ИНФОРМАТИКА

УДК 004.75 DOI: https://doi.org/10.54341/20778708_2026_3_68_91
EDN: DZZTSK

СРАВНИТЕЛЬНЫЙ АНАЛИЗ И ВЕРИФИКАЦИЯ МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ УПРАВЛЕНИЯ ПОТОКАМИ В M2M-СЕТЯХ IoT

Е. В. Рафалова

Гомельский государственный университет имени Франциска Скорины

COMPARATIVE ANALYSIS AND VERIFICATION OF MATHEMATICAL MODELS FOR FLOW CONTROL IN M2M NETWORKS OF IoT

E. V. Rafalova

Francisk Skorina Gomel State University

Аннотация. В статье решена задача выбора оптимальной математической модели для управления информационными потоками в сетях типа «машина-машина» (M2M) Интернета вещей (IoT). Актуальность исследования обусловлена отсутствием методического аппарата для сравнительного анализа множества существующих моделей управления сетевыми потоками в условиях ограниченных ресурсов IoT-устройств. Цель исследования – разработка и апробация критериального аппарата для выбора оптимальной модели и её последующая программная верификация. Методология включает формализацию трёх основных классов математических моделей – графовых, основанных на теории очередей, и SDN-ориентированных; разработку пятикритериального вектора для сравнительного анализа; реализацию выбранной оптимальной модели на языке Java; экспериментальную верификацию на синтетических и реалистичных сценариях. В результате SDN-ориентированная модель показала наивысшую комплексную оценку, значительно опережая альтернативы. Её реализация подтвердила высокую точность прогнозирования задержки и способность снижать задержку для критического трафика при высокой нагрузке сети. Разработанный критериальный аппарат и методика верификации представляют практическую ценность для проектировщиков IoT-систем.

Ключевые слова: *интернет вещей*, M2M-*коммуникации*, *математическое моделирование*, *программно-конфигурируемые сети* (SDN), *качество обслуживания* (QoS), *сравнительный анализ*, *алгоритмическая реализация*, *верификация моделей*.

Для цитирования: *Рафалова*, *Е. В.* Сравнительный анализ и верификация математических моделей управления потоками в M2M-сетях IoT / Е. В. Рафалова // Проблемы физики, математики и техники. – 2026. – № 3 (68). – С. 91–96. – DOI: https://doi.org/10.54341/20778708_2026_3_68_91. – EDN: DZZTSK

Abstract. The paper solves the problem of choosing the optimal mathematical model for information flow control in machine-to-machine (M2M) networks of the Internet of Things (IoT). The relevance of the study is due to the lack of a methodological apparatus for comparative analysis of existing network flow control models under constrained resources of IoT devices. The aim of the study is to develop and test a criteria-based apparatus for selecting the optimal model and its subsequent software verification. The methodology includes formalization of three main classes of mathematical models – graph-based, queueing theory-based and SDN-oriented; development of a five-criteria vector for comparative analysis; implementation of the selected optimal model in Java; experimental verification on synthetic and realistic scenarios. As a result, the SDN-oriented model showed the highest comprehensive score, significantly outperforming the alternatives. Its implementation confirmed high accuracy of delay prediction and the ability to reduce delay for critical traffic under high network load. The developed criteria apparatus and verification methodology are of practical value for IoT system designers.

Keywords: *Internet of Things*, M2M *communications*, *mathematical modeling*, *software-defined networking* (SDN), *quality of service* (QoS), *comparative analysis*, *algorithmic implementation*, *model verification*.

