Метод выгрузки трафика в V2X/5G сетях на основе системы граничных вычислений

А.Г. Владыко, доцент кафедры сетей связи и передачи данных СПбГУТ, к.т.н.; vladyko@sut.ru
А.С. Мутханна, доцент кафедры сетей связи и передачи данных СПбГУТ, к.т.н.; ammarexpress@gmail.com
А.Е. Кучерявый, заведующий кафедрой сетей связи и передачи данных СПбГУТ, профессор, д.т.н.; akouch@mail.ru

УДК 621.391 DOI: ???

Аннотация. *Разработана структура системы V2X (Vehicle-to-Everything), позволяющей на базе сети 5G организовать взаимодействие между объектами дорожной инфраструктуры, беспилотными и высокоавтоматизированными транспортными средствами, пешеходами и другими сетями. Для производительной работы в условиях плотного трафика транспортных средств в системе V2X использованы технология мобильных граничных вычислений MEC (Mobile Edge Computing) и алгоритм выгрузки трафика, позволяющий управлять нагрузкой данных на граничные и облачные серверы. Результаты моделирования для различных сценариев показали, что предложенные структура и алгоритм выгрузки могут значительно снизить задержки в передаче данных и энергопотребление системы.*

Ключевые слова: *V2X, граничные вычисления, MEC, 5G, выгрузка трафика.*

ВВЕДЕНИЕ

К концу 2020 г. будут готовы стандарты для построения сотовой системы пятого поколения с заявленными требованиями и начнется последний этап в ее развитии, а именно внедрение системы 5G [1, 2]. Каждый вариант использования приложений 5G представляет собой большую проблему и требует реализации определенных методов планирования сети [3]. Большинство приложений могут быть достаточно просто реализованы в 5G, однако, это не касается интеллектуальных транспортных систем и сетей автомобильного транспорта VANET (Vehicular Ad Hoc Networks).

Интеллектуальные транспортные системы (ИТС) [4–6] являются одной из наиболее быстро развивающихся новых технологий. ИТС внедряются повсеместно, поскольку они имеют огромный потенциал для повышения безопасности и эффективности дорожного движения, а также комфорта населения. ИТС поддерживают несколько интерфейсов связи между устройствами сетевой инфраструктуры: взаимодействие между транспортными средствами V2V (Vehicle-to-Vehicle); взаимодействие между транспортными средствами и сетевой инфраструктурой V2I (Vehicle-to-Infrastructure); взаимодействие между транспортными средствами и облачными системами V2C (Vehicle-to-Cloud); и взаимодействие между транспортными средствами и пешеходами (включая велосипедистов) V2P (Vehicle-to-Pedestrian) [7, 8]. Такой набор взаимодействий позволяет давать сложный, но быстрый ответ на изменения состояния сети и местоположения всех ее участников, что в конечном итоге повышает актуальность ИТС для современных городов [9].

Развитие современных транспортных систем основывается не только на повышении безопасности дорожного движения и расширении сети и транспортной инфраструктуры, но и на создании компетентного управленческого комплекса. На первый план выходят такие аспекты функционирования ИТС, как безопасность передаваемой информации, быстрое увеличение числа пользователей сети, постоянный рост сетевого трафика и многие другие факторы [10]. Для решения этих задач и устранения ряда других известных узких мест ИТС необходимо разработать эффективную архитектуру, которая будет справляться с нагрузкой, создаваемой элементами сети, правильно и своевременно управлять потоками трафика, проводить непрерывный статистический анализ и при этом соответствовать стандартам и требованиям современных сетей связи [11].

Важнейшую роль в создании ИТС играют сети VANET [12, 13]. Современная сеть VANET предоставляет большое количество новых сервисов и протоколов, поэтому переход к интегральному взаимодействию с окружающим информационным пространством, определяемый как взаимодействие V2X, должен обеспечить совместимость разнообразных сетей и систем, участвующих в предоставлении услуг пользователям VANET.

Для обеспечения эффективного функционирования сетей VANET в статье предложена структура сети VANET с использованием мобильных пограничных вычислений MEC и программно-определяемых сетей SDN (Software-Defined Network), а также алгоритм выгрузки трафика. Эти решения обеспечивают уменьшение задержки и снижают энергопотребление системы.

