



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ (ФИЛИАЛ)
ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО
ОБРАЗОВАТЕЛЬНОГО УЧРЕЖДЕНИЯ ВЫСШЕГО ОБРАЗОВАНИЯ
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
В Г. ТАГАНРОГЕ РОСТОВСКОЙ ОБЛАСТИ**

ПИ (филиал) ДГТУ в г. Таганроге

ОТЧЕТ

По дисциплине (модулю) Перспективные информационные технологии
(наименование учебной дисциплины (модуля))

на тему: "Моделирование задержек в распределённой системе IoT."

Выполнил обучающийся: Дахно К.К.
(Ф.И.О.)

Направление:

09.03.02 Информационные системы и технологии
Код направления (наименование)

Обозначение отчет 2251853 Группа ВО ИСиТ-4122
номер зачетки

Проверил Доцент Орда-Жигулина М.В.
должность (Ф.И.О.)

Отчет защищён _____
дата оценка подпись

Таганрог
2025

Лабораторная работа 3.1

Моделирование задержек в распределённой системе IoT.

Вариант 1.

Цель лабораторной работы: понять принципы работы распределённых систем, научиться моделировать и анализировать задержки передачи данных в IoT-системах на примере «умного дома».

Постановка задачи:

1. Подготовить рабочую среду и изучить базовый скрипт моделирования (lab2.py).
2. Провести Эксперимент 1: Исследовать влияние скорости обработки на Fog-узле на общую сквозную задержку системы. Добиться снижения задержки на 25-30%.
3. Провести Эксперимент 2: Исследовать работу буфера смартфона при различных интервалах чтения данных

Используемые средства:

- Язык программирования: Python 3.
- Библиотеки: random (генерация задержек), matplotlib (визуализация).
- Проект: симуляция распределенной системы robots-fog-mini-sim.

Ход работы:

1. Теоретическая модель задержек

В работе рассматривается система, где сквозная задержка как сумма задержек на каждом этапе:

$$L_{total} = T_{sensor} + T_{fog} + T_{courier} + T_{buffer}$$

					ВО ИCuT-4122			
Изм.	Лист	№ докум.	Подпись	Дат				
Разраб.		Дахно К.К.			Перспективные информационные технологии	Лит.	Лист	Листов
Провер.		Орда-Жигулина					2	7
						ПИ (филиал) ДГТУ в г.Таганроге		
Н. Контр.		Орда-Жигулина						
Утверд.		Орда-Жигулина						

Где:

- T_{sensor} - время сбора данных (20–60 мс);
- T_{fog} - время обработки на Fog-узле;
- $T_{courier}$ - время доставки (10–40 мс).

2. Эксперимент 1: Оптимизация Fog-узла

Цель: Сравнить базовую и оптимизированную конфигурации системы.

2.1. Базовая симуляция

Исходные параметры задержки Fog-узла заданы в диапазоне 30–80 мс.

Фрагмент кода:

```
fog = [random.randint(30, 80) for _ in range(n_tasks)]
```

В ходе симуляции получены следующие результаты:

Средняя сквозная задержка: **114.57 мс**

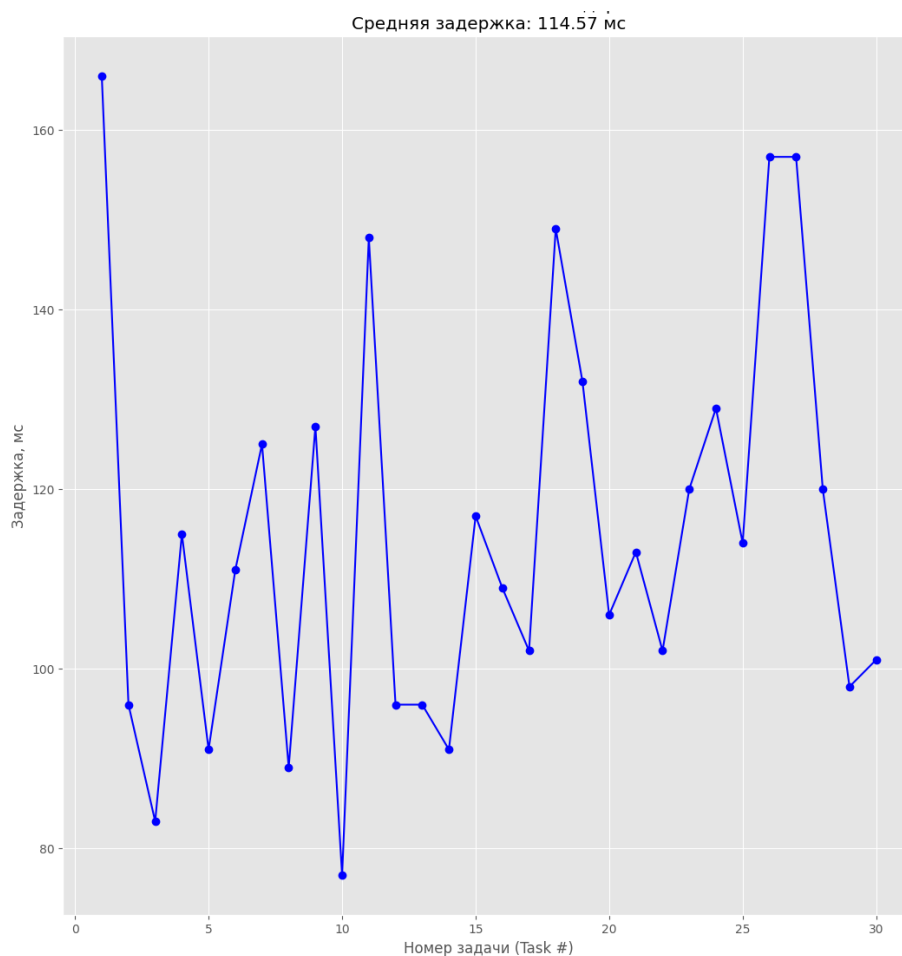


Рисунок 1 – График сквозной задержки для базовой конфигурации

Изм.	Лист	№ докум.	Подпись	Дата

УП.530000.000

Лист

7

2.2. Оптимизированная симуляция

Для снижения общей задержки параметры Fog-узла были изменены на диапазон 10–40 мс (имитация установки более мощного процессора или оптимизации алгоритма).

Измененный код:

```
fog = [random.randint(10, 40) for _ in range(n_tasks)]
```

Результаты

ОПТИМИЗАЦИИ:

