Научная статья
УДК 004.27
DOI:10.31854/1813-324X-2023-9-3-42-59

Модель интеграции граничных вычислений в структуру сети «воздух–земля» и метод выгрузки трафика для сетей Интернета Вещей высокой и сверхвысокой плотности

Аммар Салех Али Мутханна, muthanna.asa@sut.ru

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

Аннотация: *Статья посвящена научной проблеме интеграции граничных вычислений в структуру сети «воздух–земля» для сетей Интернета Вещей высокой и сверхвысокой плотности. Подобные проблемы являются наиболее актуальными сегодня в связи с появлением концепции интегрированных сетей «космос–воздух–земля–море». Разработана модель сети, в которой предложено для уменьшения задержки и энергопотребления использовать мобильные серверы граничных вычислений, расположенные на беспилотных летательных аппаратах (БПЛА), а также метод выгрузки трафика с наземной сети на мобильные серверы граничных вычислений на БПЛА. При этом процедура выгрузки трафика является трехуровневой, а на оконечных устройствах используется программный профилировщик, который определяет сложность вычисляемой задачи и по результатам его работы механизм принятия решения делает вывод о необходимости выгрузки трафика. Для оптимизации структуры сети «воздух–земля» для сетей Интернета Вещей высокой и сверхвысокой плотности с целью минимизации задержки и энергопотребления при выгрузке трафика с наземной сети на серверы граничных вычислений БПЛА разработан метаэвристический алгоритм на основе хаотического «роя сальп». Результаты моделирования показали, что предложенные модель и метод обеспечивают существенное уменьшение задержки и энергопотребления, а также доли заблокированных задач при выгрузке трафика по сравнению с известными решениями.*

Ключевые слова: *интегрированные сети, сеть «воздух–земля», сети высокой и сверхвысокой плотности, задержка, энергопотребление, алгоритм «роя сальп»*

Ссылка для цитирования: Мутханна А.С.А. Модель интеграции граничных вычислений в структуру сети «воздух–земля» и метод выгрузки трафика для сетей Интернета Вещей высокой и сверхвысокой плотности // Труды учебных заведений связи. 2023. Т. 9. № 3. С. 42–59. DOI:10.31854/1813-324X-2023-9-3-42-59

A Model for Integrating Edge Computing into an Air-Ground Network Structure and Offloading Traffic Method for High and Ultra-High Densities Internet of Things Networks

Ammar Muthanna, muthanna.asa@sut.ru

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

Abstract: *The scientific challenge of incorporating edge computing into the air-ground network architecture for high and ultra-high density Internet of Things networks is the focus of this article. These issues are particularly important right now because of the concept of "space–air–ground–sea" inegrated networks. A mechanism for offloading traffic from the ground network to mobile edge computing servers on UAVs has also been devised. This network model suggests using mobile edge computing servers deployed on unmanned aerial vehicles (UAVs) to reduce latency and*

power consumption. At the same time, a software profiler is utilized on the terminal devices to identify the difficulty of the computed task and, based on that determination, a three-level technique for offloading traffic is used.

Keywords: *integrated networks, air-ground network, high and ultra-high density networks, latency, power consumption, salp swarm algorithm*

For citation: Muthanna A. A Model for Integrating Edge Computing into an Air-Ground Network Structure and Offloading Traffic Method for High and Ultra-High Densities Internet of Things Networks. *Proc. of Telecommun. Univ.* 2023;9(3):42–59. (in Russ.) DOI:10.31854/1813-324X-2023-9-3-42-59

Введение

Развитие сетей связи в настоящее время осуществляется в направлении создания интегрированных сетей связи «космос–воздух–земля–море» (SAGSIN, *аббр. от англ.* Space–Air–Ground–Sea Integrated Networks) [1]. Это ни в коей мере не отрицает достижений в области сетей связи пятого и шестого поколений, сетей связи высокой и сверхвысокой плотности, сетей связи с ультрамалыми задержками [2, 3]. Напротив, интеграция ресурсов в сети SAGSIN должна позволить эффективно использовать возможности таких сетей.

Сценарии высокоплотного и сверхплотного построения сетей связи [4, 5] призваны в полной мере использовать возможности концепции Интернета Вещей, а сети с ультрамалой задержкой – концепции Тактильного Интернета. Для решения задач по построению таких сетей требуется использовать новые технологии в области сетей и систем связи [6]. К таким технологиям относятся, например, распределенные граничные вычисления (MEC, *аббр. от англ.* Mobile Edge Computing) [7–10] и беспилотные летательные аппараты (БПЛА) [11–15].

Беспилотные летательные аппараты могут быть использованы для сетевой поддержки в современных сетях для решения разнообразных задач, таких как увеличение покрытия сети, при чрезвычайных ситуациях, граничные вычисления на базе БПЛА и т. д. [16–21].

В статье исследуется возможность использования БПЛА в качестве мобильных серверов граничных вычислений для поддержки наземных высокоплотных и сверхплотных сетей Интернета Вещей с целью уменьшения задержки и энергопотребления, а также уменьшения доли заблокированных задач при выгрузке трафика в воздушный сегмент сети (БПЛА). Эта цель достигается путем разработки модели сети, метода выгрузки трафика и оптимизации полученных решений с использованием метаэвристического алгоритма на базе хаотического SSA (*аббр. от англ.* Salp Swarm Algorithm – алгоритм «роя сальп»).

Модель сети. Разработанная модель сети представлена на рисунке 1. Она включает в себя наземный и воздушный сегменты.

A) Наземный сегмент

Наземный сегмент представляет собой сеть Интернета Вещей с высокой плотностью или сверхвысокой плотностью с распределенными оконечными устройствами. Это четырехуровневая сеть, для обеспечения функционирования которой используется технология MEC.

Первый уровень включает в себя распределенные устройства IoT, например, датчики и исполнительные механизмы. Эти устройства используют измерения параметров окружающей среды.

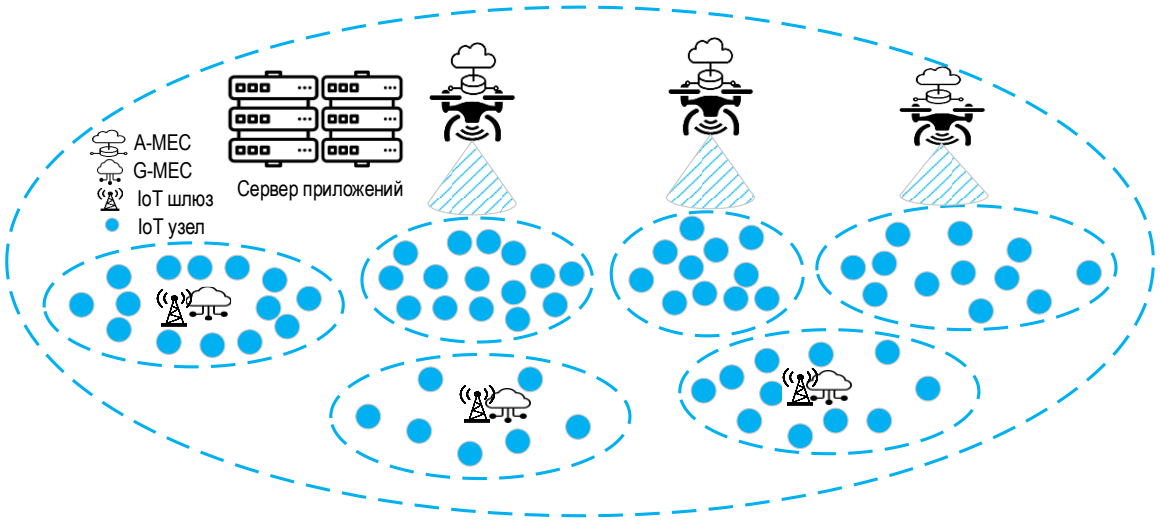

Рис. 1. Разработанная модель сети

Fig. 1. Developed Network Model

Сценарий высокоплотного и сверхплотного развертывания предполагает, что количество таких устройств будет очень велико; они могут быть распределены по широкому диапазону областей, поддерживают различные интерфейсы, в том числе интерфейс «воздух–земля» (A2G, *аббр. от англ.* Air-to-Ground) с БПЛА.

Наземные распределенные серверы граничных вычислений (G-MEC, *аббр. от англ.* Ground MEC) представляют собой *второй уровень* такой сети.

Третий уровень образуется шлюзами сети Интернета Вещей. Этот уровень представляет собой интерфейс между оконечными устройствами и облачным сервером сети Интернета Вещей [22]. При этом предполагается, что все шлюзы IoT в рассматриваемой сети подключены и к воздушной плоскости.

Четвертый уровень – это прикладной уровень, а именно: облачный сервер приложений. Все собранные данные передаются через IoT-шлюзы или воздушный сегмент на сервер приложений, где данные анализируются и хранятся. Принятые данные предварительно обрабатываются на оконечных устройствах и граничных, что обеспечивается на основе разработанного метода выгрузки трафика.

Б) Воздушный сегмент

Воздушный сегмент включает в себя множество БПЛА, например, микродронов и квадрокоптеров, развернутых для поддержки сетей Интернета Вещей высокой и сверхвысокой плотности. Каждый БПЛА имеет два интерфейса связи: A2G и A2A (*аббр. от англ.* Air-to-Air – «воздух–воздух»). A2G – это интерфейс, используемый для связи с наземным сегментом, а интерфейс A2A – для связи между БПЛА.

Каждый БПЛА имеет граничный сервер A-MEC (*аббр. от англ.* Air MEC) для предоставления вычислительных ресурсов наземной сети. A-MEC представляет собой микрооблачный сервер, обеспечивающий процедуру выгрузки трафика.

Каждый БПЛА обслуживает кластер оконечных устройств Интернета Вещей. При этом устройства IoT решают, присоединиться ли к кластеру узла БПЛА или связаться с IoT-шлюзом на основе индикатора уровня принимаемого сигнала (RSSI, *аббр. англ.* Received Signal Strength Indicator). Оконечное устройство IoT решает также, к какому БПЛА присоединиться или присоединиться к IoT-шлюзу, сравнивая их RSSI.

Математическая модель исследуемой сети и ее составляющих

Прежде всего, определим необходимые для исследования параметры и переменные. В таблице 1 приведены обозначения всех рассматриваемых параметров и переменных.

