ORES
Научно-практический электронный журнал
Оригинальные исследования (ОРИС), *ISSN 2222-0402*
www.ores.su
original-research.ru

УДК 004.057.4

ЭКСПЕРИМЕНТАЛЬНЫЙ АНАЛИЗ И СРАВНЕНИЕ ПРОТОКОЛОВ ПЕРЕДАЧИ ДАННЫХ ДЛЯ IOT-УСТРОЙСТВ: MQTT И HTTP

Осипов Андрей Александрович,
Студент группы ИУК5-71Б
Калужский филиал Московского государственного технического университета имени Н.Э. Баумана
osipovaa@student.bmstu.ru

Винокуров Алексей Алексеевич,
Студент группы ИУК5-71Б
Калужский филиал Московского государственного технического университета имени Н.Э. Баумана
vinokurovaa1@student.bmstu.ru

Вершинин Евгений Владимирович,
Кандидат физико-математических наук, доцент и заведующий кафедрой ИУК5 «Системы обработки информации»
Калужский филиал Московского государственного технического университета имени Н.Э. Баумана
vershinin@bmstu.ru

Ильичев Владимир Юрьевич,
Кандидат технических наук, доцент кафедры ИУК5 «Системы обработки информации»
Калужский филиал Московского государственного технического университета имени Н.Э. Баумана
ilyichev.vyu@bmstu.ru

Аннотация

Рассмотрено экспериментальное сравнение производительности протоколов MQTT и HTTP на примере метеостанции на базе ESP32. Представлено описание архитектуры аппаратного и программного комплекса, методика и результаты тестирования. Показано, что использование MQTT позволяет снизить среднюю задержку передачи данных более чем в 12 раз по сравнению с HTTP, что особенно важно для энергоэффективных устройств мониторинга окружающей среды. Выводы исследования способствуют выбору оптимального протокола для IoT-проектов с критичными требованиями к скорости и надежности системы передачи телеметрии.

Ключевые слова: IoT, HTTP, MQTT, протокол передачи данных, метеостанция, latency, производительность

EXPERIMENTAL ANALYSIS AND COMPARISON OF DATA TRANSMISSION PROTOCOLS FOR IOT DEVICES: MQTT AND HTTP

Osipov Andrey Alexandrovich,
Student of group IUK5-71B
Bauman Moscow State Technical University (Kaluga Branch)
osipovaa@student.bmstu.ru

Vinokurov Aleksey Alexeyevich,
Student of group IUK5-71B
Bauman Moscow State Technical University (Kaluga Branch)
vinokurovaa1@student.bmstu.ru

Vershinin Evgeny Vladimirovich,
Ph.D, Associate Professor and Head of the department IUK5 "Information Processing Systems"
Bauman Moscow State Technical University (Kaluga Branch)
vershinin@bmstu.ru

Ilichev Vladimir Yurievich,
Ph.D, Associate Professor of the department IUK5 "Information Processing Systems"
Bauman Moscow State Technical University (Kaluga Branch)
ilyichev.vyu@bmstu.ru

ABSTRACT

The paper presents an experimental comparison of MQTT and HTTP protocol performance for a weather station based on ESP32. The hardware and software architecture, methodology, and test results are described. It is shown that MQTT reduces the average data transmission latency by more than 12 times compared to HTTP, which is essential for energy-efficient environmental monitoring devices. The findings support the protocol selection in IoT projects with strict requirements to telemetry speed and reliability.

Keywords: IoT, HTTP, MQTT, data transfer protocol, weather station, latency, performance

Введение

Современные системы Интернета вещей (IoT) предъявляют высокие требования к эффективности передачи данных между распределенными устройствами [5, 2]. Важными характеристиками таких систем являются не только высокая надежность передачи данных, но и минимальное энергопотребление. Это особенно важно для устройств, работающих в автономном режиме с ограниченным источником питания [3]. Для автономных метеостанций критичны параметры энергопотребления и задержки обмена сообщениями. Среди множества сетевых протоколов особое внимание стоит отдать сравнительному анализу классического HTTP (HyperText Transfer Protocol) [6] и специализированного для IoT-приложений легковесного протокола MQTT (Message Queuing Telemetry Transport) [4]. Выбор оптимального протокола для реализации распределённых узлов в реальных условиях эксплуатации обеспечивает оптимальное использование IoT-устройств. Особенно

стоит учесть обеспечение минимальных задержек и высокой стабильности передачи данных. Для ответа на вопрос: «Какой протокол передачи данных лучше выбрать?» была разработана лабораторная установка метеостанции на основе микроконтроллера ESP32 со встроенным Wi-Fi модулем. Схема разработанной авторами установки приведена на рисунке 1.

