

Катін П.Ю.

<https://orcid.org/0000-0002-2542-9976>

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Яценко Д.В.

<https://orcid.org/0009-0004-8114-6112>

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Мішин О.В.

<https://orcid.org/0009-0008-4766-6324>

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ЧАСОВИХ ЗАТРИМОК ПЕРЕДАЧІ ТА ОБРОБКИ GNSS-ДАНИХ У РІЗНИХ ФОРМАТАХ ПРЕДСТАВЛЕННЯ ДЛЯ СИСТЕМ ІОТ

У роботі досліджено ефективність передачі навігаційних телеметричних повідомлень GNSS у системах Інтернету речей. Актуальність даного напрямку досліджень обумовлена широким використанням навігаційних систем моніторингу транспорту, робототехнічних платформ та інших IoT-застосувань, у яких необхідна оперативна передача координатних та телеметричних даних до серверних систем обробки. Навігаційні приймачі формують повідомлення відповідно до стандарту NMEA, які передаються у вигляді текстових ASCII-рядків та містять інформацію про координати об'єкта, швидкість руху, курс та час визначення позиції.

У сучасних IoT-системах передача таких повідомлень до серверної частини системи здійснюється за допомогою мережесевих протоколів прикладного рівня, зокрема HTTP. Навігаційні повідомлення можуть передаватися як у сирому текстовому вигляді, так і у структурованих форматах представлення даних, зокрема JSON або у вигляді параметрів HTTP-запитів. Використання різних форматів представлення телеметричних повідомлень може створювати додаткові накладні витрати, пов'язані із серіалізацією, кодуванням та синтаксичним аналізом даних під час їх обробки серверною системою.

У роботі проведено експериментальне дослідження впливу різних форматів передачі GNSS-повідомлень на затримку обробки HTTP-запитів у серверному застосуванні. Для проведення експерименту було розроблено програмний стенд, що моделює процес передачі телеметричних даних від клієнтського пристрою до серверної частини системи. Було сформовано схему проведення експерименту та обробку даних. Програмний стенд реалізовано у вигляді двох складових, а саме: веб-сервісу на базі фреймворку Django та клієнтського модуля для формування HTTP-запитів на основі бібліотеки requests мови програмування Python.

У рамках дослідження виконано порівняльний аналіз кількох способів передачі телеметричних повідомлень, зокрема raw NMEA, application/json, application/x-www-form-urlencoded, multipart/form-data та експериментального текстового формату TOON. Для кожного способу передачі виконано серію експериментальних вимірювань часу обробки HTTP-запитів із формуванням статистичної вибірки обсягом 300 спостережень. Обробка результатів здійснювалася із використанням методів математичної статистики, зокрема обчислення вибіркового середнього, мінімального та максимального значень, а також вибіркового середньоквадратичного відхилення. Проведено дослідження на відповідність вибірок відомим законам розподілу.

Отримані результати показали, що передача сирого NMEA-повідомлення забезпечує найменшу затримку обробки запитів, оскільки не потребує додаткових операцій серіалізації та перетворення



структури даних. Формати JSON та TOON демонструють близькі значення затримки, тоді як формати *application/x-www-form-urlencoded* та *multipart/form-data* характеризуються більшими накладними витратами під час обробки повідомлень. Результати дослідження можуть бути використані при проектуванні IoT-систем навігаційного моніторингу, у яких важливим є забезпечення мінімальної затримки передачі телеметричних даних та ефективної обробки GNSS-повідомлень.

Ключові слова: GNSS, NMEA, IoT, телеметрія, HTTP, JSON, передача даних, затримка передачі, навігаційний моніторинг, математична статистика.

Постановка проблеми. У сучасних комп'ютерних мережах важливу роль відіграють технології IoT (Internet of Things, Інтернет речей), що забезпечують взаємодію їх складових. Одним із важливих елементів IoT є системи навігаційного моніторингу, що використовують приймачі GNSS (Global Navigation Satellite System, глобальна навігаційна супутникова система) для визначення координат об'єктів, швидкості їх руху та отримання точного часу.