СТРУКТУРА СИСТЕМЫ

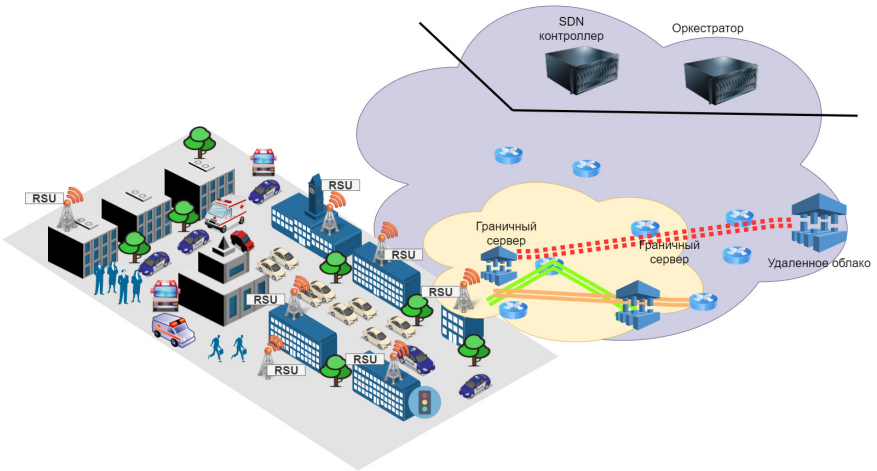
Система базируется на двух основных технологиях: SDN в ядре сети и MEC в беспроводной сети радиодоступа.

Использование технологии SDN в VANET обеспечивает более высокий уровень безопасности сети, повышение производительности, снижение задержки, уменьшение перегрузки сети и повышение эффективности маршрутизации. Технология MEC позволяет решать необходимые для управления трафиком вычислительные задачи в непосредственной близости от транспортных средств, что уменьшает задержку передачи данных и решает проблему высокой плотности трафика [14]. Благодаря внедрению технологий SDN и MEC можно избежать прохождения исходящего трафика транспортных средств через ядро сети, чтобы управлять этим трафиком и обрабатывать его на границе сети доступа.

Система VANET предусматривает установку распределенных придорожных устройств RSU (Roadside Unit) для оказания сетевой поддержки транспортным средствам. При этом ядро сети имеет интерфейс для сервера приложений, который в свою очередь предоставляет дополнительные услуги для транспортных средств. Транспортные средства взаимодействуют друг с другом на основе интерфейса V2V, а со всем остальным окружением – с использованием технологии V2X через соответствующий надежный интерфейс радиодоступа, например, DSRC (dedicated short-range communications) или сотовую связь C-V2X (Cellular V2X). Протокол DSRC основан на стандартах IEEE 802.11, которые были модифицированы для автомобильной связи. Стандарт IEEE 802.11p является наиболее распространенной технологией радиодоступа, разработанной для мобильных сетей MANET (Mobile Ad hoc Network), включая сети для транспортных средств. Физический уровень и уровень MAC IEEE 802.11p основаны на стандарте IEEE 802.11a.

Технические требования стандарта IEEE 802.11p предусматривают устойчивое функционирование при движении автомобиля со скоростью до 200 км/ч, при этом время отклика должно составлять около 100 мс, а дальность связи – до 1 км. В автомобильных сетях высокой плотности из-за коллизии пакетов производительность IEEE 802.11p оказывается низкой, что делает более желательным для таких приложений использование технологии C-V2X. Чтобы преодолеть указанные ограничения стандарта IEEE 802.11.p и повысить производительность DSRC по сравнению с C-V2X был разработан новый протокол радиодоступа – IEEE 802.11bd. Протокол IEEE 802.11bd предлагает более высокопроизводительные режимы работы с пропускной способностью от 1 Гбит/с и выше.

Рисунок 1
Структура предлагаемой системы VANET



Предлагаемая система имеет многоуровневую структуру MEC и интеллектуальное ядро сети в общей структуре системы VANET, как это показано на рис. 1. Система включает три вида взаимодействий. Первое – это связь между транспортными средствами V2V. Второе взаимодействие – V2I, т.е. данные передаются между транспортными средствами и стационарной дорожной инфраструктурой (RSU). Третье взаимодействие V2X осуществляется между транспортными средствами и любыми другими окружающими устройствами.