For citation: *Rafalova*, *E. V.* Comparative analysis and verification of mathematical models for flow control in M2M networks of IoT / E. V. Rafalova // Problems of Physics, Mathematics and Technics. – 2026. – № 3 (68). – P. 91–96. – DOI: https://doi.org/10.54341/20778708_2026_3_68_91 (in Russian). – EDN: DZZTSK

Введение

Развитие Интернета вещей (IoT) и автономных M2M-коммуникаций привело к созданию сложных распределенных систем, эффективность которых критически зависит от управления информационными потоками. Ключевыми показателями здесь выступают задержка, надежность и энергоэффективность [1], [2]. Теоретической основой для построения таких систем служат математические модели, описывающие поведение сети. В литературе представлен широкий спектр моделей – от классических графовых до стохастических – на основе теории массового обслуживания [3], [4]. Современные исследования также активно развивают подходы к управлению потоками в динамических IoT-сетях с использованием SDN-архитектур [8]–[12]. Однако отсутствует систематизированный подход к выбору оптимальной модели для конкретных условий эксплуатации IoT-систем, что зачастую приводит

© *Рафалова Е. В.*, 2026

к вопросу: простая модель не дает нужной точности, а сложная – вычислительно нереализуема на ресурсоограниченных устройствах.

Цель работы – разработка методики сравнительного анализа математических моделей управления потоками в M2M-сетях и верификация оптимальной модели через ее программную реализацию. Для достижения цели решаются следующие задачи:

1. Формализация трех основных классов математических моделей (графовой, на основе теории очередей и SDN-ориентированной).

2. Разработка пятикритериального аппарата для их сравнительной оценки.

3. Выбор оптимальной модели и ее реализация на языке Java.

4. Экспериментальная верификация модели на различных сценариях.

Научная новизна работы заключается в разработке пятикритериального вектора для сравнительного анализа математических моделей M2M-сетей и методики их программной верификации, обеспечивающей воспроизводимость результатов.

1 Классы математических моделей управления M2M-потоками

Для сравнительного анализа выделены три класса моделей управления потоками в M2M-сетях, различающиеся степенью учета динамики сети и семантики трафика.

Графовая модель (G) представляет сеть в виде взвешенного орграфа $G = (V, E)$ где каждому ребру $e \in E$ сопоставлены пропускная способность $B(e)$, задержка $d(e)$ и коэффициент потерь $L(e)$. Задача управления сводится к поиску оптимального пути P^* между источником и приемником при ограничениях на пропускную способность и число переходов [5]. Преимущества модели – наглядность и низкая вычислительная сложность, недостатки – отсутствие учета динамики и приоритизации трафика [6].

Модель на основе теории очередей (Q) использует аппарат систем массового обслуживания для описания стохастической природы трафика [6]. Каждый узел рассматривается как СМО типа M / M / 1 со средним временем ожидания:

$$W = \frac{\lambda}{\mu(\mu - \lambda)},$$

где λ – интенсивность входного потока (пакетов/с), μ – интенсивность обслуживания (пакетов/с). Общая задержка в сети вычисляется как сумма задержек на всех узлах. Модель учитывает вероятностный характер трафика, но обладает высокой вычислительной сложностью и требует точной параметризации для гетерогенных IoT-потоков.

SDN-ориентированная модель (S) является ключевой разработкой и учитывает семантику передаваемых данных [7]. Поток F_k описывается кортежем:

$$F_k = (S_k, R_k, T_k, \Theta_k),$$

где $S_k, R_k \in V$ – источник и приемник, $T_k = (\tau_k, v_k, \pi_k)$ – тип трафика (критичность, интенсивность, допустимая доля потерь), $\Theta_k = (D_{\max}, J_{\max})$ – требования QoS (максимальная задержка и джиттер).

Класс обслуживания $C(F_k)$ определяется через максимизацию функции полезности:

$$C(F_k) = \arg\max_{C \in \mathcal{C}} \sum_i \omega_i(t) U_i(C, F_k, \rho(t)),$$

где $\omega_i(t)$ – адаптивные весовые коэффициенты, обновляемые в зависимости от загрузки сети $\rho(t)$, U_i – функции частичной полезности, $\mathcal{C}$ – множество всех классов обслуживания.