ТАБЛИЦА 1. Параметры и переменные

TABLE 1. Parameters and Variables

Обозначение	Описание
D	Набор БПЛА, развернутый в воздушной плоскости
$R_{MEC\text{-}A}$	Вычислительные ресурсы сервера МЭК-А
N	Общее число развернутых БПЛА в воздухе
h	Высота БПЛА
x, y	Характеристики местоположения БПЛА (двумерные наземные координаты)
x_{Di}, y_{Di}	Характеристики местоположения i-го БПЛА, D_i (двумерные наземные координаты)
h_{Di}	Высота i-го БПЛА (D_i)
P^{Di}	Текущее местоположение i-го БПЛА, Дн
T^{Di}	Траектория БПЛА D_i
G_x^{Di}	Множество x местоположений БПЛА D_i, формирующих траекторию T^{Di}
G_y^{Di}	Множество y местоположений БПЛА D_i, формирующих траекторию T^{Di}
I	Набор оконечных устройств IoT, развернутых в наземном сегменте
M	Общее количество устройств IoT, развернутых в наземном сегменте
P^{Ij}	Текущее местоположение IoT-устройства I_j
x_{Ij}, y_{Ij}	Характеристики местоположения IoT-устройства I_j (местоположение xy)
C	Набор сформированных кластеров
CM_j^{Ci}	j-й член кластера i-го кластера (C_i)
K_i	Общее количество членов кластера i-го кластер (C_i)
G	Набор IoT-шлюзов, развернутых в наземном сегменте
W	Общее количество IoT-шлюзов, развернутых в наземном сегменте

Обозначение	Описание
$L_{Ii,Dj}$	Расстояние между оконечным устройством IoT I_i и БПЛА D_j
$L_{Ii,Gj}$	Расстояние между оконечным устройством IoT I_i и IoT-шлюзом G_j
$\mathcal{d}Di$	Продолжительность полета БПЛА D_i
h_{Gj}	Высота IoT-шлюза G_i
x_{Gi}, y_{Gi}	x-y местоположения IoT-шлюза G_i
t_s	Продолжительность временного интервала
γ_{nts}^{Di}	Коэффициент усиления канала БПЛА D_i для n-го кадра
γ_0	Принимаемая мощность на эталонном расстоянии 1 м
$\Omega_{C\text{-}Ij,Di}$	Энергия, потребляемая в восходящем соединении между IoT-устройством I_j и БПЛА D_i
$\Omega_{C\text{-}Di,Ij}$	Энергия, потребляемая в нисходящем соединении между БПЛА D_i и IoT-устройством I_j
σ^2	Мощность шума
B	Ширина полосы шума
S	Количество данных в задаче, т. е. размер в байтах
Z	Объем данных, полученных в результате обработки выгруженной задачи
$\Omega_{M\text{-}Di}$	Энергозатраты БПЛА D_i на механические операции
V_{Di}^{nts}	Вектор скорости БПЛА D_i на n-ом кадре
m_{Di}	Масса БПЛА D_i
UAV_{ID}	Идентификация БПЛА
$D_{clus,Ii}$	Решение IoT-устройства I_i присоединиться к кластеру БПЛА
NC_{bit}	Требуемые (CPU) циклы для обработки одного бита данных
$D_{h\text{-}IoT}$	Время, необходимое для обработки вычислительной задачи локально на IoT-устройстве
$\Omega_{exec\text{-}IoT}$	Энергия, необходимая для обработки вычислительной задачи локально на IoT-устройстве
$\Omega_{available\text{-}aft\text{-}IoT}$	Оставшаяся энергия IoT-устройства после обработки задачи с использованием IoT-ресурсов
$\Omega_{available\text{-}bef\text{-}IoT}$	Доступный уровень энергии IoT-устройства перед обработкой выгруженной задачи
$\Omega_{Th\text{-}IoT}$	Пороговый уровень энергии оконечного IoT-устройства
$\delta_{alloc\text{-}IoT}$	Ресурсы обработки оконечного IoT-устройства, выделенные для вычислительной задачи
ω_{IoT}	Общие ресурсы обработки оконечного IoT-устройства
$\beta_{F\text{-}IoT}$	Бинарное решение о выгрузке определяется IoT-устройством
$\beta_{\Omega\text{-}IoT}$	Бинарное энергетическое решение о выгрузке, принятое IoT-устройством
$B_{D\text{-}IoT}$	Бинарное решение о выгрузке по времени определяется IoT-устройством
τ	Пороговое время вычислительной задачи поддерживает требования параметров качества обслуживания (QoS)
$D_{h\text{-}G\text{-}MEC}$	Общее время, необходимое для обработки вычислительной задачи на сервере G-MEC, запрошенное оконечным IoT-устройством
$D_{exec\text{-}G\text{-}MEC}$	Общее время, необходимое для выполнения вычислительной задачи на сервере G-MEC
$\delta_{alloc\text{-}G\text{-}MEC}$	Ресурсы обработки блока G-MEC, выделенные для вычислительной задачи
$\omega_{G\text{-}MEC}$	Общие ресурсы обработки блока G-MEC
$T_{up\text{-}Ij\text{-}Gi}$	Общее время восходящей линии связи между оконечным IoT-устройством I_j и IoT-шлюзом G_i
$T_{up\text{-}Gi\text{-}Ij}$	Общее время нисходящего канала между IoT-шлюзом G_i и оконечным IoT-устройством I_j
$T_{u\text{-}tx\text{-}Ij\text{-}Gi}$	Общая задержка передачи по восходящей линии связи между оконечным IoT-устройством I_j и IoT-шлюзом G_i
$T_{d\text{-}tx\text{-}Gi\text{--}Ij}$	Общая задержка передачи по нисходящей линии связи между IoT-шлюзом G_i и оконечным IoT-устройством I_j
T_{pro}	Задержка распространения
$\beta_{AF\text{-}G\text{-}MEC}$	Бинарное решение о принятии запроса на выгрузку, принятое G-MEC
$D_{h\text{-}A\text{-}MEC}$	Общее время, необходимое для обработки вычислительной задачи на сервере A-MEC, выгруженной оконечным IoT-устройством
$t_{up\text{-}Ij\text{-}Di}$	Общее время восходящей линии связи, необходимое для выгрузки данных задачи с оконечного IoT-устройства I_j и БПЛА D_i
$t_{dw\text{-}Di\text{-}Ij}$	Общее время нисходящей линии связи, необходимое для отправки результата с БПЛА D_i на IoT-устройство I_j

Обозначение	Описание
$D_{exec\text{-}A\text{-}MEC}$	Общее время, необходимое для выполнения вычислительной задачи на сервере A-MEC
$\delta_{alloc\text{-}A\text{-}MEC}$	Ресурсы обработки блока A-MEC, выделенные для вычислительной задачи
$\omega_{A\text{-}MEC}$	Общие ресурсы обработки блока A-MEC
$T_{u\text{-}tx\text{-}Ij\text{-}Di}$	Суммарная задержка передачи по восходящей линии связи между оконечным IoT-устройством I_j и БПЛА D_i
$T_{d\text{-}tx\text{-}Di\text{-}Ij}$	Суммарная задержка передачи по нисходящей линии связи между БПЛА D_i и оконечным IoT-устройством I_j
$\beta_{D\text{-}A\text{-}MEC}$	Бинарное решение о времени принятия запроса на выгрузку, принятое A-MEC
$\beta_{\Omega\text{-}A\text{-}MEC}$	Бинарное энергетическое решение о принятии запроса на выгрузку, принятое A-MEC
$I()$	Индикатор режима
$\Omega_{Th\text{-}IoT}$	Пороговый уровень энергии оконечного IoT-устройства
$\Omega_{available\text{-}aft\text{-}A\text{-}MEC}$	Остаток энергии БПЛА после выполнения задачи с использованием ресурсов A-MEC
$\Omega_{Th\text{-}A\text{-}MEC}$	Пороговый уровень энергии БПЛА
$\Omega_{exec\text{-}A\text{-}MEC}$	Потребление энергии из-за вычислений при выполнении выгруженной задачи в A-MEC
$\beta_{AF\text{-}A\text{-}MEC}$	Бинарное решение о принятии запроса на выгрузку, принятое A-MEC
Ф	Набор позиций БПЛА, D_i, в каждом временном интервале
α_D	Весовой коэффициент времени
α_Ω	Весовой коэффициент энергии
$V_{\max}$	Максимальная скорость БПЛА
$x_{\min\text{-}Di}$, $x_{\max\text{-}Di}$, $y_{\min\text{-}Di}$, and $y_{\max\text{-}Di}$	Минимальное и максимальное положенияв координатах xy БПЛА D_i
$\delta_{alloc\text{-}A\text{-}MEC\text{-}\max}$	Максимальные вычислительные ресурсы A-MEC, которые могут быть выделены для вычислений
$\delta_{alloc\text{-}A\text{-}MEC\text{-}Tj}$	Ресурсы обработки A-MEC, выделенные для вычислений, связанных с задачей T_j
P^i	Вектор положения сальп
n_s	Общее количество сальп
F	Положение источника энергии
U_b	Верхняя граница, связанная с каждым измерением в пространстве поиска
L_b	Нижняя граница, связанная с каждым измерением в пространстве поиска
C_1, C_2, and C_3	Коэффициенты SSA
Δ	Текущая версия SSA
$\Delta_{\max}$	Максимальное количество итераций SSA
m	Количество измерений в пространстве поиска
$R_{b\text{-}Di\text{-}nts}$	Достижимая скорость выгрузки в n-ом временном интервале

В воздушном сегменте создается набор БПЛА D (каждый БПЛА имеет граничный сервер MEC-A с возможностью ресурсов обработки $R_{MEC\text{-}A}$), определенный в следующем виде:

$$D = \{D_1, D_2, D_3, \cdots, D_N\} \quad \text{при} \quad \forall N \in \mathbb{R}, \qquad (1)$$

где N – общее количество развернутых БПЛА в воздушной плоскости.

Трехмерная декартова система координат с координатами (x, y, h) используется для определения местоположения БПЛА и IoT-устройств, как показано на рисунке 2.

Местоположение БПЛА характеризуется двумерной наземной координатой x и y и высотой h. Таким образом, текущее местоположение БПЛА D_i равно P^{Di} и характеризуется x_{Di}, y_{Di} и h_{Di}. Оконечные IoT-устройства расположены в плоскости xy в разных местах. Каждый БПЛА движется по траектории T^{Di} с фиксированной высотой h_{Di}, имея два набора местоположений xy, G_x^{Di} и G_y^{Di}.

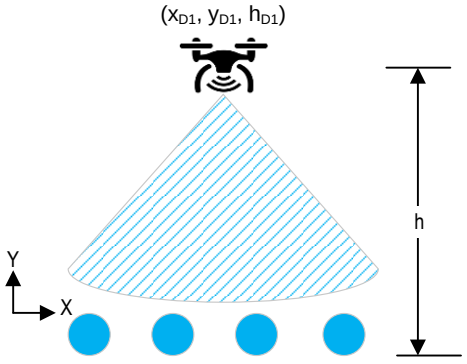

Рис. 2. БПЛА в трехмерных координатах

Fig. 2. UAV in Three-Dimensional Coordinates

Сети IoT используют набор оконечных устройств I, определенный в (2). Расположение IoT-устройств I_j равно P^{Ij} и характеризуется x_{Ij} и y_{Ij}. Некоторые устройства сгруппированы в кластеры БПЛА, а остальные узлы взаимодействуют непосредственно со шлюзами. Набор развернутых кластеров равен C, определенному в (3, 4).

$$I = \{I_1, I_2, I_3, \cdots, I_M\} \quad \text{при} \quad \forall M \in \mathbb{R}, \tag{2}$$

$$C = \{C_1, C_2, C_3, \cdots, C_N\} \quad \text{при} \quad \forall N \in \mathbb{R}, \tag{3}$$

$$C_i = \{D_i, {CM_1}^{Ci}, {CM_2}^{Ci}, \cdots, {CM_{ki}}^{Ci}\} \quad \text{при } \forall K_i \in \mathbb{R}, C_i \subset C, D_i \in D, \ {CM_j}^{Ci} \in I, \tag{4}$$

где M – общее количество развернутых IoT-устройств; ${CM_j}^{Ci}$ – j-й член i-го кластера; K_i – общее количество членов i-го кластера.

Набор IoT-шлюзов равен G и определяется следующим образом:

$$G = \{G_1, G_2, G_3, \cdots, G_W\} \quad \text{при} \quad \forall W \in \mathbb{R}, \tag{5}$$

где W — общее количество IoT-шлюзов.