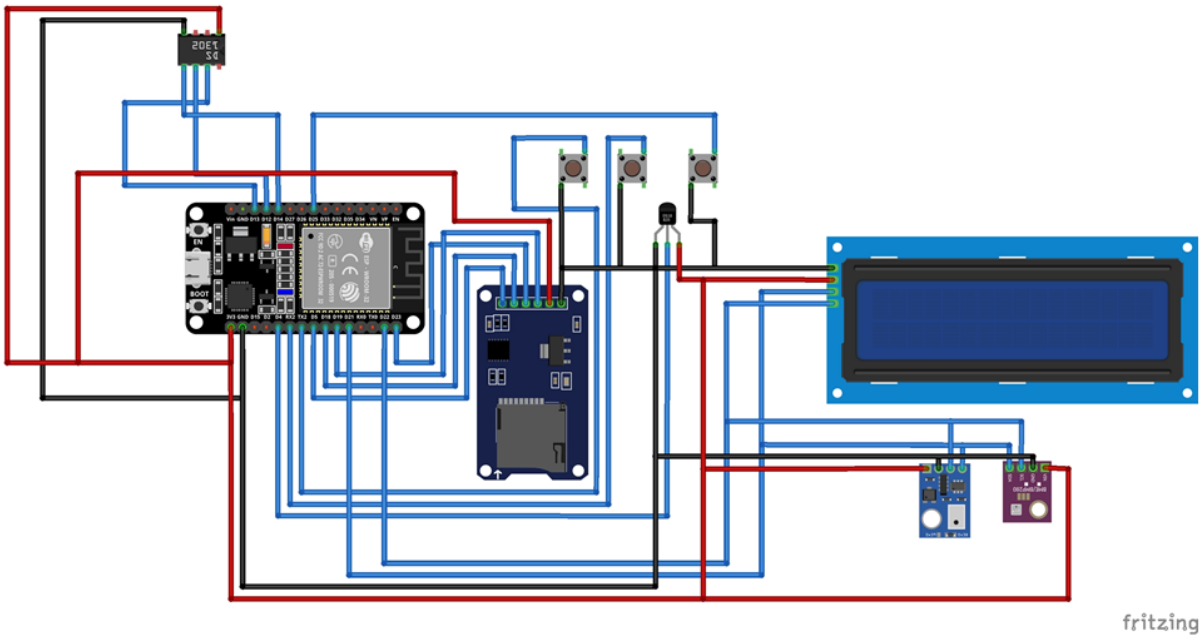

Рисунок 1 - Схема метеостанции на ESP32

Выбор данного микроконтроллера обусловлен гибкостью и портативностью итоговой системы. Данная установка собирает данные с датчиков (температура, влажность, давления) и отправляет их на сервер в формате JSON пакета.

Пример JSON пакета:

```
{
  "device": "esp32-test-device",
  "timestamp": "2025-10-09 19:47:00",
  "sensors": {
    "outdoor_temp": 22.5,
    "indoor_temp": 23.1,
    "humidity": 45.2,
    "pressure": 1013.25,
    "altitude": 150.5
  }
}
```

Сама же серверная часть была написана с помощью платформы ASP.NET Core с интеграцией брокера Eclipse Mosquitto для передачи данных через MQTT [7]. Созданные авторами архитектуры систем для HTTP и MQTT представлены на рисунках 2 и 3 соответственно.