МОДЕЛЬ СИСТЕМЫ

Рассмотрим предлагаемую модель системы. Как показано на рис. 1, модель включает граничную сеть транспортных средств Vehicular Edge Network (VEN) с односторонней дорогой, набор из N движущихся вдоль дороги транспортных средств, каждое из которых имеет M независимых вычислительных задач, которые должны быть решены. Транспортные средства соединены с едиными придорожными блоками RSU через беспроводный канал, который подключен к мобильному граничному вычислительному серверу, а также к централизованному облачному серверу через ядро сети. Обозначим множество транспортных средств и их вычислительные задачи как $\mathbf{N} = \{1, 2, ..., N\}$ и $\mathbf{M} = \{1, 2, ..., M\}$. Вычислительные задачи могут быть выполнены локально на самом транспортном средстве или выгружены и обработаны удаленно на пограничном или облачном сервере.

МОДЕЛЬ СЕТИ

В модели сети рассматривается среда с беспроводным RSU, имеющая граничный вычислительный сервер и соединенная через ядро сети с одним облачным сервером. Кроме того, существует множество $\mathbf{N} = \{1, 2, ..., N\}$

транспортных средств, в котором каждое транспортное средство имеет множество $\mathbf{M}=\{1, 2, ..., M\}$ независимых вычислительных задач. Эти задачи либо выполняются локально, либо выгружаются и выполняются граничным или облачным сервером.

Обозначим $\alpha_{i,j,k}$ двоичное вычислительное разгрузочное решение для j-ой вычислительной задачи i-го транспортного средства, которое определяет место выполнения вычислительной задачи. Более конкретно: $\alpha_{i,j,0}=1$ означает, что j-я вычислительная задача i-го транспортного средства будет выполняться локально ресурсом транспортного средства (k=0); $\alpha_{i,j,1}=1$ и $\alpha_{i,j,2}=1$ означает, что j-я вычислительная задача i-го транспортного средства будет выгружена и обработана удаленно на граничном (k=1) или облачном (k=2) серверах соответственно. В целом, каждая j-я вычислительная задача должна выполняться только один раз, независимо от того, происходит ли это локально (k=0) или удаленно ($k\in\{1,2\}$).

Максимальная скорость передачи по каналу связи восходящего и нисходящего трафика для каждого транспортного средства может быть определена следующим образом [15]:

$$r_i^U = B^U \log_2(1+\frac{p_i^T G_0^2}{\text{É}_0 B^U});$$

$$r_i^D = B^D \log_2(1+\frac{p_{bs}^T G_0^2}{\text{É}_0 B^D}),$$

где B_U и B_D –полосы соответственно нисходящего и восходящего каналов; p_i^T и p_{bs}^T –мощности передачи i-го устройства и пограничного сервера; ω_0 и G_0 – спектральная плотность мощности шума и сопутствующее усиление канала между транспортным средством и граничным сервером.

РАСЧЕТНАЯ МОДЕЛЬ

В вычислительной модели предлагаемая система имеет одно мобильное устройство, связанное с одним RSU и имеющее M вычислительных задач, которые должны быть выполнены локально на мобильном устройстве или удаленно на граничном или облачном сервере. Кроме того, каждая i-я вычислительная задача может быть охарактеризована и представлена с помощью кортежа $\{b_i, c_i\}$ в котором b_i задает размер входных данных для i-ой вычислительной задачи, которые будут передаваться с мобильного устройства на подключенный RSU для выполнения (программный код и входные параметры), а c_i определяет количество циклов процессора, необходимых для выполнения i-ой вычислительной задачи. Значения b_i и c_i могут быть получены итеративным методом.

Как уже упоминалось выше, моделируется один граничный сервер, связанный с N наборами транспортных средств, в которых каждое транспортное средство имеет M независимых вычислительных задач, которые необходимо выполнить. Для каждой j-ой вычислительной задачи будем использовать кортеж $\{a_{i,j}, b_{i,j}, c_{i,j}, \tau_{i,j}\}$, чтобы представлять требования вычислительной задачи, где $a_{i,j}$ и $b_{i,j}$ – размер входных и выходных данных; $c_{i,j}$ – итоговое количество циклов работы процессора; $\tau_{i,j}$ – допустимый срок выполнения j-ой задачи i-го автомобиля. Значения $a_{i,j}$, $b_{i,j}$ и $c_{i,j}$ можно получить итеративным способом, как представлено в [15–17].

ЛОКАЛЬНОЕ ВЫПОЛНЕНИЕ ЗАДАЧИ

При локальном выполнении, когда вся j-я вычислительная задача выполняется локально на самом транспортном средстве, общее время выполнения и потребление энергии определяются соответственно как

$$T_{i,j}^L = \frac{c_{i,j}}{f_i^L};$$

$$E_{i,j}^L = \zeta_i c_{i,j},$$

где f_i^L – тактовая частота процессора (количество циклов работы процессора в секунду) бортового устройства (onboard units, OBU) автомобиля; а ζ_i – коэффициент, который обозначает энергию, потребляемую за один цикл работы процессора, и зависит от архитектуры CPU. Принимаем $\zeta_i = 10^{-11}(f_i^L)^2$ [18–20].