Расстояние между оконечным устройством Интернета Вещей I_i и БПЛА D_j равно $L_{Ii,Dj}$, в соответствии с (6), а расстояние до шлюза G_j, равно $L_{Ii,Gj}$ как это определено в (7).

$$L_{I_{i,D_j}} = \sqrt{\left(x_{Ii} - x_{Dj}\right)^2 + \left(y_{Ii} - y_{Dj}\right)^2 + {h_{D_j}}^2}, \tag{6}$$

$$L_{I_{i,G_j}} = \sqrt{\left(x_{Ii} - x_{Gj}\right)^2 + \left(y_{Ii} - y_{Gj}\right)^2 + {h_{G_j}}^2}. \tag{7}$$

Каждый БПЛА имеет продолжительность полета $\mathbb{T}^{Di}$, которая зависит от энергии, потребляемой в процессах связи, вычислений и полета. Процессы, происходящие на одном БПЛА, отличаются от процессов, происходящих на другом; таким образом, продолжительность полета разных БПЛА неодинакова. Общее время полета разбивается на временные интервалы с длительностью интервала t_s, при этом процессы связи и вычислений выполняются в этих интервалах. В течение каждого временного интервала t_s местоположение БПЛА аппроксимируется как фиксированное. Набор местоположений xy соответствует траектории БПЛА D_i и может быть определен следующим образом:

$${G_x}^{Di} = \{{x_{Di}}^{ts}, {x_{Di}}^{2ts}, {x_{Di}}^{3ts}, \cdots, {x_{Di}}^{qts}\} \quad \forall D_i \in D, \tag{8}$$

$${G_y}^{Di} = \{{y_{Di}}^{ts}, {y_{Di}}^{2ts}, {y_{Di}}^{3ts}, \cdots, {y_{Di}}^{qts}\} \quad \forall D_i \in D. \tag{9}$$

При использовании неортогонального множественного доступа (NOMA, *аббр. от англ.* Non-Orthogonal Multiple Access) все оконечные устройства одновременно используют общую длительность кадра t_s для передачи данных по восходящему и нисходящему каналам. Будем предполагать, что связь между БПЛА и IoT-устройствами осуществляется в пределах прямой видимости.

Таким образом, усиление канала для n-го кадра можно рассчитать по выражению (10), где γ_0 – коэффициент усиления канала на эталонном расстоянии в один метр, когда передаваемая мощность составляет 1 Вт.

Модель энергопотребления

БПЛА потребляют энергию тремя способами: потребление энергии из-за вычислений, потребление энергии из-за связи и потребление энергии из-за механических операций. Энергия, потребляемая из-за вычислений, рассчитывается на основе количества выгружаемых данных. Энергия, потребляемая из-за связи, основана на объеме данных, передаваемых по восходящей и нисходящей линиям связи [23]. Энергия, потребляемая в восходящем соединении между IoT-устройством I_j и БПЛА D_i, равна $\Omega_{C\text{-}Dj,Di}$ и рассчитывается на основе объема выгружаемых данных S, как отмечено в (11). Отметим, что энергия, потребляемая в нисходящем соединении между БПЛА D_i и IoT-устройством I_j, составляет $\Omega_{C\text{-}Di,Ij}$ и рассчитывается на основе объема данных, не связанных с обработкой запроса на выгрузку Z, как в (12). Канал моделируется аддитивным белым гауссовым шумом мощности σ^2 и шириной полосы B.

Энергия, потребляемая при связи по восходящей и нисходящей линиям связи, рассчитывается в виде выражений (11, 12). Потребление энергии из-за механических операций $\Omega_{M\text{-}Di}$ смоделировано в [24] и может быть рассчитано для длительности временного интервала на основе вектора скорости БПЛА V, и массы БПЛА m_{Di} по выражениям (13, 14).

$$\gamma_{nts}{}^{Di}\left({x_{Di}}^{nts}, {y_{Di}}^{nts}\right) = \frac{\gamma_0}{\left({x_{Di}}^{nts} - {x_{Ij}}^{nts}\right)^2 + \left({y_{Di}}^{nts} - {y_{Ij}}^{nts}\right)^2} \quad \text{при} \quad \forall n \in \mathbb{R}, I_j \in I, D_i \in D, \tag{10}$$

$$\Omega_{C-Ij,Di}\left(s, {x_{Di}}^{nts}, {y_{Di}}^{nts}\right) = \frac{\sigma^2 t_s \left(2^{S/Bt_s} - 1\right)}{\gamma_{nts}{}^{Di}\left({x_{Di}}^{nts}, {y_{Di}}^{nts}\right)} \cdot \frac{1}{\left(1 - \left(2^{S/Bt_s} - 1\right)\right)} \quad \forall n \in \mathbb{R}, I_j \in I, D_i \in D, \tag{11}$$

$$\Omega_{C-Di,Ij}\left(Z, {x_{Di}}^{nts}, {y_{Di}}^{nts}\right) = \frac{\sigma^2 t_s \left(2^{Z/Bt_s} - 1\right)}{\gamma_{nts}{}^{Di}\left({x_{Di}}^{nts}, {y_{Di}}^{nts}\right)} \cdot \frac{1}{\left(1 - \left(2^{Z/Bt_s} - 1\right)\right)} \quad \forall n \in \mathbb{R}, I_j \in I, D_i \in D, \tag{12}$$

$$\Omega_{M-Di}(x_{Di}{}^{nts}, y_{Di}{}^{nts}) = 0{,}5 m_{Di} t_s \|v_{Di}{}^{nts}\|^2 \; \forall\, n \in \mathbb{R}, D_i \in D, \quad (13)$$

$$\|v_{Di}{}^{nts}\| = \sqrt{(x_{Di}{}^{nts} - x_{Di}{}^{nts+1})^2 + (y_{Di}{}^{nts} - y_{Di}{}^{nts+1})^2} \Big/ t_s \; \forall\, n \in \mathbb{R},\; D_i \in D. \quad (14)$$

Формирование кластера

Кластеры формируются по раундам, по две фазы на каждый раунд; фаза формирования кластера и фаза обеспечения связи. В начале каждого раунда каждый БПЛА транслирует сообщение о соединении в формате, представленном в таблице 2.

ТАБЛИЦА 2. Формат сообщения о присоединении

TABLE 2. Format of Join Message

Сообщение о присоединении				
Идентификация				Технические характеристики канала
Идентификатор, ID	Расположение			
$БПЛА_{ID}$	x_{Di}	y_{Di}	H_{Di}	

Сообщение о присоединении содержит идентификацию БПЛА: идентификатор БПЛА (UAV_{ID}), координаты местоположения БПЛА (x, y и h) и технические характеристики канала связи. Оконечные IoT-устройства получают сообщения о присоединении от БПЛА и решают, следует ли присоединиться к БПЛА или взаимодействовать с IoT-шлюзом.

Оконечное IoT-устройство вычисляет расстояние до БПЛА $L_{Ii,Dj}$ на основе (6) и расстояние до IoT-шлюза $L_{Ii,Gj}$ на основе (7) и выбирает кратчайший путь. IoT-устройства, которые получают несколько сообщений о присоединении, выбирают тот, у которого кратчайший путь связи, и сравнивают этот путь с путем к ближайшему IoT-шлюзу. Более того, IoT-устройства вне зоны действия IoT-шлюзов, выбирают ближайший БПЛА для присоединения.

Таким образом, решение о присоединении к кластеру БПЛА D_{clus} принимается согласно выражению:

$$D_{clus-I_i} = I\left(L_{I_{i,G_j}}, \min\left(L_{I_{i,D_j}}\right)\right) = \begin{cases} 0 & IF\ \ (L_{I_{i,G_j}} \le \min(L_{I_{i,D_j}})) \\ 1 & IF\ \ (L_{I_{i,G_j}} > \min(L_{I_{i,D_j}})) \vee (L_{I_{i,G_j}} \to \infty) \end{cases} \forall\ D_j \in D, \qquad I_i \in I \quad (15)$$

Оконечные IoT-устройства при наличии кратчайшего пути к выбранному БПЛА отправляют ответное сообщение с положительным решением о кластеризации. Таким образом, формируются кластеры, в которых БПЛА выступает в роли головного узла кластера, и этап формирования кластера заканчивается. Оконечные IoT-устройства взаимодействуют и обмениваются данными с головным узлом кластера (БПЛА), который управляет кластером на фазе обеспечения связи.

Метод выгрузки трафика

Рассмотрим разработанный метод выгрузки трафика с наземной сети на мобильные серверы граничных вычислений на БПЛА. Оконечные IoT-устройства обрабатывают данные, используя имеющиеся у них ресурсы, т. е. осуществляется локальное выполнение требуемых задач. Такое решение допустимо для только для задач, требующих ограниченных ресурсов, в том числе и энергоресурсов. Задачи, требующие больших ресурсов, следует переносить на G-MEC или на A-MEC. Разработанный метод выгрузки является бинарным, поскольку предполагается, что БПЛА используются только для поддержки высокоплотных и сверхплотных сетей Интернета Вещей и не имеют иных локальных задач. На рисунке 3 представлен алгоритм реализации разработанного метода выгрузки с указанием основных шагов. Оконечные устройства и блоки MEC используют ранее разработанную структуру, представленную в [25].

БПЛА можно рассматривать как мобильные серверы граничных вычислений, которые получают выгруженные задачи от оконечных IoT-устройств и либо обрабатывают их сами, либо переносят на другие граничные вычислительные узлы. Выгрузка трафика между наземным и воздушным сегментами осуществляется по восходящим и нисходящим интерфейсам с использованием дуплекса с частотным разделением. NOMA используется для улучшения связности и спектральной эффективности обслуживаемых кластеров.

Разработанный метод выгрузки состоит из трех уровней, каждый из которых представляет собой сервер граничных вычислений. На оконечных устройствах есть программный профилировщик, который вычисляет спецификации задачи, включая объем данных, т. е. размер в байтах S и требуемые циклы центрального процессора (CPU, *аббр. от англ.* Central Processing Unit) для обработки одного бита данных NC_{bit}.

Планировщик ресурсов оконечного устройства предоставляет механизму принятия решений доступные в настоящее время ресурсы для решаемой задачи. Механизм принятия решений оконечного устройства рассчитывает необходимое время для обработки задачи $D_{h\text{-}IoT}$, необходимую энергию для

обработки задачи $\Omega_{exec\text{-}oT}$ и остаточную энергию после обработки задачи с использованием ресурсов IoT, $\Omega_{available\text{-}aft\text{-}IoT}$ следующим образом:

$$D_{h-IoT} = \frac{S.NC_{bit}}{\delta_{alloc-IoT}}, \quad \delta_{alloc-IoT} \in \omega_{IoT}, \qquad (16)$$

$$\Omega_{exec-IoT} = S.NC_{bit}\mathrm{E}_{Cyc-IoT} \qquad (17)$$

$$\Omega_{available-aft-IoT} = \\ = \Omega_{available-bef-IoT} - \Omega_{exec-IoT}. \qquad (18)$$