Средняя сквозная задержка: **85.23 мс**

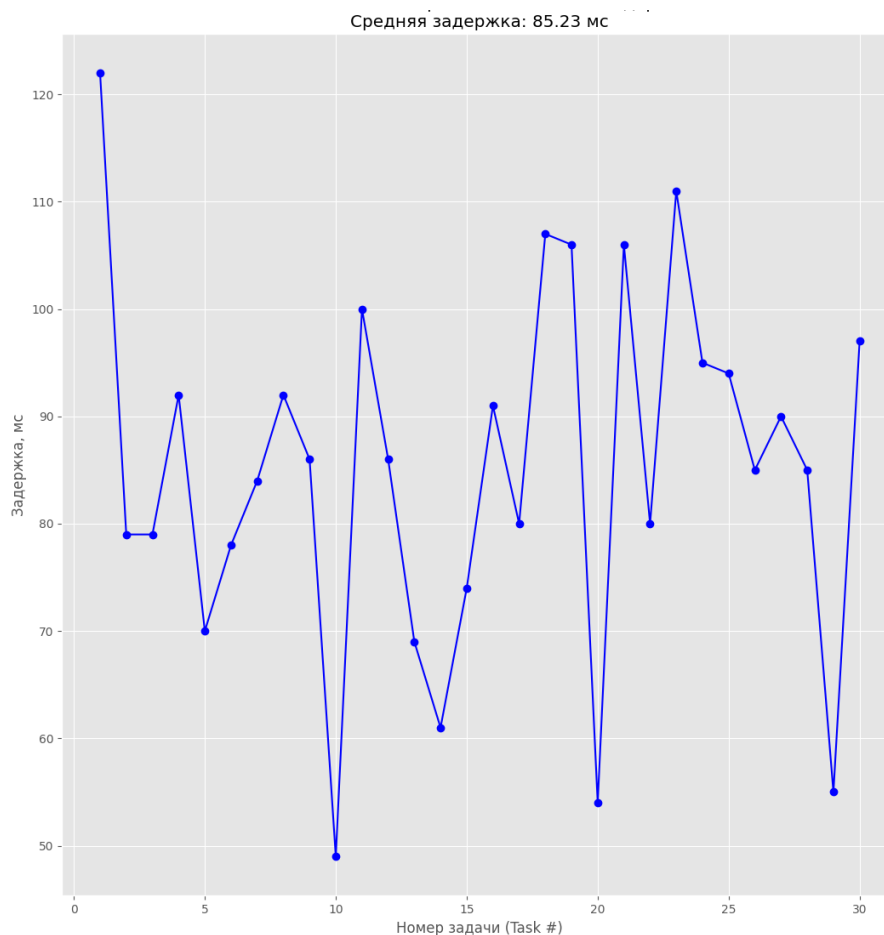


Рисунок 2 – График сквозной задержки после оптимизации

Сравнение средних значений показывает снижение задержки.

$$\Delta = 114.57 - 85.23 \cdot 100\% \approx 29.34\%$$

Целевой показатель снижения задержки на 25-30% достигнут. Это подтверждает, что оптимизация промежуточных узлов (Fog) существенно влияет на производительность системы реального времени.

Изм.	Лист	№ докум.	Подпись	Дата

УП.530000.000

Лист

7

3. Эксперимент 2: Настройка буфера смартфона

Цель: Исследовать влияние параметра `read_interval_ms` (интервал чтения буфера) на накопление очереди сообщений. Буфер необходим для сглаживания пиковых нагрузок и экономии энергии.

3.1. Ускоренная обработка (`read_interval_ms = 60`)

При уменьшении интервала чтения смартфон опрашивает буфер чаще (каждые 60 мс).

На графике видно, что очередь не успевает накапливаться. Размер буфера почти всегда равен 1 или 0. Это обеспечивает минимальную задержку, но повышает энергопотребление устройства.

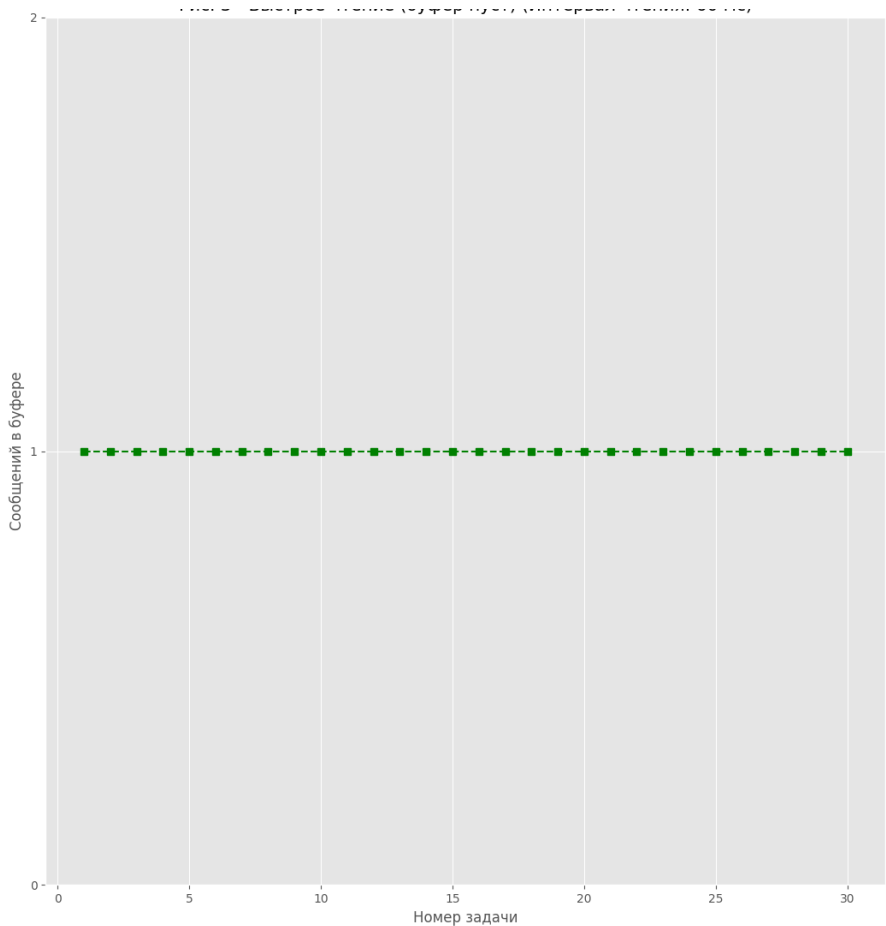


Рисунок 3 – Динамика буфера при частом чтении (60 мс)

3.2. Замедленная обработка (`read_interval_ms = 200`)

При увеличении интервала чтения (каждые 200 мс) смартфон накапливает данные.