Приймачі формують навігаційні повідомлення відповідно до стандарту NMEA (National Marine Electronics Association, стандарт навігаційних повідомлень). Вони передаються у вигляді текстових ASCII-рядків та містять інформацію про координати, швидкість, курс руху, час, тощо.

У IoT-архітектурах навігаційні повідомлення передаються від пристроїв до серверних систем обробки даних через мережеві протоколи прикладного рівня. Найбільш поширеними протоколами є HTTP (HyperText Transfer Protocol, протокол передачі гіпертексту) та MQTT (Message Queuing Telemetry Transport, протокол телеметричного обміну повідомленнями). Формати представлення навігаційних даних можуть мати текстовий вигляд (сирі дані, raw NMEA) і структурований формат серіалізації, зокрема JSON (JavaScript Object Notation, формат представлення структурованих даних). Може застосовуватися у вигляді параметрів HTTP-запитів.

Процеси серіалізації, кодування та обробки даних створюють затримку інформації під час передавання. У системах управління на основі технологій IoT затримка повідомлень може мати негативний вплив на роботу системи у цілому. Таким чином, актуальною є задача дослідження залежності часу затримки передачі повідомлень від формату даних GNSS.

Аналіз останніх досліджень і публікацій. На теперішній час спостерігається стрімкий розвиток систем IoT та їх широке використання у задачах моніторингу транспорту, логістики, навігації та керування рухомими об'єктами. Важливим елементом IoT є система передачі телеметричних даних від периферійних пристроїв до серверних

систем обробки та особливості протоколів передачі даних, що описані у [1-5, 10, 11, 13, 14].

Важливим фактором ефективності передачі є формат представлення даних. Зокрема у веб-сервісах широко застосовується текстовий формат JSON [1], який забезпечує зручність інтеграції з програмними застосунками. Альтернативою можуть виступати більш компактні формати, наприклад CBOR [3], що дозволяють зменшити обсяг передаваних повідомлень.

Для побудови серверної частини IoT-систем широко використовуються сучасні веб-фреймворки, наприклад Django [6], що забезпечують швидку розробку серверних застосунків та підтримку різних форматів обміну даними. Широко використовуються навігаційні супутникові системи GNSS, що формують повідомлення стандарту NMEA-0183 [4].

Передача телеметричних даних у системах Інтернету речей може здійснюватись з використанням протоколів HTTP або MQTT. Протокол MQTT визначений стандартом OASIS [11] та широко використовується у IoT-системах. Дослідження ефективності використання MQTT у IoT-системах наведено у роботі [13], де розглянуто можливості застосування цього протоколу.

У [14] наведено порівняльний аналіз протоколів MQTT та HTTP з точки зору ефективності передачі даних у IoT-застосуваннях. Автори показують, що текстова структура HTTP-повідомлень і значний обсяг службових заголовків можуть призводити до збільшення обсягу передаваних даних.

Таким чином, аналіз існуючих наукових досліджень показує, що значна увага приділяється архітектурі IoT-систем, мережевим протоколам передачі даних та програмній архітектурі вбудованих пристроїв.

Водночас у науковій літературі недостатньо досліджено вплив різних форматів представлення GNSS-повідомлень на часову затримку. Відсутні експериментальні дослідження, у яких проводиться порівняння затримок передачі повідомлень GNSS у різних форматах представлення даних (raw NMEA, JSON, *application/x-www-form-urlencoded*, *multipart/form-data*).

Відомі роботи [12], у яких вирішуються питання щодо реалізації і використання методики дослідження затримок обчислення у мікроконтролерних системах але ця методологія потребує подальшого розвитку для вирішення мережевих завдань.