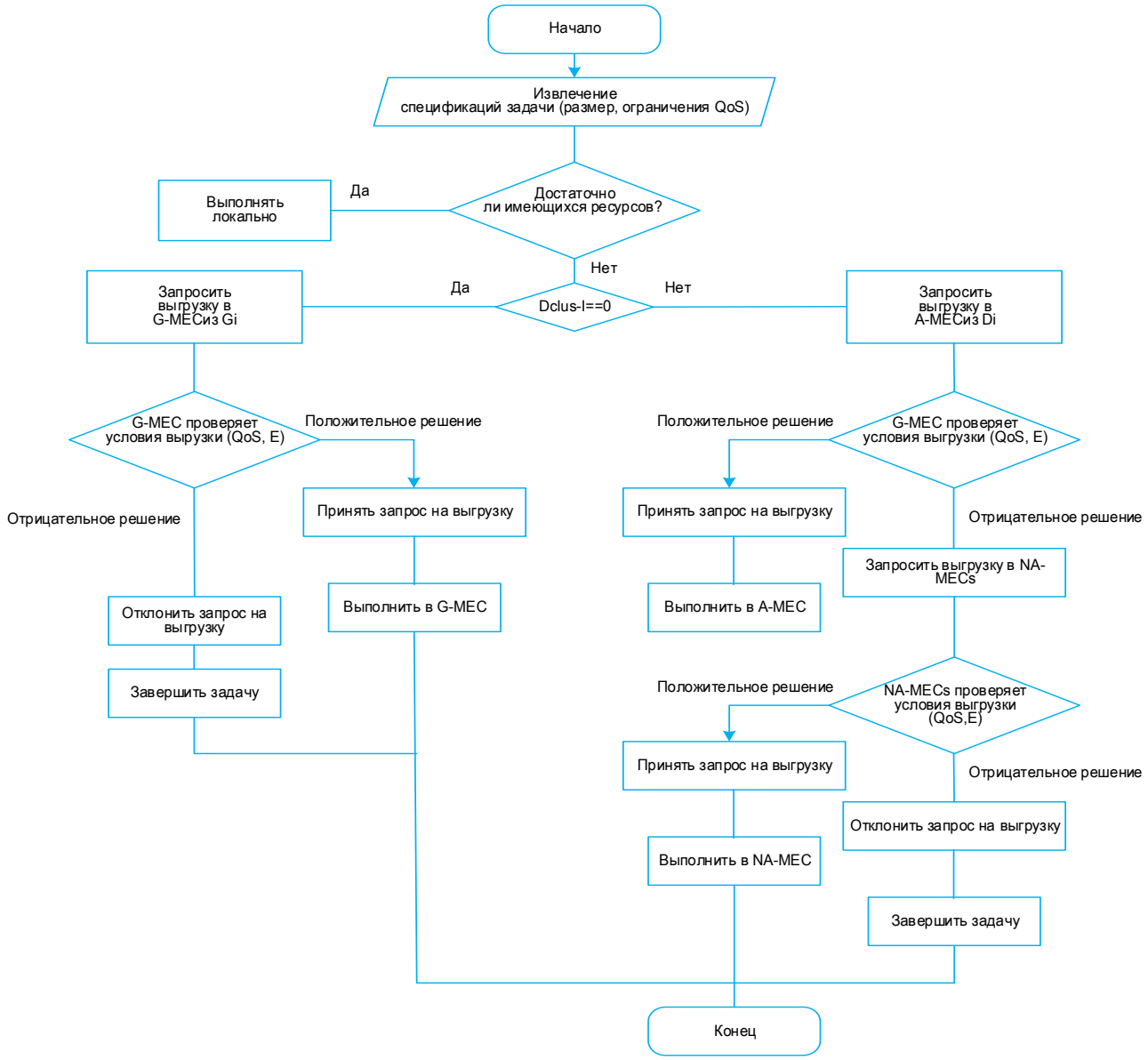

Рис. 3. Алгоритм выгрузки трафика для разработанного метода

Fig. 3. Offloading Traffic Algorithm for the Developed Method

На основе рассчитанных параметров QoS, задержки и энергопотребления принимается решение об обработке на месте или передачи задачи на граничный сервер MEC.

Бинарное решение выгрузки $B_{F\text{-}IoT}$ рассчитывается следующим образом:

$$\beta_{\Omega-IoT} = I\left(\Omega_{available-aft-IoT}, \Omega_{Th-IoT}\right) = \begin{cases} 0 & IF \ \left(\Omega_{available-aft-IoT} \le \Omega_{Th-IoT}\right) \\ 1 & IF \ \left(\Omega_{available-aft-IoT} > \Omega_{Th-IoT}\right) \end{cases} \qquad (19)$$

$$\beta_{\mathrm{D}-IoT} = I(D_{h-IoT}, \tau) = \begin{cases} 0 & IF \ \left(D_{h-IoT} \le \tau\right) \\ 1 & IF \ \left(D_{h-IoT} > \tau\right) \end{cases} \qquad (20)$$

$$\beta_{\mathrm{F}-IoT} = \beta_{\Omega-IoT} \vee \beta_{\mathrm{D}-IoT}. \qquad (21)$$

Пусть доступных ресурсов, включая энергопотребление, оконечного устройства достаточно для выполнения задачи. В этом случае бинарное решение о выгрузке минимально, т. е. $\beta_{F\text{-}IoT} = 0$, и устройство не выполняет выгрузку. Однако, если ресурсов недостаточно, включая ограничения по энергопотреблению устройства, конечное устройство решает передать задачу, т. е. $\beta_{F\text{-}IoT} = 1$, на соответствующий граничный сервер MEC.

Устройство, подключенное к IoT-шлюзу, т. е. не являющееся членом кластера БПЛА, запрашивает выгрузку в G-MEC. Сообщение с запросом содержит необходимую информацию для обработки задачи и идентификации устройства. G-MEC обрабатывает полученные запросы на выгрузку и отправляет информацию о задаче в механизм принятия решений: принять или отклонить запрос. Это зависит от доступных вычислительных ресурсов G-MEC и ограничений QoS запрашиваемой задачи. Планировщик ресурсов G-MEC предоставляет механизму принятия решений доступные ресурсы для принятия решения о приеме или отклонении запрошенной задачи.

Механизм принятия решений G-MEC вычисляет время, необходимое для обработки задачи по запросу $D_{hG\text{-}MEC}$, следующим образом:

$$D_{exec-G-MEC} = \frac{S.NC_{bit}}{\delta_{alloc-G-MEC}}, \quad \delta_{alloc-G-MEC} \in \omega_{G-MEC}, \tag{22}$$

$$D_{h-G-MEC} = D_{exec-G-MEC} + T_{up-Ij-Gi} + T_{dw-Gi-Ij}, \tag{23}$$

$$T_{up-Ij-Gi} = T_{u-tx-Ij-Gi} + T_{pro}, \tag{24}$$

$$T_{dw-Gi-Ij} = T_{d-tx-Gi-Ij} + T_{pro}. \tag{25}$$

Затем механизм принятия решений определяет бинарное решение о принятии или отклонении запроса на выгрузку $\beta_{AF\text{-}G\text{-}MEC}$ путем сравнения общего времени, необходимого для обработки запрошенной задачи $D_{hG\text{-}MEC}$, со временем QoS. При отрицательном решении задача завершается.

$$\beta_{AF-G-MEC} = I(D_{h-G-MEC}, \tau) = \begin{cases} 1 & IF \ (D_{h-G-MEC} \le \tau) \\ 0 & IF \ (D_{h-G-MEC} > \tau) \end{cases}. \tag{26}$$

Выгрузка в воздушный сегмент

В случае, если механизм принятия решений устройств – членов кластера, подключенных к БПЛА, отклоняет локальное выполнение задачи, то устройства запрашивают выгрузку у соответствующего БПЛА через интерфейс восходящей линии связи. Сервер граничных вычислений A-MEC беспилотного летательного аппарата обрабатывает полученные запросы на выгрузку от обслуживаемых оконечных устройств, путем извлечения информации об устройстве и выполняемой задаче и предоставляет их механизму принятия решений A-MEC.

Механизм принятия решений также получает информацию о доступных в данный момент времени ресурсах от планировщика ресурсов и вычисляет время, необходимое для обработки запрошенной задачи, $D_{hA\text{-}MEC}$. Это время включает время восходящей линии связи, необходимое для выгрузки данных задачи в БПЛА $t_{up\text{-}Ij\text{-}Di}$, время нисходящей линии связи для отправки результата $t_{dw\text{-}Di\text{-}Ij}$, и время вычислений $t_{exec\text{-}A\text{-}MEC}$. Анализ расчетов времени передачи по восходящей и нисходящей линиям связи представлен в Приложении I к статье [26].

$$D_{h-A-MEC} = T_{up-Ij-Di} + T_{dw-Di-Ij} + D_{exec-A-MEC}, \tag{27}$$

$$D_{exec-A-MEC} = \frac{S.NC_{bit}}{\delta_{alloc-A-MEC}}, \quad \delta_{alloc-A-MEC} \in \omega_{A-MEC}, \tag{28}$$

$$T_{up-Ij-Di} = T_{u-tx-Ij-Di} + T_{pro}, \tag{29}$$

$$T_{dw-Di-Ij} = T_{d-tx-Di-Ij} + T_{pro}. \tag{30}$$

Далее механизмом принятия решений вычисляется время в двоичном формате и принимается решение о принятии или отклонении запроса на выгрузку $\beta_{DA\text{-}MEC}$ путем сравнения общего времени, необходимого для обработки запрошенной задачи, $D_{hA\text{-}MEC}$, со временем QoS.

$$\beta_{D-A-MEC} = I(D_{h-A-MEC}, \tau) = \begin{cases} 1 & IF \ (D_{h-A-MEC} \le \tau) \\ 0 & IF \ (D_{h-A-MEC} > \tau) \end{cases}. \tag{31}$$

При отрицательном решении, т. е. $\beta_{DA\text{-}MEC} = 0$, БПЛА D_i запрашивает ресурсы у соседнего БПЛА. БПЛА Di передает запрос на выгрузку БПЛА в рое. Связь между БПЛА осуществляется по ранее разработанной схеме маршрутизации, представленной в [27]. Кроме того, бинарное решение о принятии запроса по выгрузке $\beta_{\Omega\text{-}A\text{-}MEC}$ рассчитывается путем сравнения оставшейся энергии БПЛА D_i после выполнения запрошенной задачи $\Omega_{available\text{-}aft\text{-}A\text{-}MEC}$ с пороговым уровнем энергии БПЛА φ $\Omega_{Th\text{-}A\text{-}MEC}$. БПЛА сначала рассчитывает расход энергии на вычисление запрашиваемой задачи $\Omega_{exec\text{-}A\text{-}MEC}$, как в (32), и расход энергии на передачу результатов расчета $\Omega_{C\text{-}Di,Ij}$, как в (12):

$$\Omega_{exec-A-MEC} = S.NC_{bit} \mathrm{E}_{Cyc-A-MEC}, \tag{32}$$

$$\Omega_{available-aft-A-MEC} = \Omega_{available-bef-A-MEC} - \Omega_{exec-A-MEC} - \Omega_{C-Di,Ij}\left(R, {x_{Di}}^{nts}, {y_{Di}}^{nts}\right), \tag{33}$$

$$\beta_{\Omega-A-MEC} = I\left(\Omega_{available-aft-A-MEC}, \Omega_{Th-A-MEC}\right) = \begin{cases} 1 & IF \ (\Omega_{available-aft-A-MEC} \le \Omega_{Th-A-MEC}) \\ 0 & IF \ (\Omega_{available-aft-A-MEC} > \Omega_{Th-A-MEC}) \end{cases}, \tag{34}$$

$$\beta_{AF-A-MEC} = \beta_{\Omega-A-MEC} \wedge \beta_{D-A-MEC}. \tag{35}$$

При отказе о приеме запроса на выгрузку БПЛА запрашивает выгрузку у соседнего БПЛА в рое с до-

ступными ресурсами. Беспилотный летательный аппарат D_i передает запрос на выгрузку соседним БПЛА. Этот запрос принимается и обрабатывается с помощью описанного ранее алгоритма. Соседние узлы отправляют ответы для принятия или отклонения запросов на выгрузку, а БПЛА завершает задачу или выгружает ее на основе полученных ответов.

Оптимизация среднего энергопотребления и задержки для обработки задач, выгруженных на БПЛА, с помощью хаотического «роя сальп»

Описанная ранее схема выгрузки потребляемой энергии и задержка, необходимые для выполнения вычислительных задач, в основном зависит от местоположения БПЛА. Основная цель этого раздела – оптимизировать энергопотребление и минимизировать среднюю задержку для выполнения задач выгрузки.