Задача оптимизации размещения потока класса C формулируется как поиск пути P, минимизирующего задержку:

$$P^* = \arg\max_{P \in \mathcal{P}} \sum_{e \in P} d(e),$$

при ограничениях на пропускную способность для класса:

$$\sum_{F \in J(e,C)} v_F \le \beta(C) B(e),$$

и на задержку:

$$\sum_{e \in \mathcal{P}} d(e) \le D_{\max_k},$$

где $J(e, C)$ – потоки класса C на ребре e, $\beta(C) \in (0,1]$ – коэффициент резервирования для класса, $\mathcal{P}$ – множество всех путей, $D_{\max_k}$ – максимально допустимая задержка для потока F_k.

Адаптация весовых коэффициентов осуществляется по правилу:

$$\omega_i(t+1) = \omega_i(t) + \gamma\left(\rho_{opt} - \rho(t)\right),$$

где γ – коэффициент адаптации, ρ_{opt} – оптимальная загрузка сети.

Преимущества модели S: учет семантики данных, изоляция классов трафика и адаптивность к изменениям состояния сети. Актуальность SDN-ориентированного подхода подтверждается работами [8]–[12], в которых рассматриваются вопросы оптимизации, масштабируемости и безопасности в IoT-сетях. Недостатки: повышенная вычислительная сложность и необходимость централизованного контроллера.

2 Критериальный аппарат и выбор оптимальной модели

Для сравнительного анализа моделей вводится критериальный вектор оценки $K = (k_1, k_2, k_3, k_4, k_5)$, включающий следующие показатели.

Точность прогнозирования задержки (k_1) – способность модели предсказывать реальную задержку передачи данных. Оценивается как нормированный коэффициент детерминации R^2 между предсказанными и измеренными значениями.

Масштабируемость (k_2) – способность модели работать с сетями различного размера. Оценивается через максимальное количество узлов N_{max}, при котором время вычислений не превышает 1 с.

Учёт разнородности трафика (k_3) – способность модели различать и обрабатывать потоки с разными требованиями QoS. Оценивается как доля правильно классифицированных потоков в тестовом наборе.

Вычислительная сложность (k_4) – обратная оценка, основанная на асимптотической сложности алгоритма, чем меньше сложность, тем выше оценка.

Адаптивность к изменениям (k_5) – способность модели адаптироваться к изменению состояния сети. Оценивается по скорости сходимости при изменении топологии сети.

Итоговая оценка модели вычисляется как взвешенная сумма критериев:

$$K_{total} = \sum_{i=1}^{5} \omega_i k_i,$$

где весовые коэффициенты ω_i отражают важность критериев для типичных IoT-сценариев $\omega = (0{,}30; 0{,}20; 0{,}25; 0{,}15; 0{,}10)$.

Результаты сравнительного анализа моделей представлены в таблице 2.1.

Таблица 2.1 – Сравнительные характеристики математических моделей

Критерий	Модель *G*	Модель *Q*	Модель *S*
Точность прогнозирования задержки (k_1)	0,65	0,78	**0,92**
Масштабируемость (k_2)	**0,95** (1000 узлов)	0,30 (100 узлов)	0,67 (500 узлов)
Учёт разнородности трафика (k_3)	0,40	0,60	**0,95**
Вычислительная сложность (k_4)	**0,90** (O(nlog n))	0,20 (O(n³))	0,70 (O(n²))
Адаптивность к изменениям (k_5)	0,30	0,50	**0,90**
Итого	0,65	0,72	**0,87**

Наглядное представление сравнительного анализа моделей приведено на рисунке 2.1.