Існує багато сучасних робіт із математичної статистики [6-9, 15], що дають можливість розвинути результати роботи [12] для дослідження часових затримок передачі GNSS-повідомлень у IoT-системах.

Постановка завдання. Метою статті є експериментальне дослідження часу передачі (затримки) навігаційних GNSS-повідомлень у системах Інтернету речей при використанні різних форматів представлення даних у HTTP-запитах та статистична обробка цих результатів.

Експериментальна програмна архітектура та обробка результатів. Розроблена архітектура експериментальної системи, що побудована за клієнт-серверною архітектурою та складається з таких основних компонентів:

- модуль генерації телеметричних даних;
- клієнтський модуль формування HTTP-запитів;
- серверний модуль приймання та обробки даних;
- модуль збору та статистичної обробки результатів.

У роботі досліджується передача телеметричних повідомлень, сформованих на основі стандарту NMEA, з використанням кількох форматів представлення даних. Основною метрикою оцінювання ефективності є затримка доставки повідомлення від клієнтського пристрою до серверного застосунку.

Обмеження на проведення експерименту наступні: дослідження виконано у локальному програмному середовищі, використана операційна система загального призначення, дослідження провадилися на одній робочій машині без урахування затримок глобальної мережі передачі даних.

Обробка результатів експерименту на першому етапі передбачала:

отримання з використанням програмної архітектури вибірок часу затримки для різних форматів передавання даних, кількість елементів у вибірках дорівнювала 300, оскільки система вважається априорі технічно справною, введено припущення про нормальний закон розподілу вибірки;

далі здійснювалася типова статистична обробка вибірок, приклад методології наведений у [12], на

завершальному етапі дослідження перевірялась відповідність розподілу у вибірках нормальному закону.

Виклад основного матеріалу. Наведемо опис архітектури експериментальної системи та призначення її основних складових. До них належить модуль генерації телеметричних даних, що формує тестові GPS-повідомлення у форматі NMEA.

Клієнтський модуль реалізовано мовою програмування Python з використанням бібліотеки requests. Його основною функцією є формування та відправлення HTTP-запитів до серверного додатку у різних форматах передачі даних.

Для кожного формату клієнтський модуль формує відповідний HTTP-запит та передає телеметричні дані серверу. Серверний модуль реалізовано у вигляді веб-додатку на базі фреймворку Django.

Сервер приймає HTTP-запити, виконує їх синтаксичний аналіз відповідно до формату передачі даних та здійснює обробку отриманої телеметричної інформації. У ході експерименту для кожного запиту фіксувався час його обробки сервером. Отримані значення часу обробки використовувалися для формування статистичної вибірки.

Статистична обробка результатів експерименту виконувалася із використанням бібліотеки Pandas, яка забезпечує зручні засоби роботи з табличними даними та дозволяє обчислювати основні статистичні характеристики вибірки, такі як середнє значення, мінімум, максимум та середньоквадратичне відхилення.

Таким чином, розроблена експериментальна система дозволяє моделювати процес передачі телеметричних даних у різних форматах та виконувати порівняльний аналіз ефективності їх обробки серверною системою.

Затримка обробки запиту визначалася як інтервал часу між моментом відправлення HTTP POST-запиту клієнтом та моментом отримання відповіді сервера. Для вимірювання часу використовувалася високоточна функція `time.perf_counter()`.

Дослідженню підлягали такі формати передачі даних:

- raw NMEA-потік (`text/nmea`);
- `application/json`;
- експериментальний текстовий формат TOON;
- `application/x-www-form-urlencoded`;
- `multipart/form-data`.

Для формату `application/json` клієнт виконує серіалізацію структури даних у текстовий формат JSON, а сервер – десеріалізацію та парсинг отриманого повідомлення, що потребує додаткових обчислювальних операцій під час обробки запиту.

Формат `application/x-www-form-urlencoded` передбачає кодування полів у вигляді послідовності параметрів `key=value&key=value` із можливим URL-екранізуванням символів. Такий підхід збільшує обсяг переданих даних та потребує додаткового розбору ключів і значень на стороні сервера.