УДАЛЕННОЕ ВЫПОЛНЕНИЕ ЗАДАЧИ

При удаленном выполнении, когда j-я вычислительная задача i-го транспортного средства выгружена и обработана на граничном или облачном сервере, общее время выполнения можно определить соответственно как

$$T_{i,j}^E = T_{i,j}^{\text{off}} + T_{i,j}^{E_\text{exec}} + T_{i,j}^{\text{down}};$$

$$T_{i,j}^C = T_{i,j}^{\text{off}} + \xi + T_{i,j}^{C_\text{exec}} + T_{i,j}^{\text{down}},$$

где ξ – задержка распространения для передачи вычислительной задачи между граничным и облачным серверами; $T_{i,j}^{\text{off}}$, $T_{i,j}^{\text{down}}$, $T_{i,j}^{E_\text{exec}}$ и $T_{i,j}^{C_\text{exec}}$ – времена соответственно выгрузки, загрузки и выполнения для обработки j-й задачи вычисления i-го автомобиля на граничном и облачном серверах, которые можно выразить следующим образом:

$$T_{i,j}^{\text{off}} = \frac{a_{i,j}}{r_i^U};$$

$$T_{i,j}^{\text{down}} = \frac{b_{i,j}}{r_i^D};$$

$$T_{i,j}^{E_\text{exec}} = \frac{c_{i,j}}{f_i^E};$$

$$T_{i,j}^{C_\text{exec}} = \frac{c_{i,j}}{f_i^C},$$

где f_i^E и f_i^C – количество циклов работы процессоров в

секунду для граничного и облачного серверов для i-го транспортного средства.

Потребление энергии для выгрузки и обработки j-й вычислительной задачи i-го транспортного средства удаленно на граничном и облачном серверах может быть определено следующим образом:

$$E_{i,j}^{E} = p_i^T T_{i,j}^{\text{off}} + {}^2\, T_{i,j}^{E_\text{exec}} + p_i^R T_{i,j}^{\text{down}};$$

$$E_{i,j}^{C} = p_i^T T_{i,j}^{\text{off}} + {}^2\, T_{i,j}^{C_\text{exec}} + p_i^R T_{i,j}^{\text{down}},$$

где p_i^R – мощность приема OBU транспортного средства; β – постоянная величина потребления энергии в режиме ожидания OBU при обработке задачи на граничном сервере и в облаке.

Общие расходы ресурсов на выполнение вычислений j-й задачи i-го транспортного средства в терминах задержки и потребления энергии могут быть определены как

$$Z_{i,j} = w_{i,j}^{t}(T_{i,j}) + w_{i,j}^{e}(E_{i,j}),$$

где w_i^e и $w_i^t \in [0, 1]$ обозначают веса соответственно задержки и энергопотребления для принятия решения i-ым транспортным средством. При этом $T_{i,j}$ и $E_{i,j}$ – это общее время и энергия, которые могут быть выражены как

$$T_{i,j} = \left[\alpha_{i,j,0} T_{i.j}^{L} + \alpha_{i,j,1} T_{i,j}^{E} + \alpha_{i,j,2} T_{i,j}^{C}\right];$$

$$E_{i,j} = \left[\alpha_{i,j,0} E_{i,j}^{L} + \alpha_{i,j,1} E_{i,j}^{E} + \alpha_{i,j,2} E_{i,j}^{C}\right].$$

ЗАДАЧА ОПТИМИЗАЦИИ

Сформулируем задачу оптимизации следующим образом:

$$\min_{x,y} \quad Z_{i,j} \qquad (1)$$

ограничения

$$\left[E_{i,j} - E_{i,j}^{L}\right] \le 0, \quad \forall_{i,j} \quad C1;$$

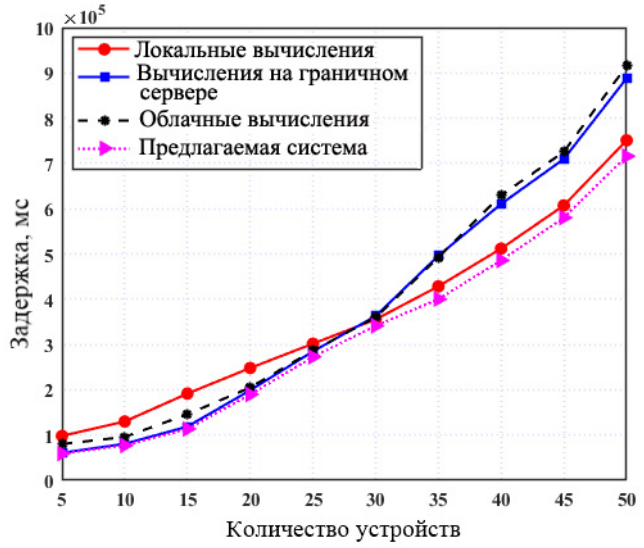
$$T_{i,j} \le \tau_{i,j}, \quad \forall_{i,j} \quad C2;$$

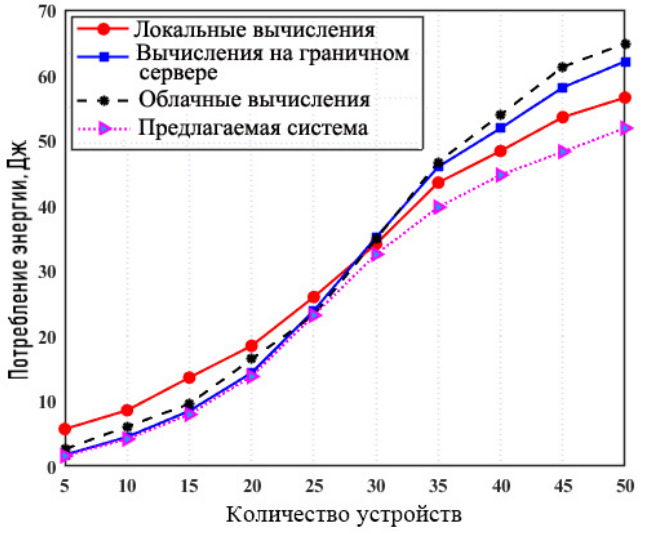
$$\sum_{k=0}^{2} \alpha_{i,j,k} = 1, \quad \forall_{i,j} \quad C3;$$

$$\alpha_{i,j,k} \in \{0,1\}, \quad \forall_{i,j} \quad C4.$$

Целевой функцией задачи оптимизации является минимизация суммарных затрат ресурсов всей системы в целом по задержке и энергопотреблению при выгрузке трафика (вычислительной задачи). Ограничения $C1$ и $C2$ являются верхними границами потребления энергии и задержки соответственно. Ограничение $C3$ гарантирует, что каждая j-я вычислительная задача должна выполняться только один раз. Наконец, последнее ограничение $C4$ требует, чтобы переменная решения выгрузки

Рисунок 2
Сравнение задержек (а) и потребления энергии (б) при различном числе транспортных средств





вычислений являлась двоичной. Поскольку целевая функция линейна, а все ограничения также линейны, задача оптимизации (1) является целочисленной линейной оптимизационной задачей, в которой оптимальное решение может быть получено с помощью метода ветвей и границ [21].

АЛГОРИТМ ВЫГРУЗКИ ВЫЧИСЛЕНИЙ ДЛЯ МНОЖЕСТВА ТРАНСПОРТНЫХ СРЕДСТВ

Рассмотрим алгоритм выгрузки многопользовательских вычислений для получения оптимального решения выгрузки задач в соответствии с (1). Все транспортные средства инициализируют свое решение о выгрузке, но при $\alpha_{i,j,0} = 1$ используется локальное решение задачи. Каждое устройство загружает на граничный сервер

Рисунок 3
Сравнение задержки (а) и энергопотребления (б) при различных вычислительных возможностях граничного сервера