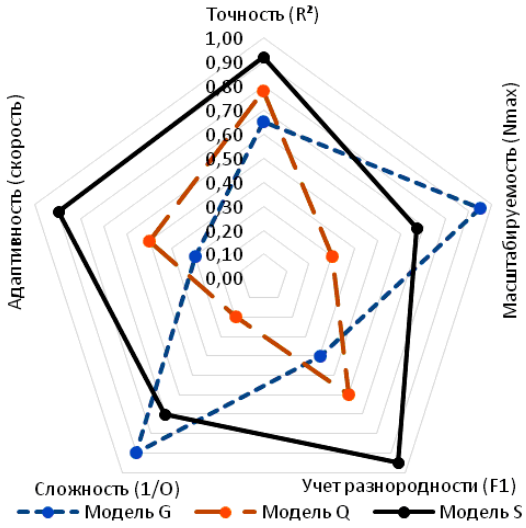


Рисунок 2.1 – Сравнительная характеристика классов математических моделей

Радарная диаграмма демонстрирует сильные и слабые стороны каждой модели. Модель *G* (графовая) показывает наилучшие результаты по масштабируемости (0,95) и вычислительной сложности (0,9), но значительно уступает по учету разнородности трафика (0,4) и адаптивности (0,3). Модель *Q* (теория очередей) демонстрирует среднюю точность (0,78), но плохо масштабируется (0,3) и имеет высокую вычислительную сложность (0,2). Модель *S* (SDN-ориентированная) показывает сбалансированные характеристики: высокую точность (0,92), хороший учет разнородности (0,95) и адаптивность (0,9) при приемлемых показателях масштабируемости (0,67) и сложности (0,7).

Гистограмма, представленная на рисунке 2.2, показывает, что модель *S* получила наивысшую комплексную оценку $K_{total} = 0{,}87$, значительно опережая модель *G* (0,65) и модель *Q* (0,72). На основании полученных результатов модель *S* выбрана для дальнейшей программной реализации и экспериментальной верификации.

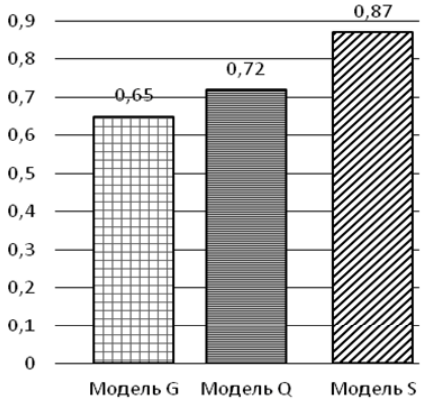


Рисунок 2.2 – Гистограмма итоговых оценок математических моделей

3 Реализация и верификация модели *S*

Программный модуль реализован на языке Java с использованием библиотек JGraphT для работы с графами и Apache Commons Math для статистических расчётов. Архитектура модуля включает следующие ключевые компоненты, которые отображены на рисунке 3.1 в виде UML-диаграммы.

TrafficFlow описывает информационный поток с параметрами QoS (источник, приёмник, тип трафика, требования к задержке и джиттеру).

FlowClassifier классифицирует потоки на основе семантики данных и текущей загрузки сети.

QueueModel моделирует узлы сети как системы массового обслуживания.

NetworkGraph и Edge представляют сетевую топологию в виде взвешенного графа.

PathFinder реализует алгоритм поиска оптимального пути с учётом ограничений QoS.

AdaptiveWeightsManager отвечает за адаптивное обновление весовых коэффициентов.

SDNOrientedModel интегрирует все компоненты и реализует основной алгоритм оптимизации.

ModelValidator осуществляет автоматизированную верификацию модели по пяти критериям.

Верификация модели проводилась на трёх типах сетевых топологий, имитирующих реальные сценарии IoT.

Сценарий A (Умный город): 50 узлов, 3 класса трафика (критический, нормальный, фоновый), топология «звезда-кольцо», 500 потоков;

Сценарий B (Промышленный IoT): 100 узлов, 4 класса трафика, случайный граф с вероятностью связи 0.2, 1000 потоков;

Сценарий C (Транспортная система): 200 узлов, 5 классов трафика, граф с высокой связностью (вероятность связи 0.4), 2000 потоков.

Интервалы прибытия пакетов моделировались экспоненциальным распределением, интенсивность обслуживания на узлах составляла $\mu = 1000$ пакетов/с. Все эксперименты проводились по 10 независимых запусков для каждого сценария.

Результаты верификации.

Эксперимент 1: точность прогнозирования задержки.