`Multipart/form-data` використовує механізм `boundary`-роздільників для інкапсуляції окремих частин повідомлення, кожна з яких містить службові заголовки, що створює додаткові накладні витрати при парсингу.

Експериментальний текстовий формат `TOON` передбачає компактне представлення структурованих даних із зменшенням синтаксичної надлишковості `JSON`, однак також потребує попереднього перетворення даних перед передачею.

У сирому режимі передавалося оригінальне `NMEA`-речення без попереднього перетворення. У структурованих режимах виконувалося виділення полів і формування контейнера передачі.

Для кожного формату передачі даних було сформовано статистичну вибірку обсягом $N = 300$ вимірювань. Повторення експерименту дозволяє зменшити вплив випадкових факторів та отримати більш достовірну оцінку часу обробки запитів.

Для статистичного аналізу результатів використовувалися такі характеристики вибірки:

Вибіркове середнє

Середній час обробки запитів для кожного формату передачі даних визначається за формулою:

$$\bar{t} = \frac{1}{N} \sum_{i=1}^N t_i, \quad (1)$$

де,

t_i – значення часу обробки i -го запиту;

N – кількість вимірювань.

Вибіркове середнє дозволяє оцінити типову затримку обробки запитів для відповідного формату передачі даних.

Мінімальне та максимальне значення

Для характеристики меж зміни часу обробки визначалися також мінімальне та максимальне значення вибірки

$$t_{\{min\}} = \min(t_1, t_2, \dots, t_N) \quad (2)$$

$$t_{\{max\}} = \max(t_1, t_2, \dots, t_N) \quad (3)$$

Ці показники дозволяють оцінити найкращий та найгірший випадок обробки запитів у рамках проведеного експерименту.

Вибіркове середньоквадратичне відхилення

Для оцінки розсіювання значень часу обробки запитів відносно середнього значення використовувалося вибіркове стандартне відхилення

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (t_i - \bar{t})^2} \quad (4)$$

де,

\bar{t} – вибіркове середнє значення часу обробки.

Стандартне відхилення характеризує стабільність роботи системи: чим менше значення σ , тим меншою є варіативність часу обробки запитів.

Результати дослідження зведені у таблицю 1, що містить формати передавання даних та результати статистичної обробки даних.

Результати таблиці 1 демонструють, що передача сирого `NMEA`-повідомлення забезпечує найменшу середню затримку та мінімальний обсяг переданих даних. Формати з попередньою адаптацією збільшують розмір `payload` та створюють додаткові накладні витрати на серіалізацію і розбір структури. Різниця між `JSON` та `TOON` знаходиться в межах статистичної похибки вимірювання у локальному стенді. Методи `form` та `multipart` характеризуються найбільшою затримкою через додаткове кодування та обробку полів.

Для візуалізації результатів експерименту на рис. 1 представлено графічне відображення таблиці 1.

У роботі [12] передбачався нормальний закон розподілу вибірок. Для більш точного оцінювання результатів і для виконання програми проведення експерименту було проведено окреме статистичне дослідження відповідності отриманих вибірок нормальному закону. Отже, відповідно до результатів, що наведені у [12] сформовано початкову гіпотезу про нормальний закон розподілу в експериментальних вибірках. Щільність нормального

Таблиця 1

Результати вимірювання затримки доставки (300 повторів)

Формат передачі	Середня затримка, мс	Мінімальна затримка, мс	Максимальна затримка, мс	Середньоквадратичне відхилення, мс
form	1.799304	1.410667	43.690584	2.668516
multipart	1.965303	1.573125	7.992958	0.745291
json	1.409744	1.249500	2.796708	0.140111
toon	1.606506	1.224667	17.171750	1.256161
raw_nmea	1.407530	1.230875	2.195667	0.117626