Задача оптимизации определяется следующим по выражению (36) при ограничениях (37–42), где φ – множество позиций БПЛА D_i в каждом временном интервале; α_D и α_Ω – весовые коэффициенты времени и энергии, соответственно; $\delta_{alloc\text{-}A\text{-}MEC\text{-}\max}$ – максимальные вычислительные ресурсы A-MEC, которые могут быть выделены для вычислений; $V_{\max}$ – максимальная скорость БПЛА; $x_{\min\text{-}Di}$, $x_{\max\text{-}Di}$, $y_{\min\text{-}Di}$, $y_{\max\text{-}Di}$ – минимальные и максимальные координаты *xy* БПЛА *Di*.

$$\min_{\varphi}\Big(\sum_{j=1}^{k}\alpha_D D_{h-A-MEC}{}^{Di,Ij}+\sum_{j=1}^{k}\alpha_\Omega\big(\Omega_{C-Di,Ij}\big(Z,x_{Di}{}^{nts},y_{Di}{}^{nts}\big)+\Omega_{C-Ij,Di}\big(s,x_{Di}{}^{nts},y_{Di}{}^{nts}\big)+\Omega_{exec-A-MEC}\big)\Big), \quad (36)$$

C1:

$$\sqrt{(x_{Di}{}^{nts}-x_{Di}{}^{nts+1})^2+(y_{Di}{}^{nts}-y_{Di}{}^{nts+1})^2}\Big/_{t_s}\le V_{\max}\quad \forall\, n\in\mathbb{R},\ D_i\in D, \quad (37)$$

C2:

$$X_{\min-Di}<X_{Di}<X_{\max-Di}\quad \forall\ D_i\in D, \quad (38)$$

C3:

$$Y_{\min-Di}<Y_{Di}\le Y_{\max-Di}\quad \forall\ D_i\in D, \quad (39)$$

C4:

$$\sum_{j=1}^{N_{f-Di}}\delta_{alloc-A-MEC-Tj}\le\delta_{alloc-A-MEC-\max}\quad\forall\ \delta_{alloc-A-MEC-Tj},\ \delta_{alloc-A-MEC-\max}\in\omega_{A-MEC}, \quad (40)$$

C5:

$$\beta_{AF-A-MEC}\in\{0,1\}, \quad (41)$$

C6:

$$\sum_{j=1}^{N_{f-Di}}\beta_{AF-A-MEC-Tj}\le k_i. \quad (42)$$

Первое ограничение C1 вводится для того, чтобы скорость БПЛА в любой момент времени не превышала максимальной. Ограничения C2 и C3 вводятся для обеспечения того, чтобы каждый БПЛА покрывал только запланированную область полета и избегал столкновений между БПЛА роя. C4 гарантирует, что вычислительные ресурсы, направленные на обработку различных параллельно выполняемых задач, меньше, чем максимальные ресурсы БПЛА, выделенные для вычислений.

Хаотический «рой сальп» для получения оптимальных позиций БПЛА

SSA – это основанный на популяции сальп метаэвристический метод, который имитирует поведение сальп в океанах [28]. Это относительно новый класс оптимизации роя частиц, который имитирует семейство сальп [29]. Популяция сальп состоит из двух типов: сальпы-лидера и сальпы-последователя. Сальпа-лидер является первой сальпой в цепочке и отвечает за руководство роем. Сальпы, которые следуют за сальпой-лидером к источнику пищи, являются сальпами-последователями. Каждая сальпа занимает *m*-мерное пространство поиска, где *m* обозначает количество переменных в данной задаче, аналогично другим алгоритмам на основе роя [28]. Сальпы имеют вектор положения P^i из n_s элементов, который представляет общее количество развернутых роев и определяется следующим образом:

$$P^i=\{P_1{}^i,P_2{}^i,P_3{}^i,\cdots,P_m{}^i\},\ \ i=1,\ 2,\ 3,\ 4,\cdots,n_s. \quad (43)$$

Сальпы ищут «место пищи» в пространстве и в результате поиска обновляют свои позиции до тех

пор, пока не достигнут источника пропитания, т. е. оптимального решения. Согласно уравнениям (44, 45), где F указывает положение пищи; U_b и L_b – верхняя и нижняя границы, связанные с каждым измерением в пространстве поиска, сальпа-лидер сначала обновляет свое положение; затем все следующие сальпы выслеживают его.

$$P_j^1 = \begin{cases} F_j + C_1\left(\left(U_{b_j} - U_{l_j}\right)C_2 + L_{b_j}\right), & C_3 \geq 0 \\ F_j - C_1\left(\left(U_{b_j} - U_{l_j}\right)C_2 + L_{b_j}\right), & C_3 < 0 \end{cases}, \quad j = 1, 2, 3, \cdots, m, \tag{44}$$

$$P_j^z = \frac{1}{2}\left(P_j^z + P_j^{z-1}\right) \quad 2 \leq z < n_s, \ z \in \mathbb{Z}^+, \tag{45}$$

Работа алгоритма основана на коэффициентах C_1, C_2 и C_3, которые используются для обновления положения сальпы при каждой итерации.

Коэффициент C_1 вводится для поддержания уровня стабильности между разведкой и эксплуатацией и определяется следующим образом:

$$C_1 = 2e^{-\left(\frac{4\Delta}{\Delta_{\max}}\right)^2}, \tag{46}$$

где Δ – текущая итерация; $\Delta_{\max}$ – максимальное количество итераций.

Другие коэффициенты могут быть обновлены с использованием случайных функций со значениями от 0 до 1. Для предлагаемой модели мы вводим хаотические карты для обновления коэффициентов рассматриваемого SSA [30].

Рассматриваемая карта является логистической и определяется следующим образом:

$$y(\Delta + 1) = ay(\Delta)[1 - y(\Delta)], a = 4{,}0, y(0) = 0{,}7. \tag{47}$$

Таким образом, C_2 может быть обновлен следующим образом:

$$C_2{}^{\Delta} = y(\Delta). \tag{48}$$

В SSA параллельно выполняет поиск в рабочем пространстве, чтобы получить решение, представляющее собой набор оптимальных положений роя БПЛА в течение каждого временного интервала. Алгоритм 1 выполняет поставленную задачу и получает оптимальное решение. Алгоритм используется во вложенном цикле с ранее представленным алгоритмом выгрузки.

Алгоритм «роя сальп» начинается с инициализации начальных параметров SSA на основе хаотических карт, включая нижнюю и верхнюю границу, а также максимальное количество итераций. Затем он случайным образом инициализирует n_s сальпы, представляющие количество местоположений БПЛА. Приспособленность различных сальп, в том числе ведущих и ведомых, определяется с помощью (36). «Место пропитания» считается местом с наибольшей приспособленностью сальпы (роя БПЛА). Параметры SSA обновляются, и в результате – местоположения сальпы. Процесс обновления местоположения сальпы повторяется до тех пор, пока не будет найдено оптимальное решение или не будет достигнуто максимальное количество итераций.

АЛГОРИТМ 1. Хаотическая SSA для оптимального положения БПЛА

```
Initialize Ub, Lb, Δmax, m, ns
Initialize positions of salps Pj (j = 1, 2, 3, ..., ns)
While (Δ ≤ Δmax)
    Calculate the fitness function of each salp position
    F = The best salp position
    Update the value of C1 (using (46))
    Calculate the logistic map's current value y(Δ)
    For (m = 1: m ≤ ns) do
     if (m == 1)
        Update the position of the leader (Pj^1) (using (44))
     else
        Update the position of follower salp (Pj^m) (using (45))
     end if
    end for
    Adjust salps based on Ub and Lb
    Call offloading algorithm
    Update the best salp based on the results of offloading algorithm
    Δ← Δ+1
Return F
```

Результаты моделирования

Разработанные решения были промоделированы в среде NS-3 на основе LIMosim framework [31]. Кроме того, в процессе оценки характеристик разработанных решений была использована среда Cloudsim [32]. В качестве наземной сети рассматривалась сеть IoT с 1000 оконечными устройствами и двумя шлюзами, распределенными на площади 25 км^2. Воздушная сеть состояла из четырех БПЛА, находящихся на равной высоте 30 м. В таблице 3 представлены рассматриваемые параметры сети и моделирования, а в таблице 4 – минимальное и максимальное положения xy каждого БПЛА. Расположение IoT-шлюзов представлено в таблице 5.

Чтобы оценить характеристики разработанных решений для архитектуры распределенных граничных вычислений «воздух–земля» и оптимизированной модели, были рассмотрены четыре системы, которые определяются следующим образом:

– Система (1): представляет традиционные сети IoT без поддержки G-MEC или A-MEC;

– Система (2): представляет сеть IoT, поддерживаемую только наземной шиной распределенных граничных вычислений; развернуты серверы G-MEC;

– Система (3): представляет собой сеть IoT с поддержкой распределенных G-MEC и A-MEC: развернуты как соответствующие серверы;

– Система (4): представляет собой оптимизированную систему «воздух–земля»; и серверы G-MEC, и серверы A-MEC поддерживают сеть IoT.

ТАБЛИЦА 3. Параметры моделирования

TABLE 3. Simulation Parameters

Параметр	Ценить
Сетевая зона	5 × 5 км²
N	4
W	2
M	5 × 5 км ²
B	4
σ^2	2
γ_0	1000
h_{Di}	20 МГц
t_s	−60 дБм
$V_{\max}$	−30 дБ
V_{Di}	30 м
$E_{Cyc\text{-}IoT}$	45 мс
$\Omega_{Th\text{-}IoT}$	50 м/с
$E_{Cyc\text{-}A\text{-}MEC}$	ϵ (1, 50) м/с
$\Omega_{Th\text{-}A\text{-}MEC}$	1J/ГГц
$\omega_{A\text{-}MEC}$	ϵ [1.0,3.0] ГГц
$\omega_{G\text{-}MEC}$	ϵ [1.5,5.0] ГГц
ω_{IoT}	ϵ [0.5,1.0] ГГц
$\tau_{app\text{-}I}$	7 мс
$\tau_{app\text{-}II}$	10 мс
$\tau_{app\text{-}III}$	16 мс
$\tau_{app\text{-}IV}$	24 мс

ТАБЛИЦА 4. Минимальное и максимальное положение с координатами *xy* для каждого БПЛА

TABLE 4. Minimum and Maximum Position with xy Coordinates for Each UAV

D_i	$X_{\min\text{-}Di}$	$X_{\max\text{-}Di}$	$Y_{\min\text{-}Di}$	$Y_{\max\text{-}Di}$
1	4	5	4	5
2	4	5	0	1
3	0	1	4	5
4	0	1	0	1

ТАБЛИЦА 5. Расположение IoT-шлюзов

TABLE 5. IoT Gateways Location

G_i	X_{Gi}	Y_{Gi}
1	1	2,5
2	4	2,5

Энергопотребление, задержка и доступность для решения вычислительных задач рассматриваются как показатели функционирования систем. Потребление электроэнергии оценивалось для трех систем: традиционной сети IoT, т. е. системы (1), развитой сети IoT «воздух–земля», т. е. системы (3), и оптимизированной сети IoT «воздух–земля», т. е. системы (4).

На рисунке 5 представлено сравнение среднего энергопотребления в процентах для трех систем при разных значениях числа решаемых задач для каждого оконечного устройства.

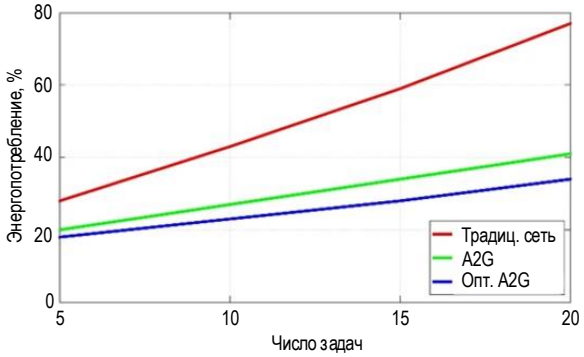

Рис. 5. Среднее энергопотребление при разном числе решаемых задач (в процентах от начального значения доступного энергопотребления)

Fig. 5. Average Power Consumption at Different Number of Solved Tasks (as a Percentage of the Initial Value Available Energy Consumption)

Эти значения были зафиксированы при заданном числе оконечных устройств – 1000, а задачи относились к категории II (обработка неподвижных изображений). Потребление энергии увеличивается по мере увеличения числа вычислительных задач на оконечных устройствах, однако разработанная модель «воздух–земля» сохраняет энергию для большего числа вычислительных задач по сравнению с традиционными сетями IoT. Предлагаемая модель снижает энергопотребление в среднем на 27 % по сравнению с традиционной моделью IoT. Более того, оптимизированная сеть IoT «воздух–земля» позволяет экономить еще больше энергии. При этом достигается более высокая энергоэффективность (в среднем на 6 %) по сравнению с моделью «воздух–земля».

На рисунке 6 показан процент энергопотребления для трех систем в четырех категориях приложений: первая – это облегченные приложения, которые включают простые задачи, например такие, которые необходимы для обработки простых веб-страниц; вторая – приложения на основе неподвижных изображений; третья – простые видеоприложения, которые требуют достаточно простую обработку видео; четвертая категория включает работу с 360-градусными панорамами и обработкой видео, например, дополненной реальности.