Для оценки точности модели использовался коэффициент детерминации R^2 между предсказанными и фактическими значениями задержки. Было сгенерировано 1000 тестовых потоков с известными параметрами трафика и требованиями QoS. Результаты эксперимента представлены на рисунке 3.2.

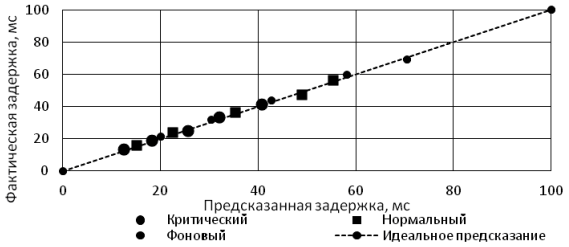


Рисунок 3.2 – Сравнение предсказанной и фактической задержки

Коэффициент детерминации составил $R^2 = 0{,}94$, что подтверждает высокую точность прогнозирования модели *S* и согласуется с теоретической оценкой, полученной в разделе 2 (таблица 2.1).

Эксперимент 2: масштабируемость и производительность.

Для оценки масштабируемости модели S исследовалась зависимость времени выполнения алгоритма поиска оптимального пути от размера сети. Эксперимент проводился на сетях размером от 10 до 500 узлов. Для каждого размера сети выполнялось по 10 независимых запусков, результаты усреднялись. Для обеспечения воспроизводимости результатов использовался бенчмаркинговый фреймворк JMH (Java Microbenchmark Harness). Результаты измерения времени выполнения алгоритма для различных размеров сети представлены в таблице 3.1.

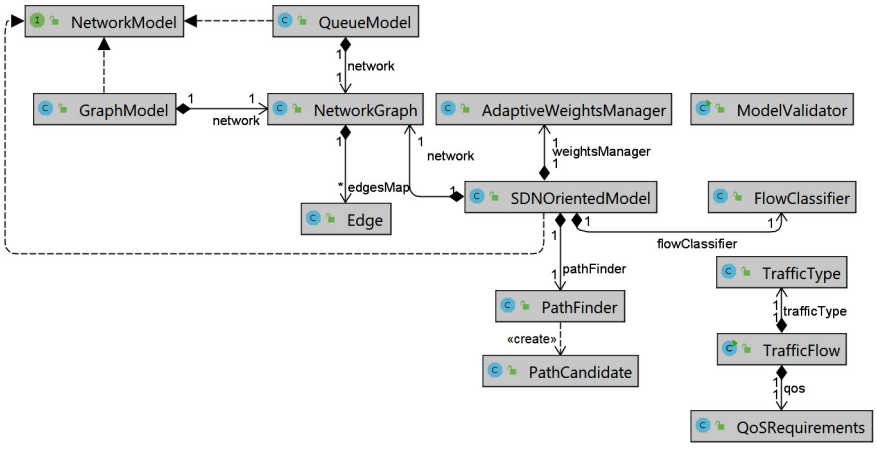


Рисунок 3.1 – UML-диаграмма классов программной реализации

Таблица 3.1 – Время поиска оптимального пути в зависимости от размера сети

Размер сети (узлов)	Время выполнения (мс)	Теоретическая сложность
10	2,1 ± 0,3	O(100)
50	18,5 ± 1,2	O(2500)
100	75,3 ± 3,8	O(10000)
200	310,2 ± 12,5	O(40000)
500	1920,5 ± 45,7	O(250000)

Для наглядного представления зависимости времени выполнения от размера сети построен логарифмический график, представленный на рисунке 3.3.