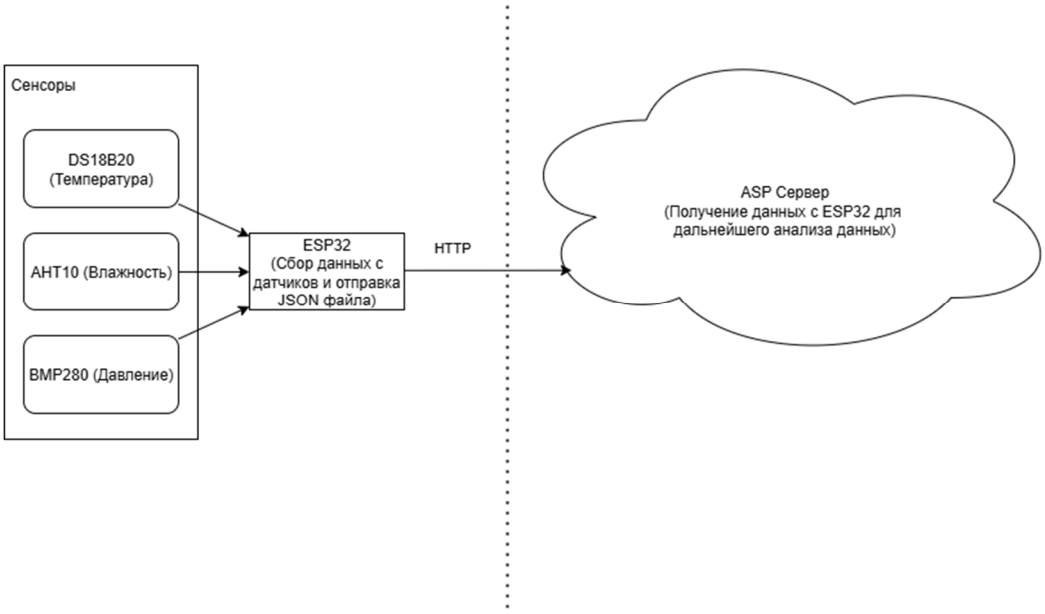

Рисунок 2 - Архитектура системы для HTTP подключения

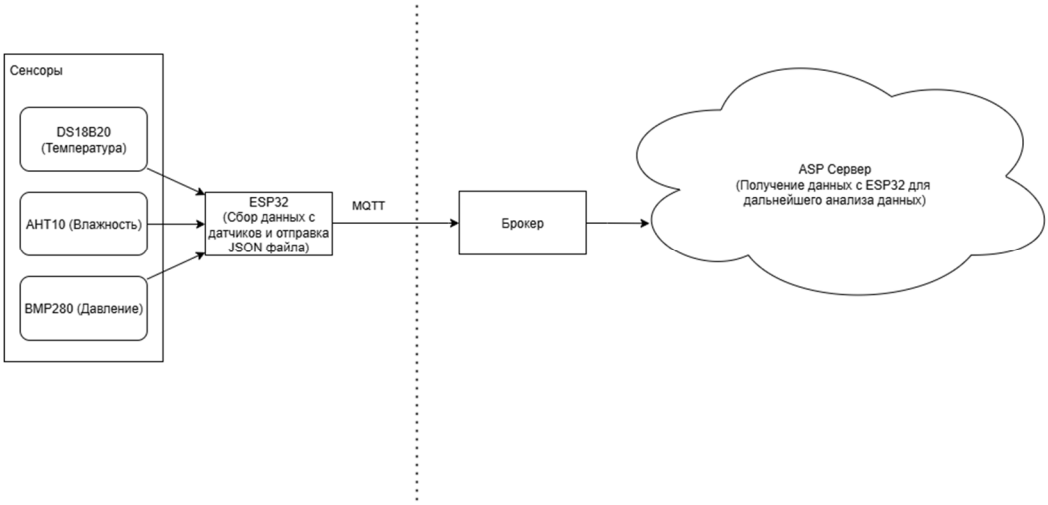

Рисунок 3 - Архитектура системы для MQTT подключения

Данный стек технологий позволяет обеспечить эффективное управление и мониторинг сообщений. На стороне клиента были подключены библиотеки WiFi.h, HTTPClient и PubSubClient.