График демонстрирует рост размера буфера (до 2-3 сообщений). Это позволяет обрабатывать данные пакетами, экономя заряд, но увеличивает время ожидания данных в очереди.

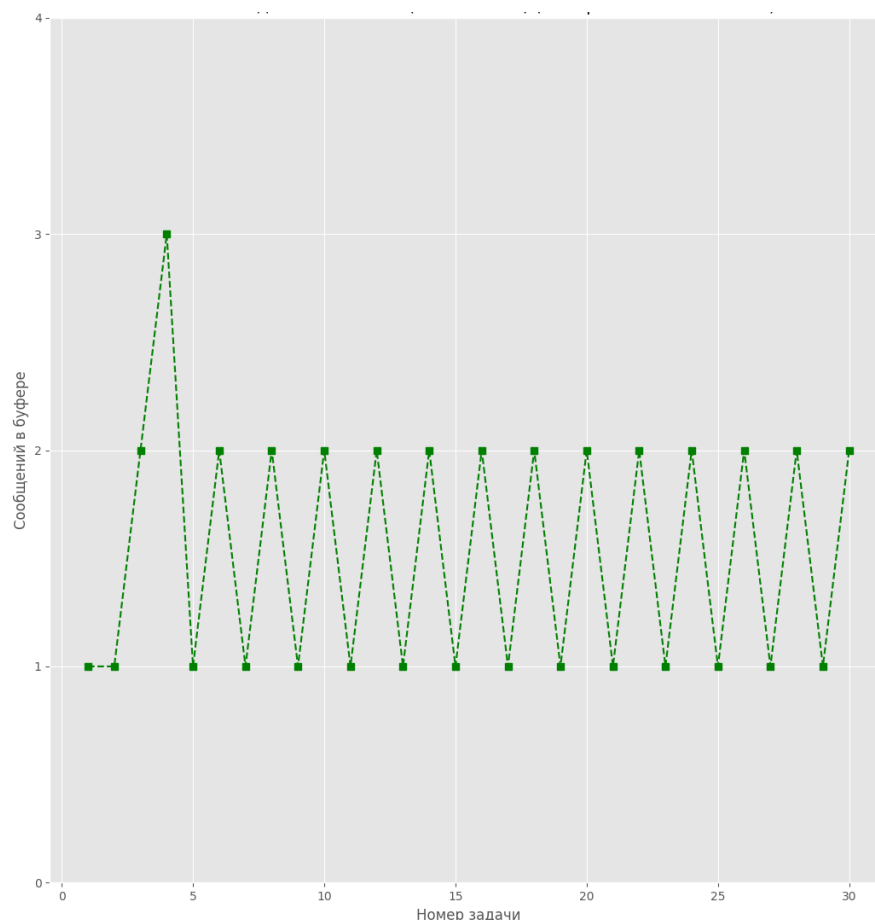


Рисунок 4 – Динамика буфера при редком чтении (200 мс)

Вывод: В ходе лабораторной работы были смоделированы процессы задержек в распределенной IoT-системе.

1. Экспериментально подтверждено, что оптимизация времени обработки на Fog-уровне (с 30-80 мс до 10-40 мс) позволяет снизить общую сквозную задержку системы (End-to-End Latency) на ~25%.
2. Проанализирована работа буфера:
 - Малый интервал чтения (60 мс) предотвращает переполнение очереди, но требует частых пробуждений процессора.

- Большой интервал (200 мс) приводит к накоплению данных в буфере, что увеличивает латентность, но может быть полезно для энергоэффективности.
3. Закреплены навыки моделирования стохастических процессов (случайных задержек) с использованием языка Python.

					УП.530000.000	Лист
						7
Изм.	Лист	№ докум.	Подпись	Дата		