При переходе из первой категории в четвертую задачи требуют все более высоких ресурсных возможностей, что снижает вероятность локального выполнения. Очевидно, что энергопотребление для решения задач более высоких категорий выше, чем у более низких категорий.

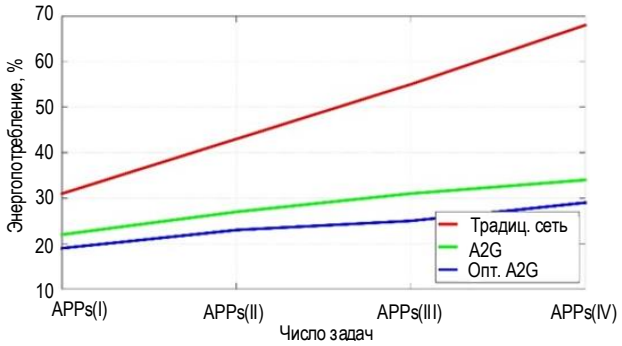

Рис. 6. Среднее энергопотребление для различных типов приложений

Fig. 6. Average Power Consumption for Various Types of Applications

При этом число развернутых устройств составляло 1000 устройств, а число задач на каждом устройстве – 10. Как видим, разработанная система «воздух–земля» снижает энергопотребление в среднем на 29 % по сравнению с традиционными сетями, в то время как оптимизированная модель позволяет снизить потребление энергии на 9 %. На рисунке 6 можно заметить, что разработанная система и оптимизированная система позволяют снизить энергопотребление на более высокий процент для приложений с более высокой рабочей нагрузкой, например, третьей и четвертой категорий.

Влияние плотности расположения оконечных устройств на энергетические характеристики было также проанализировано для трех упомянутых систем. На рисунке 7 показан процент энергопотребления трех систем с разным числом оконечных устройств.

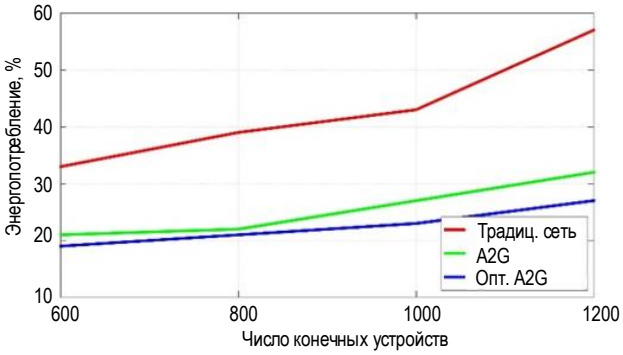

Рис. 7. Среднее энергопотребление при различном количестве развернутых узлов

Fig. 7. Average Power Consumption at Various Number of Nodes Deployed

Количество оконченных устройств в сети начиналось с 600 и было увеличено до 1200. Рассматриваемые задачи относились ко второй категории, а число задач на каждом оконечном устройстве равнялось десяти. Разработанная модель «воздух–земля» снизила потребление энергии в среднем на 19 % по сравнению с традиционной сетью, а оптимизированная модель обеспечила снижение еще на 4 %. Для сети с большим числом развернутых оконечных устройств разработанная модель позволяет снизить потребление энергии на 27 % по сравнению с традиционной моделью. Это связано с высокой нагрузкой на наземное исполнение при плотном развертывании традиционной модели. Отметим, что разработанная схема выгрузки, естественно, снижает общее потребление энергии.

Задержка – важнейший показатель сети, анализируемый в процессе оценки ее функционирования. Средняя задержка обработки рассмотренных вычислительных задач была измерена для ранее упомянутых четырех систем. При этом изменялось число развернутых устройств, требуемых к решению вычислительных задач и категорий приложений.

На рисунке 8 представлена средняя задержка, необходимая для обработки вычислительных задач в каждой из четырех рассматриваемых систем с изменением числа задач, требуемых для выполнения каждым оконечным устройством. Эти показатели были определены для 1000 таких устройств и задач приложений второй категории. Разработанная система «воздух–земля», т. е. система 3, обеспечивает более высокую эффективность системы по задержкам, чем традиционные системы без граничных вычислений и системы только с G-MEC. Это повышение эффективности возрастает с увеличением числа вычислительных задач на оконечных устройствах. Внедрение серверов A-MEC обеспечивает выгрузку вычислительных задач и, таким образом, сокращает общее время, необходимое для обработки этих задач. Более того, разработанная оптимизированная модель обеспечивает дополнительное снижение задержки в среднем на 2,5 %.

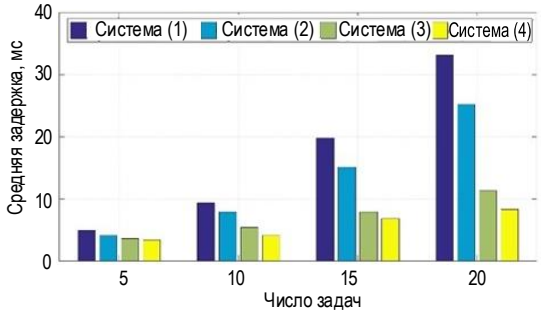

Рис. 8. Средняя задержка для четырех систем с разным числом оконечных устройств

Fig. 8. Average Delay for Four Systems with Different Number of End Devices

На рисунке 9 показана средняя задержка обработки задач в каждой системе из четырех рассматриваемых систем для четырех категорий приложений. Эти значения были получены для сети с 1000

оконечных устройств и десятью задачами на каждом из них. Разработанные система «воздух–земля» и оптимизированная система снижают среднюю задержку, необходимую для обработки вычислительных задач, особенно четвертой категории.

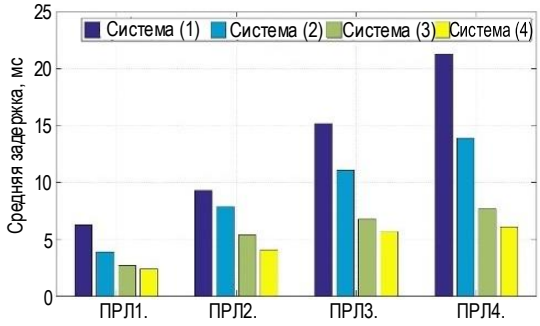

Рис. 9. Средняя задержка при различном числе вычислительных задач

Fig. 9. Average Delay at Different Number Computing Tasks

Кроме того, была измерена средняя задержка для четырех систем с разным числом оконечных устройств, результаты представлены на рисунке 10. Эти показатели были оценены для случая, когда на каждом устройстве было десять задач второй категории. Как показывают результаты, с увеличением числа оконечных устройств традиционные модели не могут поддерживать требуемую задержку, в то время как разработанная система и оптимизированная система достигают существенно более высокой эффективности для обеспечения значений задержки, главным образом в сценариях плотного развертывания.

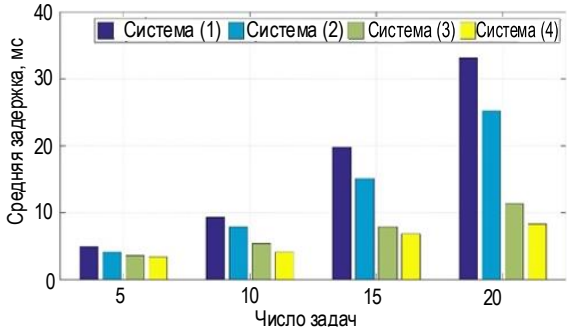

Рис. 10. Средняя задержка для четырех систем с разным числом вычислительных задач для оконечных устройств

Fig. 10. Average Latency for Four Systems with Different Number of Computing Tasks for End Devices

Третьим исследуемым показателем является процент заблокированных задач, который можно использовать в качестве меры доступности системы. Число заблокированных задач фиксировалось для каждой системы в разных случаях. На рисунке 11 представлен процент заблокированных задач для четырех упомянутых систем при различных значениях вычислительных задач для оконечных устройств. С увеличением числа задач процент заблокированных, естественно, увеличился во всех четырех системах, что связано с нехваткой вычислительных ресурсов. Однако процент заблокированных задач для разработанной системы «воздух–земля» меньше, чем у традиционных и наземных систем. Это связано с дополнительными ресурсами, предоставляемыми со стороны воздушного сегмента сети.

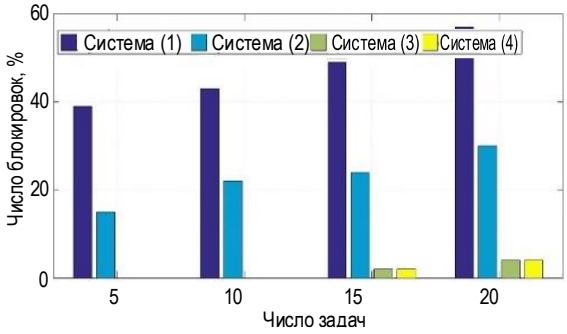

Рис. 11. Процентное соотношение заблокированных задач при различных значениях вычислительных задач для оконечных устройств

Fig. 11. Percentage of Blocked Tasks for Different Values of Computational Tasks for End Devices Tasks

На рисунке 12 представлены результаты оценки процента заблокированных задач для четырех систем по четырем категориям приложений. Как показывают результаты, процент заблокированных задач увеличивается для третьей и четвертой категорий приложений; однако это увеличение минимально для разработанной системы «воздух–земля» и оптимизированной системы. Это связано с тем, что для этих категорий приложений требуются более высокие вычислительные ресурсы, что уменьшает вероятность локального выполнения. Предложенные решения обеспечивает дополнительную возможность с использованием ресурсов воздушного сегмента сети для выгрузки данных, снижая вероятность блокировки задачи.

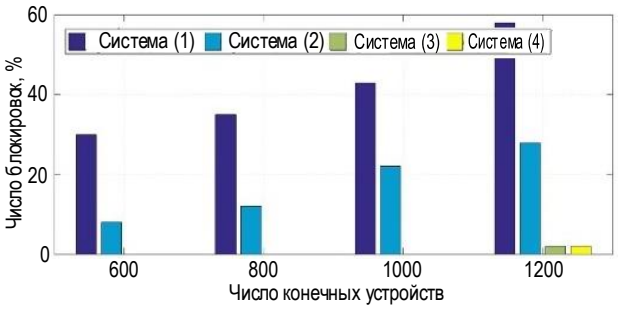

Рис. 12. Процентное соотношение заблокированных задач в зависимости от категории приложений

Fig. 12. Percentage of Blocked Tasks Depending on the Application Category

На рисунке 13 представлены результаты по заблокированным задачам в процентном соотношении для четырех систем при разном количестве оконечных устройств. По мере роста числа таких устройств увеличивается вероятность блокировки

из-за высокой нагрузки на доступные ресурсы. Однако предлагаемые решения, как уже отмечалось выше, обеспечивают дополнительные ресурсы через воздушный сегмент сети.

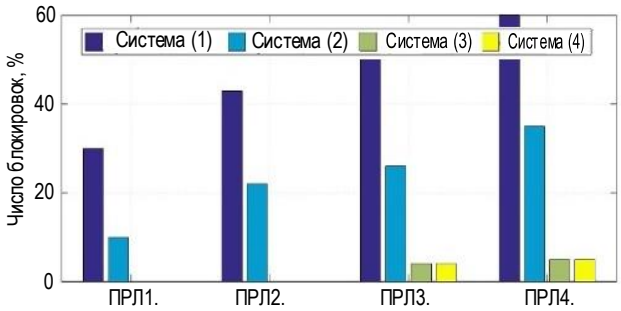

Рис. 13. Процентное соотношение заблокированных задач при разном количестве оконечных устройств

Fig. 13. Percentage of Blocked Tasks for Different Number of End Devices

На рисунке 14 показаны результаты моделирования для четырех систем при различной плотности устройств. Предложенные модель и метод показали более высокую эффективность для значений задержки во всем диапазоне плотности, особенно при высокой плотности наземной сети. Для сетей со сверхвысокой плотностью традиционные модели IoT не могут достичь сверхнизкой или даже низкой задержки. Однако предложенные модель и метод могут поддерживать приложения с ультрамалой задержкой даже при сверхвысокой плотности наземной сети.