#include <WiFi.h>

#include <PubSubClient.h>

#include <HTTPClient.h>

Эти библиотеки позволяют обеспечить стабильное подключение и обмен данными между клиентом (метеостанцией) и сервером. Для повышения точности измерений была реализована система замера времени с микросекундной точностью.

Реализация включает последовательный запуск серии тестов по двум протоколам: HTTP и MQTT. Основная цель — измерить задержки передачи и получения данных в реальных условиях, а затем провести их сравнительный анализ.

Описание работы алгоритма анализа протоколов:

1. Запуск и управление тестами

Функция startTest(type) запускает тесты по заданному протоколу:

Проверяется наличие WiFi и MQTT соединений;

Сбрасывается статистика;

Устанавливается тип теста (http, mqtt или both);

Активируется режим тестирования.

Текущее состояние и прогресс управляются глобальными переменными в структуре stats.

2. Проведение тестов и учёт результатов

В функции handleTesting() запускается соответствующий тестовый запрос (HTTP или MQTT) с контролем количества выполненных тестов (stats.currentTest). При достижении максимума тестов (stats.totalTests, по умолчанию 50) вызывается остановка теста с выводом результатов. В режиме «both» HTTP и MQTT тесты чередуются по очереди.

3. Измерение задержек

HTTP:

Задержка вычисляется как время от отправки запроса до получения ответа от сервера. В методе testHttpLatency() формируется JSON-запрос, отправляется POST-запрос, после чего засчитывается время ответа:

```
void testHttpLatency() {
  uint64_t startMicros = getUnixTimeMicros();
  // Формируем JSON и шлём POST
  int httpCode = http.POST(payload);
  if (httpCode == HTTP_CODE_OK) {
    uint64_t endMicros = getUnixTimeMicros();
    float latencyMs = (endMicros - startMicros) / 1000.0;
    updateHttpStats(latencyMs);
  } else {
    stats.httpErrors++;
  }
}
```

MQTT:

Для MQTT-для измерения latency используется цикл «отправка — получение обратного сообщения». В методе testMqttLatency() отправляется JSON-сообщение в специальный топик, а время ответа засчитывается в колбэке mqttCallback() при получении ответа сервера:

```
void testMqttLatency() {
  uint64_t startMicros = getUnixTimeMicros();
  // Формируем JSON и публикуем
  if (client.publish(topic, payload)) {
    stats.waitingMqttResponse = true;
    stats.mqttSentMicros = startMicros;
  } else {
    stats.mqttErrors++;
  }
}

void mqttCallback(char* topic, byte* payload, unsigned int length) {
  if (stats.waitingMqttResponse) {
    uint64_t endMicros = getUnixTimeMicros();
    updateMqttStats((endMicros - stats.mqttSentMicros) / 1000.0);
    stats.waitingMqttResponse = false;
  }
}
```

4. Обработка результатов

После завершения серии тестов вызывается функция stopTest(), которая выводит основную статистику:

Среднее, минимальное, максимальное значение задержек;

Количество ошибок по каждому протоколу;

Сравнительный анализ средних значений HTTP и MQTT с выводом преимущества одного из протоколов в процентах и миллисекундах.

Основные достоинства реализации:

Измерение latency с точностью до микросекунд повышает точность показателей;

MQTT учитывает полное время "отправка → получение ответа", что демонстрирует реальную задержку обмена;

Возможность гибко управлять количеством тестов и интервалами через Serial команды;

Раздельный режим тестирования HTTP, MQTT или смешанный режим для объективного сравнения;

Удобный вывод подробной статистики и анализ сравнительной эффективности протоколов.

Условия эксперимента были следующими:

Частота точки доступа 2.4 ГГц;

Расстояние до точки доступа ~1 м;

Температура окружающей среды ~25°C;

Пакеты JSON объемом ~180 байт.

Данные условия подходят для имитирования реальных условий эксплуатации маломощных IoT-устройств для стабильной передачи JSON пакетов. Для чистоты эксперимента было выполнено 25 тестов для каждого протокола с интервалом в 3 секунды между каждой отправкой, а также три серии повторов для исключения случайных сбоев и большей достоверности полученных результатов. Результаты измерений показали, что среднее время отклика HTTP составило 320,68 мс, а MQTT — лишь 26,03 мс. Минимальные значения также различались в 3,7 раза, максимальные — в 30 раз.