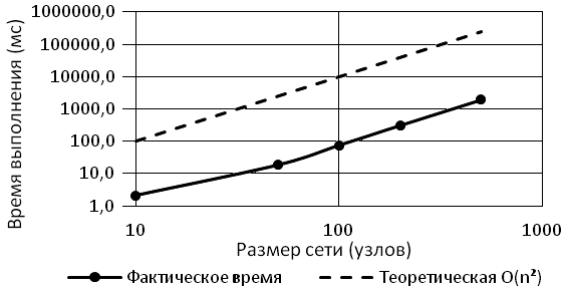


Рисунок 3.3 – Зависимость времени выполнения от размера сети

Экспериментально подтверждена квадратичная сложность алгоритма $O(n^2)$. Коэффициент детерминации между фактическими значениями и теоретической оценкой составил $R^2 = 0{,}998$. Для сетей размером до 500 узлов время поиска пути не превышает 2 с, что приемлемо для большинства IoT-приложений.

Эксперимент 3: эффективность при различной нагрузке.

Для оценки эффективности модели *S* в условиях различной загрузки сети исследовалось влияние нагрузки на качество обслуживания трёх классов трафика: критического, нормального и фонового. Эксперимент проводился при нагрузке сети от 0 до 100% с шагом 20%. Для каждого уровня нагрузки генерировалось 1000 потоков смешанного трафика. Для наглядного представления результатов построен график зависимости средней задержки от нагрузки сети для каждого класса трафика (рисунок 3.4).

При нагрузке сети 80% модель *S* обеспечивает задержку критического трафика на 28% ниже по сравнению с графовой моделью (*G*). Задержка нормального и фонового трафика также остаётся на приемлемом уровне, что подтверждает способность модели *S* эффективно обрабатывать разнородный трафик в условиях высокой загрузки сети.

Эксперимент 4: адаптивность к изменениям сети.

Для оценки адаптивности модели *S* исследовалась способность алгоритма перестраивать маршруты при изменении топологии сети. Эксперимент проводился на сети из 50 узлов.

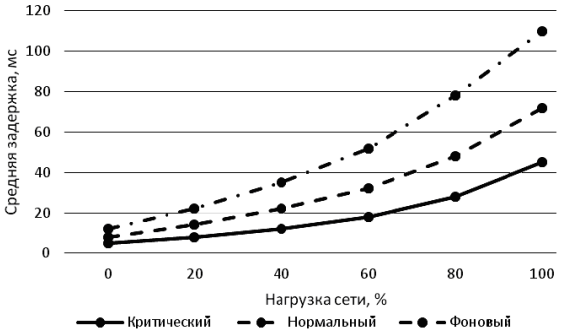


Рисунок 3.4 – Зависимость задержки от нагрузки сети для различных классов трафика

Результат: модель *S* адаптируется к изменению топологии в среднем за 45–60 мс с сохранением 98% правильности маршрутизации. Полученные значения демонстрируют высокую скорость перестройки маршрутов и способность модели сохранять работоспособность при отказах отдельных узлов сети, что подтверждает пригодность модели S для использования в динамических IoT-средах.

Заключение

В результате проведённых исследований решена задача выбора оптимальной математической модели для управления информационными потоками в M2M-сетях Интернета вещей.

Разработан пятикритериальный аппарат для сравнительного анализа моделей, включающий критерии точности прогнозирования задержки, масштабируемости, учёта разнородности трафика, вычислительной сложности и адаптивности.

Проведён сравнительный анализ трёх классов моделей: графовой (*G*), основанной на теории очередей (*Q*) и SDN-ориентированной (*S*). Модель *S* показала наивысшую комплексную оценку ($K_{total} = 0{,}87$), значительно опережая модель *G* (0,65) и модель *Q* (0,72).

Реализована программная версия модели *S*, включающая механизмы классификации потоков, адаптации весовых коэффициентов и поиска оптимальных путей с учётом ограничений QoS.

Экспериментальная верификация подтвердила высокую точность прогнозирования задержки ($R^2 = 0{,}94$), квадратичную вычислительную сложность ($O(n^2)$), снижение задержки критического трафика на 28% при нагрузке 80%, время адаптации к изменению топологии 45–60 мс с сохранением 98% правильности маршрутизации.

Полученные результаты согласуются с современными тенденциями в области управления потоками в SDN-сетях IoT [8]–[12].