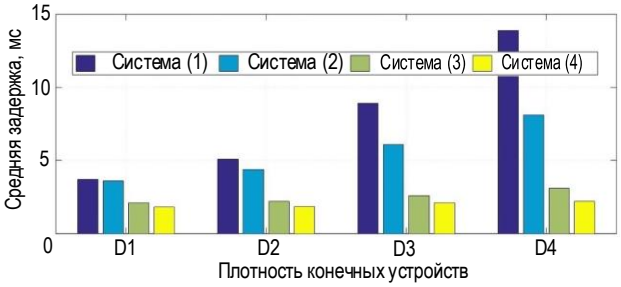

Рис. 14. Средняя задержка для различной плотности наземной сети

Fig. 14. Average Delay for Different Density Ground Network

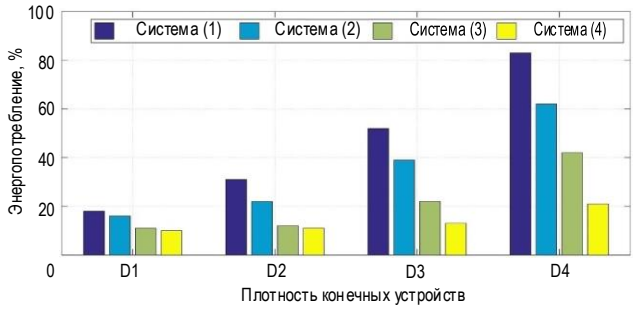

Рис. 15. Процент потребление энергии для различной плотности наземной сети

Fig. 15. Percentage of Energy Consumption for Different Terrestrial Network Density

Заключение

Решена научная проблема, отличающаяся от известных тем, что предложены модель и метод интеграции граничных вычислений в структуру сети «воздух–земля» для сетей Интернета Вещей высокой и сверхвысокой плотности, основанные на интегральном решении задач по размещению граничных серверов на БПЛА и оптимизации структуры сети с использованием метаэвристического алгоритма «роя сальп».

Для решения научной задачи интеграции граничных вычислений в структуру сети «воздух–земля» для сетей Интернета Вещей высокой и сверхвысокой плотности разработана модель сети, отличающаяся от известных тем, что для уменьшения задержки и энергопотребления в такой сети предложено использовать MEC на БПЛА, а связь между наземным и летающим сегментом обеспечивать с использованием NOMA.

Для решения задачи минимизации задержки и энергопотребления в разработанной модели предложен метод выгрузки трафика с наземной сети на MEC на БПЛА, отличающийся от известных тем, что процедура выгрузки трафика является трехуровневой, причем на оконечных устройствах используется программный профилировщик, который определяет сложность вычисляемой задачи и по результатам его работы механизм принятия решения определяет необходимость выгрузки трафика. Кроме того, на втором уровне процедуры сервер БПЛА, на который выгружается трафик, может в условиях недостаточного объема ресурсов принять решение выгрузить трафик на сервер другого БПЛА.

Для решения научной задачи оптимизации структуры сети «воздух–земля» для сетей Интернета Вещей высокой и сверхвысокой плотности, в отличие от известных решений, для минимизации задержки и энергопотребления при выгрузке трафика с наземной сети на серверы граничных вычислений БПЛА был разработан метаэвристический алгоритм на основе хаотического «роя сальп».

Результаты моделирования доказали, что разработанные модель и метод интеграции граничных вычислений в структуру сети «воздух–земля» для сетей Интернета Вещей высокой и сверхвысокой плотности обеспечивают уменьшение задержки до более, чем в 2 раза по сравнению с сетью без использования технологий граничных вычислений и на 30–40 % по сравнению с сетью с использованием только наземных граничных вычислений. Кроме того, использование оптимизации на основе метаэвристического хаотического «роя сальп» дает дополнительный выигрыш около 10 % по сравнению с использованием неоптимизированного алгоритма.

Таким образом достигается уменьшение энергопотребления до более, чем в 2 раза по сравнению с сетью без использования технологий граничных

вычислений. Кроме того, использование оптимизации на основе метаэвристического хаотического «роя сальп» дает дополнительный выигрыш в 5–10 % по сравнению с использованием неоптимизированного алгоритма.

Результат работы показали уменьшение доли заблокированных задач по выгрузке трафика в десятки раз по сравнению с сетью без использования технологий граничных вычислений, в разы по сравнению с сетью с использованием только наземных граничных вычислений. Использование оптимизации на основе метаэвристического хаотического «роя сальп» не дает практически значимого эффекта по сравнению с неоптимизированным алгоритмом.

Определены зависимости значений задержки, энергопотребления и доли заблокированных задач по выгрузке трафика.

Список источников

1. Дунайцев Р.А., Бородин А.С., Кучерявый А.Е. Интегрированная сеть космос-воздух-земля-море как основа сетей связи шестого поколения // Электросвязь. 2022. № 10. С. 5–8. DOI: 10.34832/ELSV2022.35.10.001
2. Ateya A.A., Muthanna A., Makolkina M., Koucheryavy A. Study of 5G Services Standardization: Specifications and Requirements // Proceedings of the10th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT, Moscow, Russia, 05–09 November 2018). IEEE, 2018. DOI:10.1109/ICUMT.2018.8631201
3. Guo F., Yu F.R., Zhang H., Li X., Ji H., Leung V.C.M. Enabling Massive IoT Toward 6G: A Comprehensive Survey // IEEE Internet of Things Journal. 2021. Vol. 8. Iss. 15. PP. 11891–11915. DOI:10.1109/JIOT.2021.3063686
4. Laghari A.A., Wu K., Laghari R.A., Ali M., Khan A.A. A Review and State of Art of Internet of Things (IoT) // Archives of Computational Methods in Engineering. 2022. Vol. 29. Iss. 3. PP. 1395–1413. DOI:10.1007/s11831-021-09622-6
5. Ateya A.A., Algarni A.D., Hamdi M., Koucheryavy A., Soliman N.F. Enabling Heterogeneous IoT Networks over 5G Networks with Ultra-Dense Deployment—Using MEC/SDN // Electronics. 2021. Vol. 10. Iss. 8. P. 910. DOI:10.3390/electronics10080910
6. Bhuiyan M.N., Rahman M.M., Billah M.M., Saha D. Internet of Things (IoT): A Review of Its Enabling Technologies in Healthcare Applications, Standards Protocols, Security, and Market Opportunities // IEEE Internet of Things Journal. 2021. Vol. 8. Iss. 13. PP. 10474–10498. DOI:10.1109/JIOT.2021.3062630
7. Carvalho G., Cabral B., Pereira V., Bernardino J. Edge computing: current trends, research challenges and future directions // Computing. 2021. Vol. 103. PP. 993–1023. DOI:10.1007/s00607-020-00896-5
8. Haibeh L.A., Yagoub M.C.E., Jarray A. A Survey on Mobile Edge Computing Infrastructure: Design, Resource Management, and Optimization Approaches // IEEE Access. 2022. Vol. 10. PP. 27591–27610. DOI:10.1109/ACCESS.2022.3152787
9. Cruz P., Achir N., Viana A.C. On the Edge of the Deployment: A Survey on Multi-access Edge Computing // ACM Computing Surveys. 2022. Vol. 55. Iss. 5. PP. 1–34. DOI:10.1145/3529758
10. Kong L., Tan J., Huang J., Chen G., Wang S., Jin X., et al. Edge-computing-driven Internet of Things: A survey // ACM Computing Surveys. 2022. Vol. 55. Iss. 8. PP. 1–41. DOI:10.1145/3555308
11. Mohsan S.A.H., Khan M.A., Noor F., Ullah I., Alsharif M.H. Towards the Unmanned Aerial Vehicles (UAVs): A Comprehensive Review // Drones. 2022. Vol. 6. Iss. 6. P. 147. DOI:10.3390/drones6060147
12. Amarasingam N., Salgadoe A.S.A., Powell K., Gonzalez L.F., Natarajan S. A review of UAV platforms, sensors, and applications for monitoring of sugarcane crops // Remote Sensing Applications: Society and Environmentvol. 2022. Vol. 26. P. 100712. DOI:10.1016/j.rsase.2022.100712
13. Liu Y., Dai H.-N., Wang Q., Shukla M.K., Imran M. Unmanned aerial vehicle for internet of everything: Opportunities and challenges // Computer Communications. 2020. Vol. 155. PP. 66–83. DOI:10.1016/j.comcom.2020.03.017
14. Pakrooh R., Bohlooli A. A Survey on Unmanned Aerial Vehicles-Assisted Internet of Things: A Service-Oriented Classification // Wireless Personal Communications. 2021. Vol. 119. Iss. 2. PP. 1541–1575. DOI:10.1007/s11277-021-08294-6
15. Idrissi M., Salami M., Annaz F. A Review of Quadrotor Unmanned Aerial Vehicles: Applications, Architectural Design and Control Algorithms // Journal of Intelligent & Robotic Systems. 2022. Vol. 104. DOI:10.1007/s10846-021-01527-7
16. Labib N.S., Brust M.R., Danoy G., Bouvry P. The Rise of Drones in Internet of Things: A Survey on the Evolution, Prospects and Challenges of Unmanned Aerial Vehicles // IEEE Access/ 2021. Vol. 9. PP. 115466–115487. DOI:10.1109/ACCESS.2021.3104963
17. Siddharthraju K., Dhivyadevi R., Supriya M., Jaishankar B., Shanmugaraja T. A Survey on IoE-Enabled Unmanned Aerial Vehicles // Mohindru V., Singh Y., Bhatt R., Gupta A.K. (Ed.) Unmanned Aerial Vehicles for Internet of Things (IoT). Wiley, 2021. PP. 173–192. DOI:10.1002/9781119769170.ch10
18. Shehzad M.K., Ahmad A., Hassan S.A., Jung H. Backhaul-Aware Intelligent Positioning of UAVs and Association of Terrestrial Base Stations for Fronthaul Connectivity // IEEE Transactions on Network Science and Engineering. 2021. Vol. 8. Iss. 4. PP. 2742–2755. DOI:10.1109/TNSE.2021.3077314
19. Alsamhi S.H., Shvetsov A.V., Kumar S., Hassan J., Alhartomi M.A., Shvetsova S.V., et al. Computing in the Sky: A Survey on Intelligent Ubiquitous Computing for UAV-Assisted 6G Networks and Industry 4.0/5.0 // Drones. 2022. Vol. 6. Iss. 7. P. 177. DOI:10.3390/drones6070177
20. Yazid Y., Ez-Zazi I., Guerrero-González A., El Oualkadi A., Arioua M. UAV-Enabled Mobile Edge-Computing for IoT Based on AI: A Comprehensive Review // Drones. 2021. Vol. 5. Iss. 4. P. 148. DOI:10.3390/drones5040148
21. Zhang S., Liu W., Ansari N. Joint Wireless Charging and Data Collection for UAV-Enabled Internet of Things Network // IEEE Internet of Things Journal. 2022. Vol. 9. Iss. 23. PP. 23852–23859. DOI:10.1109/JIOT.2022.3190813
22. Beniwal G., Singhrova A. A systematic literature review on IoT gateways // Journal of King Saud University – Computer and Information Sciences. 2021. Vol. 34. Iss. 10. PP. 9541–9563. DOI:10.1016/j.jksuci.2021.11.007