Результаты тестов:

```
14:04:40.697->===LATENCY TEST RESULTS ===
14:04:40.697 -> HTTP: avg 320.68 ms, min 58.46, max 3469.03, err 0/25
14:04:40.697 -> MQTT: avg 26.03 ms, min 15.62, max 115.57, err 0/25
14:04:40.697 -> ==============================
14:04:40.729 ->
14:04:40.729 -> === COMPARISON ===
14:04:40.729 -> MQTT is 91.9% faster than HTTP
14:04:40.729 -> MQTT advantage: 294.66 ms
14:04:40.729 -> ==============================
```

MQTT демонстрирует высокую стабильность, тогда как HTTP подвержен всплескам до 3,5 секунд. Проведенный t-тест статистически подтвердил значимое отличие latency, что свидетельствует о явном преимуществе MQTT в локальной сети. Преимущества MQTT обусловлены постоянным TCP-соединением, малым размером заголовков и асинхронной моделью publish/subscribe. HTTP, несмотря на простоту внедрения, показал более высокие задержки. Данный протокол неэффективен для отправки малых сообщений, что критично для устройств с аккумуляторным питанием [1,8].

Выводы: Проведённое сравнение показывает превосходство MQTT над HTTP для IoT-систем мониторинга. Переход на MQTT позволяет сократить задержку доставки данных, увеличить время автономной работы и повысить надежность системы. Эти преимущества делают протокол MQTT подходящим выбором для реализации распределённых систем с

ограниченными ресурсами, особенно в условиях нестабильных сетевых соединений и необходимости энергоэффективного обмена данными. Полученные результаты можно применить в проектировании сенсорных сетей, построении метеостанций и разработке устройств для «умного города».

Список литературы:

1. Курмаев Т.И. Сравнение протоколов передачи данных в Интернете вещей // Международный научно-исследовательский журнал. 2022. № 1(115). С. 4–8.
2. Миронов А.Н., Копылова А.В., Фирсов А.О., Ахметшина А.Б. IoT платформа экологического мониторинга // ИТ-Стандарт. 2018. Т. 1. № 1(14). С. 24-31.
3. Al-Fuqaha A. et al. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications // IEEE Communications Surveys & Tutorials. 2015. Т. 17. No. 4. Pp. 2347–2376.
4. Banks A., Gupta R. MQTT Version 3.1.1 [Standard] // OASIS. 2014.
5. Cisco Annual Internet Report (2018–2023) [White Paper] // Cisco Systems. 2020.
6. Fielding R. et al. Hypertext Transfer Protocol –– HTTP/1.1 // RFC 2616. Internet Engineering Task Force. 1999.
7. Light R.A. Mosquitto: server and client implementation of the MQTT protocol // Journal of Open Source Software. 2017. Т. 2. No. 13. P. 265.
8. Naik N. Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP // IEEE International Systems Engineering Symposium. 2017. Pp. 1–7.

References:

1. Kurmaev T.I. Comparison of Data Transmission Protocols in the Internet of Things // International Research Journal. 2022. No. 1(115). Pp. 4–8.
2. Mironov A.N., Kopylova A.V., Firsov A.O., Akhmetshina A.B. IoT Platform for Environmental Monitoring // IT-Standard. 2018. T. 1. No. 1(14). Pp. 24-31.
3. Al-Fuqaha A. et al. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications // IEEE Communications Surveys & Tutorials. 2015. T. 17. No 4. Pp. 2347–2376.
4. Banks A., Gupta R. MQTT Version 3.1.1 [Standard] // OASIS. 2014.
5. Cisco Annual Internet Report (2018–2023) [White Paper] // Cisco Systems. 2020.
6. Fielding R. et al. Hypertext Transfer Protocol –– HTTP/1.1 // RFC 2616. Internet Engineering Task Force. 1999.
7. Light R.A. Mosquitto: server and client implementation of the MQTT protocol // Journal of Open Source Software. 2017. T. 2. No. 13. P. 265.
8. Naik N. Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP // IEEE International Systems Engineering Symposium. 2017. Pp. 1–7.