Практическая значимость. Разработанная методика и программная реализация могут быть использованы при проектировании систем управления для IoT-сетей умных городов, промышленных M2M-сетей [13], а также в учебном процессе при подготовке специалистов в области сетевых технологий.

ЛИТЕРАТУРА

1. *Baktir*, *A. C.* How Can Edge Computing Benefit from Software-Defined Networking: A Survey, Use Cases, and Future Directions / A. C. Baktir, A. Ozgovde, C. Ersoy // IEEE Communications Surveys & Tutorials. – 2017. – Vol. 19, № 4. – P. 2359–2391.

2. *Network Slicing for IoT: A Survey of State-of-the-Art Challenges and Opportunities* / T. Oliveira, R. Pires, C. Marques, [et al.] // IEEE Internet of Things Journal. – 2022. – Vol. 9, № 15. – P. 13004–13024.

3. *Кузнецов*, *А. Н.* Применение технологий программно-конфигурируемых сетей в инфокоммуникационных системах / А. Н. Кузнецов, А. В. Петров, И. К. Сидоров // Труды СПИИРАН. – 2019. – Т. 18, № 6. – С. 1345–1368.

4. *Traffic Management in Software-Defined Networks*: *A Survey* / A. Mahmood, H. Zen, R. Hill, [et al.] // Journal of Network and Computer Applications. – 2023. – Vol. 210. – Art. № 103539.

5. *Семенов*, *М. И.* Методы обеспечения качества обслуживания в беспроводных сенсорных сетях Интернета вещей / М. И. Семенов, В. С. Васильев // Известия высших учебных заведений. Приборостроение. – 2021. – Т. 64, № 8. – С. 688–697.

6. *Gundall*, *M.* Integration of 5G with TSN as Prerequisite for a Highly Flexible Future Industrial IoT / M. Gundall, M. Strufe, H. D. Schotten // IEEE 2nd 5G World Forum (5GWF). – Dresden, 2019. – P. 299–304.

7. *Шабанов*, *А. А.* Адаптивные алгоритмы управления трафиком в программно-конфигурируемых сетях / А. А. Шабанов, Д. В. Иванов // Информатика и её применения. – 2020. – Т. 14, вып. 3. – С. 89–97.

8. *Модели управления процессами сбора данных в сетях интернета вещей с динамической структурой* / М. Т. Аунг, С. А. Аббас, Н. А. Жукова, В. В. Чернокульский // Computational nanotechnology. – 2020. – № 3. – С. 62–71.

9. *Huang*, *Y.* Distributed Cross-Domain Optimization for Software Defined Industrial Internet of Things / Y. Huang, S. Luo, W. Xu // Information. – 2023. – Vol. 14, № 2. – Art. № 101.

10. *Houhamdi*, *Z.* An Optimized SDN Framework for the Internet of Things / Z. Houhamdi, M. R. Athamena, B. Athamena // 11th International Conference on Software Defined Systems (SDS). – 2024. – P. 164–169.

11. *Ashraf*, *U.* ZFort A Scalable Zero-Trust Approach for Trust Management and Traffic Engineering in SDN Based IoTs / U. Ashraf // Internet of Things. – 2024. – Vol. 28. – Art. № 101352.

12. *Qi*, *Jin.* Optimized Transmission of Multi-Path Low-Latency Routing for Electricity Internet of Things / *Jin Qi* // PLOS ONE. – 2025. – Vol. 20, № 2. – Art. № e0314253.

13. *Рафалова*, *Е. В.* Особенности проектирования системы интернета вещей / Е. В. Рафалова, У. Д. Мешкова // Материалы X Республиканской научной конференции студентов, магистрантов и аспирантов «Актуальные вопросы физики и техники», Гомель, 21 апреля 2022 г. / Гомел. гос. ун-т им. Ф. Скорины ; редкол.: [и др.]. – Гомель : ГГУ им. Ф. Скорины, 2022. – Ч. 1. – С. 325–326.

Поступила в редакцию 21.06.2026.

Информация об авторах

Рафалова Елена Владиславовна – аспирант