23. Jeong S., Simeone O., Kang J. Mobile cloud computing with a UAV-mounted cloudlet: optimal bit allocation for communication and computation // IET Communications. 2017. Vol. 11. Iss. 7. PP. 969–974. DOI:10.1049/iet-com.2016.1114
24. Jeong S., Simeone O., Kang J. Mobile Edge Computing via a UAV-Mounted Cloudlet: Optimization of Bit Allocation and Path Planning // IEEE Transactions on Vehicular Technology. 2018. Vol. 67. Iss. 3. PP. 2049–2063. DOI:10.1109/TVT.2017.2706308
25. Ateya A.A.A., Muthanna A., Kirichek R., Hammoudeh M., Koucheryavy A. Energy- and Latency-Aware Hybrid Offloading Algorithm for UAVs // IEEE Access. 2019. Vol. 7. PP. 37587–37600. DOI:10.1109/ACCESS.2019.2905249
26. Solomon M.G., Kim, D. Fundamentals of communications and networking. Jones & Bartlett Learning, 2021.
27. Ateya A.A.A., Muthanna A., Gudkova I., Gaidamaka Y., Algarni A.D. Latency and energy-efficient multi-hop routing protocol for unmanned aerial vehicle networks // International Journal of Distributed Sensor Networks. 2019. Vol. 15. Iss. 8. DOI:10.1177/1550147719866392
28. Castelli M., Manzoni L., Mariot L., Nobile M.S., Tangherloni A. Salp Swarm Optimization: A critical review // Expert Systems with Applications. 2022. Vol. 189. P. 116029. DOI:10.1016/j.eswa.2021.116029
29. Pradhan A., Bisoy S.K., Das A. A survey on PSO based meta-heuristic scheduling mechanism in cloud computing environment // Journal of King Saud University – Computer and Information Sciences. 2022. Vol. 34. Iss. 8. PP. 4888–4901. DOI:10.1016/j.jksuci.2021.01.003
30. Parthiban S., Harshavardhan A., Neelakandan S., Prashanthi V., Alolo A.-R.A.A., Velmurugan S. Chaotic Salp Swarm Optimization-Based Energy-Aware VMP Technique for Cloud Data Centers // Computational Intelligence and Neuroscience. 2022. Vol. 2022. P. 4343476. DOI:10.1155/2022/4343476
31. Sliwa B., Patchou M., Wietfeld C. Lightweight Simulation of Hybrid Aerial- and Ground-Based Vehicular Communication Networks // Proceedings of the 90th Vehicular Technology Conference (VTC2019-Fall, Honolulu, USA, 22–25 September 2019). IEEE, 2019. DOI:10.1109/VTCFall.2019.8891340
32. Goyal T., Singh A., Agrawal A. Cloudsim: simulator for cloud computing infrastructure and modeling // Procedia Engineering. 2012. Vol. 38. PP. 3566–3572. DOI:10.1016/j.proeng.2012.06.412

References

1. Dunaytsev R.A., Borodin A.S., Koucheryavy A.E. Space-air-ground-sea integrated networking as a basis for 6G networks. *Electrosvyaz*. 2022;10:5–8. (in Russ.) DOI:10.34832/ELSV2022.35.10.001
2. Ateya A.A., Muthanna A., Makolkina M., Koucheryavy A. Study of 5G Services Standardization: Specifications and Requirements. *Proceedings of the10th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops, ICUMT, 05–09 November 2018, Moscow, Russia*. IEEE; 2018. DOI:10.1109/ICUMT.2018.8631201
3. Guo F., Yu F.R., Zhang H., Li X., Ji H., Leung V.C.M. Enabling Massive IoT Toward 6G: A Comprehensive Survey. *IEEE Internet of Things Journal*. 2021;8(15):11891–11915. DOI:10.1109/JIOT.2021.3063686
4. Laghari A.A., Wu K., Laghari R.A., Ali M., Khan A.A. A Review and State of Art of Internet of Things (IoT). *Archives of Computational Methods in Engineering*. 2022;29(3):1395–1413. DOI:10.1007/s11831-021-09622-6
5. Ateya A.A., Algarni A.D., Hamdi M., Koucheryavy A., Soliman N.F. Enabling Heterogeneous IoT Networks over 5G Networks with Ultra-Dense Deployment—Using MEC/SDN. *Electronics*. 2021;10(8):910. DOI:10.3390/electronics10080910
6. Bhuiyan M.N., Rahman M.M., Billah M.M., Saha D. Internet of Things (IoT): A Review of Its Enabling Technologies in Healthcare Applications, Standards Protocols, Security, and Market Opportunities. *IEEE Internet of Things Journal*. 2021;8(13):10474–10498. DOI:10.1109/JIOT.2021.3062630
7. Carvalho G., Cabral B., Pereira V., Bernardino J. Edge computing: current trends, research challenges and future directions. *Computing*. 2021;103:993–1023. DOI:10.1007/s00607-020-00896-5
8. Haibeh L.A., Yagoub M.C.E., Jarray A. A Survey on Mobile Edge Computing Infrastructure: Design, Resource Management, and Optimization Approaches. *IEEE Access*. 2022;10:27591–27610. DOI:10.1109/ACCESS.2022.3152787
9. Cruz P., Achir N., Viana A.C. On the Edge of the Deployment: A Survey on Multi-access Edge Computing. *ACM Computing Surveys.* 2022;55(5):1–34. DOI:10.1145/3529758
10. Kong L., Tan J., Huang J., Chen G., Wang S., Jin X., et al. Edge-computing-driven Internet of Things: A survey. *ACM Computing Surveys*. 2022;55(8):1–41. DOI:10.1145/3555308
11. Mohsan S.A.H., Khan M.A., Noor F., Ullah I., Alsharif M.H. Towards the Unmanned Aerial Vehicles (UAVs): A Comprehensive Review. *Drones*. 2022;6(6):147. DOI:10.3390/drones6060147
12. Amarasingam N., Salgadoe A.S.A., Powell K., Gonzalez L.F., Natarajan S. A review of UAV platforms, sensors, and applications for monitoring of sugarcane crops. *Remote Sensing Applications: Society and Environmentvol*. 2022;26:100712. DOI:10.1016/j.rsase.2022.100712
13. Liu Y., Dai H.-N., Wang Q., Shukla M.K., Imran M. Unmanned aerial vehicle for internet of everything: Opportunities and challenges. *Computer Communications*. 2020;155:66–83. DOI:10.1016/j.comcom.2020.03.017
14. Pakrooh R., Bohlooli A. A Survey on Unmanned Aerial Vehicles-Assisted Internet of Things: A Service-Oriented Classification. *Wireless Personal Communications*. 2021;119(2):1541–1575. DOI:10.1007/s11277-021-08294-6
15. Idrissi M., Salami M., Annaz F. A Review of Quadrotor Unmanned Aerial Vehicles: Applications, Architectural Design and Control Algorithms. *Journal of Intelligent & Robotic Systems.* 2022;104. DOI:10.1007/s10846-021-01527-7
16. Labib N.S., Brust M.R., Danoy G., Bouvry P. The Rise of Drones in Internet of Things: A Survey on the Evolution, Prospects and Challenges of Unmanned Aerial Vehicles. *IEEE Access.* 2021;9:115466–115487. DOI:10.1109/ACCESS. 2021.3104963
17. Siddharthraju K., Dhivyadevi R., Supriya M., Jaishankar B., Shanmugaraja T. A Survey on IoE-Enabled Unmanned Aerial Vehicles. *In: Mohindru V., Singh Y., Bhatt R., Gupta A.K. (Ed.) Unmanned Aerial Vehicles for Internet of Things (IoT)*. Wiley; 2021. p.173–192. DOI:10.1002/9781119769170.ch10

18. Shehzad M.K., Ahmad A., Hassan S.A., Jung H. Backhaul-Aware Intelligent Positioning of UAVs and Association of Terrestrial Base Stations for Fronthaul Connectivity. *IEEE Transactions on Network Science and Engineering*. 2021;8(4):2742–2755. DOI:10.1109/TNSE.2021.3077314
19. Alsamhi S.H., Shvetsov A.V., Kumar S., Hassan J., Alhartomi M.A., Shvetsova S.V., et al. Computing in the Sky: A Survey on Intelligent Ubiquitous Computing for UAV-Assisted 6G Networks and Industry 4.0/5.0. *Drones*. 2022;6(7):177. DOI:10.3390/drones6070177
20. Yazid Y., Ez-Zazi I., Guerrero-González A., El Oualkadi A., Arioua M. UAV-Enabled Mobile Edge-Computing for IoT Based on AI: A Comprehensive Review. *Drones*. 2021;5(4):148. DOI:10.3390/drones5040148
21. Zhang S., Liu W., Ansari N. Joint Wireless Charging and Data Collection for UAV-Enabled Internet of Things Network. *IEEE Internet of Things Journal*. 2022;9(23):23852–23859. DOI:10.1109/JIOT.2022.3190813
22. Beniwal G., Singhrova A. A systematic literature review on IoT gateways. *Journal of King Saud University – Computer and Information Sciences*. 2021;34(10):9541–9563. DOI:10.1016/j.jksuci.2021.11.007
23. Jeong S., Simeone O., Kang J. Mobile cloud computing with a UAV-mounted cloudlet: optimal bit allocation for communication and computation. *IET Communications*. 2017;11(7):969–974. DOI:10.1049/iet-com.2016.1114
24. Jeong S., Simeone O., Kang J. Mobile Edge Computing via a UAV-Mounted Cloudlet: Optimization of Bit Allocation and Path Planning. *IEEE Transactions on Vehicular Technology*. 2018;67(3):2049–2063. DOI:10.1109/TVT.2017.2706308
25. Ateya A.A.A., Muthanna A., Kirichek R., Hammoudeh M., Koucheryavy A. Energy- and Latency-Aware Hybrid Offloading Algorithm for UAVs. *IEEE Access*. 2019;7:37587–37600. DOI:10.1109/ACCESS.2019.2905249
26. Solomon M.G., Kim D. *Fundamentals of communications and networking*. Jones & Bartlett Learning; 2021.
27. Ateya A.A.A., Muthanna A., Gudkova I., Gaidamaka Y., Algarni A.D. Latency and energy-efficient multi-hop routing protocol for unmanned aerial vehicle networks. *International Journal of Distributed Sensor Networks.* 2019;15(8). DOI:10.1177/1550147719866392
28. Castelli M., Manzoni L., Mariot L., Nobile M.S., Tangherloni A. Salp Swarm Optimization: A critical review. *Expert Systems with Applications*. 2022;189:116029. DOI:10.1016/j.eswa.2021.116029
29. Pradhan A., Bisoy S.K., Das A. A survey on PSO based meta-heuristic scheduling mechanism in cloud computing environment. *Journal of King Saud University – Computer and Information Sciences*. 2022;34(8):4888–4901. DOI:10.1016/j.jksuci.2021.01.003
30. Parthiban S., Harshavardhan A., Neelakandan S., Prashanthi V., Alolo A.-R.A.A., Velmurugan S. Chaotic Salp Swarm Optimization-Based Energy-Aware VMP Technique for Cloud Data Centers. *Computational Intelligence and Neuroscience*. 2022;2022:4343476. DOI:10.1155/2022/4343476
31. Sliwa B., Patchou M., Wietfeld C. Lightweight Simulation of Hybrid Aerial- and Ground-Based Vehicular Communication Networks. *Proceedings of the 90th Vehicular Technology Conference, VTC2019-Fall, 22–25 September 2019, Honolulu, USA*. IEEE; 2019. DOI:10.1109/VTCFall.2019.8891340
32. Goyal T., Singh A., Agrawal A. Cloudsim: simulator for cloud computing infrastructure and modeling. *Procedia Engineering.* 2012;38:3566–3572. DOI:10.1016/j.proeng.2012.06.412

Статья поступила в редакцию 09.06.2023; одобрена после рецензирования 25.06.2023; принята к публикации 05.07.2023.
The article was submitted 09.06.2023; approved after reviewing 25.06.2023; accepted for publication 05.07.2023.

Информация об авторе:

МУТХАННА Аммар Салех Али

кандидат технических наук, доцент кафедры сети связи и передачи данных Санкт-Петербургского государственного университета телекоммуникаций им. проф. М.А. Бонч-Бруевича,
https://orcid.org/0000-0003-0213-8145