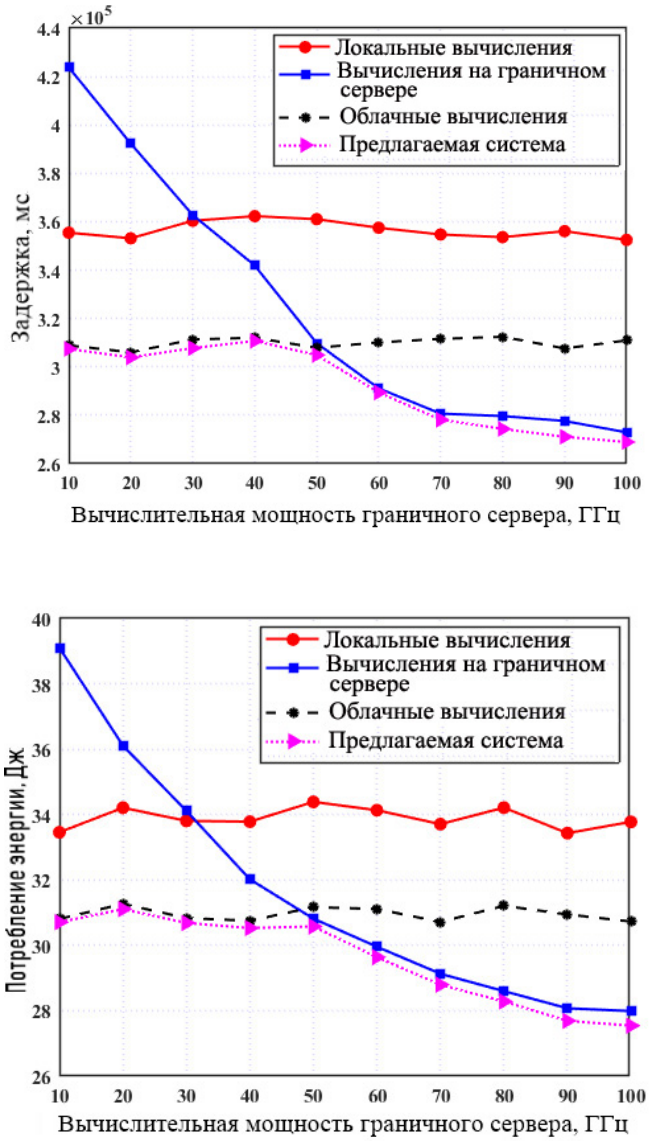
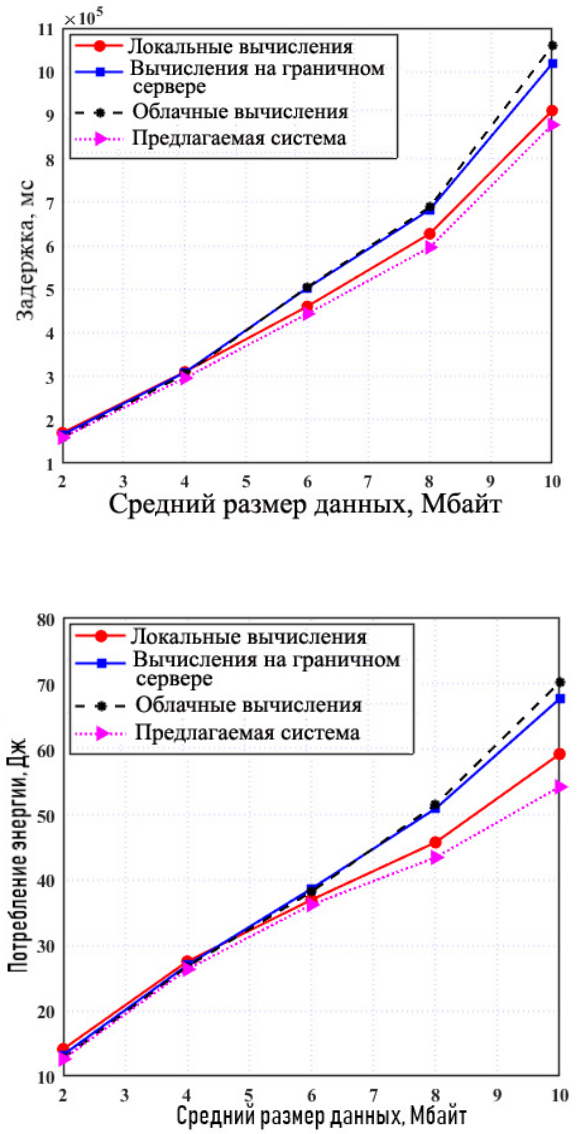


Рисунок 4
Сравнение задержки (а) и энергопотребления (б) при различных значениях объема данных



требования вычисления задач, которые включают в себя кортеж $\{a_{i,j}, b_{i,j}, c_{i,j}, \tau_{i,j}, p_i^T, p_i^R, \zeta_i\}$ и локальные возможности вычисления по тактовой частоте f_i^L. После этого граничный сервер определяет восходящую линию связи и скорость передачи данных по нисходящей линии связи для каждого транспортного средства, основываясь на текущем числе транспортных средств. Граничный сервер находит оптимальное место выполнения для каждой вычислительной задачи (например, локальное, граничное или облачное) путем решения указанной выше задачи оптимизации. После этого каждое транспортное средство получает информацию о месте выполнения своих вычислительных задач от граничного сервера, и тем самым минимизируются суммарные затраты ресурсов всей системы с точки зрения времени и энергии.

