PP. 134–152. DOI:10.1016/j.vehcom.2014.05.004
10. Khan Z., Koubaa A., Fang S., Lee M.Y., Muhammad K. A Connectivity-Based Clustering Scheme for Intelligent Vehicles // Applied Sciences. 2021. Vol. 11. Iss. 5. ID 2413. DOI:10.3390/app11052413
11. Paramonov A., Khayyat M., Chistova N., Muthanna A., Elgendy I.A., Koucheryavy A., et al. An Efficient Method for choosing Digital Cluster Size in Ultralow Latency Networks // Wireless Communications and Mobile Computing. 2021. Vol., 2021. ID 9188658. DOI:10.1155/2021/9188658
12. Luoto P., Bennis M., Pirinen P., Samarakoon S., Horneman K., Latva-aho M. Vehicle Clustering for Improving Enhanced LTE-V2X Network Performance // Proceedings of the European Conference on Networks and Communications (EuCNC, Oulu, Finland, 12–15 June 2017). IEEE, 2017. DOI:10.1109/EuCNC.2017.7980735
13. AlNagar Y., Hosny S., El-Sherif A.A. Proactive Caching for Vehicular Ad hoc Networks Using The City Model // Proceedings of the Wireless Communications and Networking Conference Workshop (WCNCW, Marrakech, Morocco, 15–18 April 2019). IEEE, 2019. DOI:10.1109/WCNCW.2019.8902590

References

1. Abbas N., Zhang Y., Taherkordi A., Skeie T. Mobile Edge Computing: A Survey. *IEEE Internet of Things Journal*. 2018;5(1): 450–465. DOI:10.1109/JIOT.2017.2750180
2. Brehon-Grataloup L., Kacimi R., Beylot A.L. Mobile edge computing for V2X architectures and applications: A survey. *Computer Networks*. 2022;206:108797. DOI:10.1016/j.comnet.2022.108797
3. Vladyko A., Khakimov A., Muthanna A., Ateya A.A., Koucheryavy A. Distributed Edge Computing to Assist Ultra-Low-Latency VANET Applications. *Future Internet*. 2019;11(6):128. DOI:10.3390/fi11060128
4. Vladyko A., Elagin V., Spirkin A., Muthanna A., Ateya A.A. Distributed Edge Computing with Blockchain Technology to Enable Ultra-Reliable Low-Latency V2X Communications. *Electronics*. 2022;11(2):173. DOI:10.3390/electronics11020173
5. Dziyauddin R.A., Niyato D., Luong N.C., Izhar M.A.M., Hadhari M., Daud S. Computation Offloading and Content Caching Delivery in Vehicular Edge Computing: A Survey. *arXiv preprint*. 2019. DOI:10.48550/arXiv.1912.07803
6. Abuelenin S.M., Abul-Magd A.Y., Empirical study of traffic velocity distribution and its effect on VANETs connectivity. *Proceedings of the International Conference on Connected Vehicles and Expo, ICCVE, 03–07 November 2014, Vienna, Austria*. IEEE; 2014. p.391–395. DOI:10.1109/ICCV.2014.7297577
7. Godot. *Godot Engine*. 2022. URL: <https://godotengine.org> [Accessed 30th June 2022]
8. Cooper C., Franklin D., Ros M., Safaei F., Abolhasan M. A Comparative Survey of VANET Clustering Techniques. *IEEE Communications Surveys and Tutorials*. 2017;19(1):657–681. DOI:10.1109/COMST.2016.2611524
9. Bali R.S., Kumar N., Rodrigues J.J. Clustering in vehicular ad hoc networks: taxonomy, challenges and solutions. *Vehicular Communications*. 2014;1(3):134–152. DOI:10.1016/j.vehcom.2014.05.004
10. Khan Z., Koubaa A., Fang S., Lee M.Y., Muhammad K. A Connectivity-Based Clustering Scheme for Intelligent Vehicles. *Applied Sciences*. 2021;11(5):2413. DOI:10.3390/app11052413
11. Paramonov A., Khayyat M., Chistova N., Muthanna A., Elgendy I.A., Koucheryavy A., et al. An Efficient Method for Choosing Digital Cluster Size in Ultralow Latency Networks. *Wireless Communications and Mobile Computing*. 2021;2021:9188658. DOI:10.1155/2021/9188658
12. Luoto P., Bennis M., Pirinen P., Samarakoon S., Horneman K., Latva-aho M. Vehicle Clustering for Improving Enhanced LTE-V2X Network Performance. *Proceedings of the European Conference on Networks and Communications, EuCNC, 12–15 June 2017, Oulu, Finland*. IEEE; 2017. DOI: 10.1109/EuCNC.2017.7980735
13. AlNagar Y., Hosny S., El-Sherif A.A. Proactive Caching for Vehicular Ad hoc Networks Using The City Model. *Proceedings of the Wireless Communications and Networking Conference Workshop, WCNCW, 15–18 April 2019, Marrakech, Morocco*. IEEE; 2019. DOI:10.1109/WCNCW.2019.8902590

Статья поступила в редакцию 08.06.2022; одобрена после рецензирования 10.06.2022; принята к публикации 15.06.2022.

The article was submitted 08.06.2022; approved after reviewing 10.06.2022; accepted for publication 15.06.2022.

Информация об авторах:

ПЛОТНИКОВ
Павел Владимирович

кандидат физико-математических наук, доцент кафедры высшей математики Санкт-Петербургского государственного университета телекоммуникаций им. проф. М.А. Бонч-Бруевича

✉ <https://orcid.org/0000-0001-8869-6142>

ВЛАДЫКО
Андрей Геннадьевич

кандидат технических наук, доцент, декан факультета фундаментальной подготовки Санкт-Петербургского государственного университета телекоммуникаций им. проф. М.А. Бонч-Бруевича

✉ <https://orcid.org/0000-0002-8852-5607>