РЕЗУЛЬТАТЫ МОДЕЛИРОВАНИЯ

При моделировании использовался симулятор на основе языка программирования Python, компьютер был оснащен процессором Intel Core i7-4770 с тактовой частотой 3,4 ГГц и ОЗУ объемом 8 Гбайт, работал он под управлением 64-разрядной ОС Windows 10 Professional. Рассматривалась многозадачная облачная вычислительная система с одним облачным сервером, одним RSU, двадцатью транспортными средствами (N=20), при этом каждое устройство имело M=5 независимых вычислительных задач, которые могли выполняться локально или выгружаться и обрабатываться удаленно

на доступном граничном или облачном сервере. Каждая вычислительная задача имела входные данные, размер которых был равномерно распределен в пределах диапазона от 5 до 15 Мбайт при этом предполагалось, что размер выходных данных составляет 20% от размера входных данных. Вычислительная способность центрального процессора каждого OBU транспортного средства задана равномерно распределенной в пределах диапазона его тактовой частоты {0,5; 0,6; ...; 1,0} ГГц. При этом тактовые частоты граничного и облачного серверов составляют 20 и 50 ГГц соответственно.

На рис. 2, *а* и *б* показано потребление энергии и задержка при выполнении вычислительных задач для четырех разных сценариев в зависимости от числа транспортных средств. При этом рассматриваются следующие сценарии: локальные вычисления, вычисления на граничном сервере, вычисления на облачном сервере, вычисления в предлагаемой системе. Из рис. 2, *а* и *б* видно, что предложенный метод выгрузки достигает наилучшего результата по сравнению с другими сценариями. Кроме того, сценарии граничного и облачного исполнения превосходят локальное исполнение по мере увеличения числа транспортных средств (когда число транспортных средств более чем 25).

Задержка и энергопотребление для выполнения вычислительных задач транспортных средств в зависимости от различных значений возможностей граничного сервера при четырех различных сценариях показаны на рис. 3, *а* и *б*, соответственно. Как видим, предложенный метод выгрузки позволяет существенно уменьшить задержку и энергопотребление в области заданных значений вычислительных возможностей граничного сервера.

Наконец, на рис. 4, *а* и *б* представлены зависимости задержки и энергопотребления при выполнении вычислительных задач для разных значений объема данных. Из рис. 4 видно, что при граничных и облачных сценариях вычислений значения задержек и энергопотребления превышают аналогичные при локальном выполнении вычислительных задач, когда размер данных превышает 6 Мбайт. Как видно из рис. 4, предложенный метод позволяет добиться наилучших результатов. Когда размер передаваемых данных увеличивается, потребление энергии и задержка увеличиваются для всех сценариев естественным образом по мере увеличения размера данных вычислительных задач.

ВЫВОДЫ

1. В статье предложена модель VEN-системы для сетей V2X/5G, функционирующая на основе технологий MEC и SDN, а также алгоритм выгрузки, учитывающий минимизацию суммарных затрат ресурсов системы по задержке и энергопотреблению при выгрузке вычислительной задач транспортных средств.
2. Результаты моделирования показали, что рассмотренный метод выгрузки трафика позволяет уменьшить как задержку в сети, так и потребление энергии по сравнению с известными методами.
3. Метод выгрузки трафика в сетях V2X/5G на основе граничных вычислений может быть рекомендован в качестве основного для сетей автомобильного транспорта VANET.

Исследование выполнено в рамках исполнения Государственного контракта № П33-1-26/9.

СПИСОК ЛИТЕРАТУРЫ

1. Атея, А.А. Интеллектуальное ядро для сетей связи 5G и тактильного интернета на базе программно-конфигурируемых сетей / А.А. Атея, А.С. Мутханна, А.Е. Кучерявый // Электросвязь. – 2019. – № 3. – С. 34-40.

2. Akyildiz, I.F. 5G roadmap: 10 key enabling technologies / I.F. Akyildiz, S. Nie, S.C. Lin, M. Chandrasekaran // Computer Networks. – 2016. – Vol. 106. – P. 17-48.

3. The Tactile Internet. ITU-T Technology Watch Report. – ITU, 2014. Режим доступа: https://www.itu.int/dms_pub/itu-t/opb/gen/T-GEN-TWATCH-2014-1-PDF-E.pdf (дата обращения 12.07.2020).

4. Васильев, В.Н. Выбор структуры сети передачи данных для общегородской интеллектуальной транспортной системы / В.Н. Васильев, В.А. Григорьев, И.А. Хворов и др. // Электросвязь. – 2015. – № 6. – С. 35-41.

5. Васильев, В.Н. Системы связи на транспорте: тенденции развития и задачи регулирования / В.Н. Васильев, В.А. Григорьев, И.А. Хворов, Ю.А. Распаев // Электросвязь. – 2016. – № 2. – С. 18-23.

6. Baras, S. VANETs-based intelligent transportation systems: an overview / S. Baras, I. Saeed, H.A. Tabaza, M. Elhadef // Lecture Notes in Electrical Engineering. – 2018. – Vol. 474. – P. 265-273.

7. Agarwal, P. Technical review on different applications, challenges and security in VANET / P. Agarwal // Journal of Multimedia Technology & Recent Advancements. – 2017. – Vol. 4, Issue 3. – P. 21-30.

8. Jameel, F. Impact of co-channel interference on the Performance of VANETs under α-μ Fading / F. Jameel, Z. Hamid, F. Jabeen, M.A. Javed // AEU - International Journal of Electronics and Communications. – 2018. – Vol. 83. – P. 263-269.

9. Aliyu, A. Cloud Computing in VANETs: Architecture, Taxonomy, and Challenges / A. Aliyu, A.H. Abdullah, O. Kaiwartya et al. // IETE Technical Review. – 2018. – Vol. 35. – P. 523-547.

10. Cooper, C. Comparative Survey of VANET Clustering Techniques / C. Cooper, D. Franklin, M. Ros et al. // IEEE Communications Surveys & Tutorials. – 2017. – Vol. 19, Issue 1. – P. 657-681.

11. Hussain, R. Realization of VANET-Based Cloud Services through Named Data Networking / R. Hussain, S.H. Bouk, N. Javaid et al. // IEEE Communications Magazine. – 2018. – Vol. 56, Issue 8. – P. 168-175.

СПИСОК ЛИТЕРАТУРЫ

12. Кучерявый, А.Е. Сети связи общего пользования. Тенденции развития и методы расчета / А.Е. Кучерявый, А.И. Парамонов, Е.А. Кучерявый. – М.: ФГУП ЦНИИС, 2008. – 296 с.

13. Кучерявый, А.Е. Самоорганизующиеся сети / А.Е. Кучерявый, А.В. Прокопьев, Е.А. Кучерявый. – СПб.: Любавич, 2011. – 312 с.

14. Vladyko, A. Distributed Edge Computing to Assist Ultra-Low-Latency VANET Applications / A. Vladyko, A. Khakimov, A. Muthanna et al. // Future Internet. – 2019. – Vol. 11, Issue 6. – P. 128.

15. Elgendy, I.A. Resource allocation and computation offloading with data security for mobile edge computing / I.A. Elgendy, W. Zhang, Y.C. Tian, K. Li // Future Generation Computer Systems. – 2019. – Vol. 100. – P. 531-541.

16. Lyu, X. Adaptive receding horizon offloading strategy under dynamic environment / X. Lyu, H. Tian // IEEE Communications Letters. – 2016. – Vol. 20. – P. 878-881.

17. Elgendy, I.A. An efficient framework to improve the performance ofmobile applications / I.A. Elgendy, M. El-kawkagy, A. Keshk // International Journal of Digital Content Technology and its Applications (JDCTA). – 2015. – Vol. 9. – P. 43-54.

18. Chen, X. Decentralized Computation Offloading Game for Mobile Cloud Computing / X. Chen // IEEE Transactions on Parallel and Distributed Systems. – 2015. – Vol. 26. – P. 974-983.

19. Wen, Y. Energy-optimal mobile application execution: Taming resource-poor mobile devices with cloud clones / Y. Wen, W. Zhang, H. Luo // Proceedings IEEE INFOCOM. – 2012. – P. 2716-2720.

20. Miettinen, A.P. Energy efficiency of mobile clients in cloud computing / A.P. Miettinen, J.K. Nurminen // Proceedings of the 2nd USENIX conference on Hot topics in cloud computing (HotCloud'10). – 2010. – P. 4.

21. Stidsen, T.R. A branch and bound algorithm for a class of biobjective mixed integer programs / T.R. Stidsen, K.A. Andersen, B. Dammann // Management Science. – 2014. – Vol. 60, Issue 4. – P. 1009-1032.

Получено 12.07.20