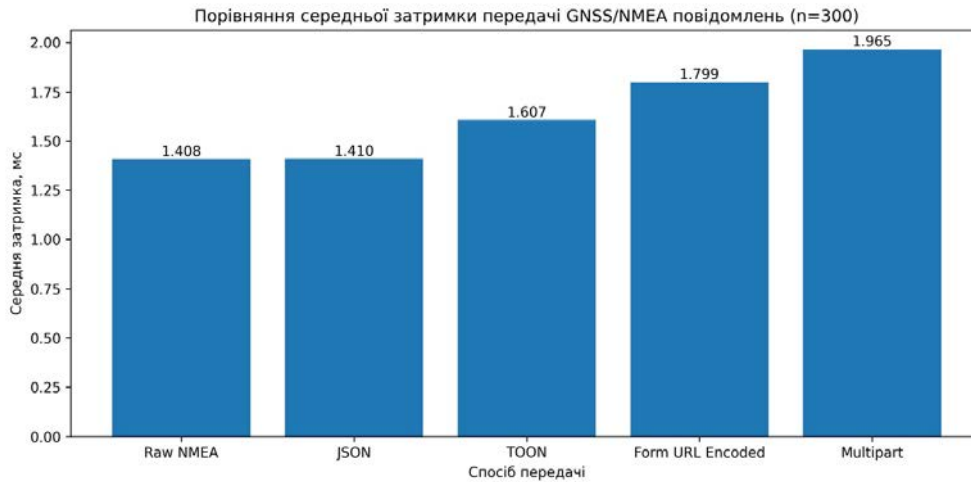


Рис. 1. Порівняння середньої затримки передачі GNSS/NMEA повідомлень для різних форматів

розподілу визначається наступною формулою [6-9, 15]:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \quad (5)$$

де μ – середнє значення, σ – середньоквадратичне відхилення.

Цей розподіл має властивість симетрії відносно середнього значення, крім того, ця функція ніколи не досягає строго нуля.

Гіпотеза про нормальний розподіл затримок не підтвердилась. На рис.2 наведено гістограму розподілу вимірів для одного з форматів (TOON) та графіки різних розподілів. Графіки гістограми рис.2, вказують на часткову невідповідність розподілу у вибірці нормальному закону розподілу. Відповідно до рис.2 явно виражена несиметричність гістограми результатуючих вибірок. Це можна пояснити механізмом утворення затримки. Обмеження з лівого боку гістограми рис.2 обумовлено тим, що на виконання кожного кроку алгоритму потрібен певний мінімальний час. Саме це дає строге обмеження на мінімальне значення затримки.

Наявність поширення правого боку гістограми рис.2 обумовлено випадковими затримками обчислювальної системи та операційної системи, яка відповідно умов експерименту не є операційною системою реального часу. Саме ці умови впливають на випадковий характер і будуть збільшувати це значення.

Наявність строгого мінімуму та явну несиметричність розподілу відносно середнього значення більш точно моделюють гамма-розподіл та логарифмічний нормальний розподіл, формула

якого визначається виразом [6-9, 15]:

$$f(x) = \frac{x^{\alpha-1} e^{-\beta x} \beta^\alpha}{\Gamma(\alpha)}, \quad (6)$$

де α, β – параметри розподілу. Гамма розподіл падає строго до нуля при $x=0$, і, додавши додатковий параметр зсуву, можна підігнати цей нуль відповідно до моделі затримки.

Також для математичного моделювання закону розподілу отриманих вибірок можна застосувати щільність логарифмічного нормального розподілу, що визначається наступним виразом [6-9, 15]:

$$f(x) = \frac{1}{x\sqrt{2\pi}\sigma} e^{-\frac{(\ln(x)-\mu)^2}{2\sigma^2}}, \quad (7)$$

Як видно з назви, цей розподіл можна отримати, якщо прийняти, що логарифм випадкової величини має нормальний розподіл, з відповідними параметрами.

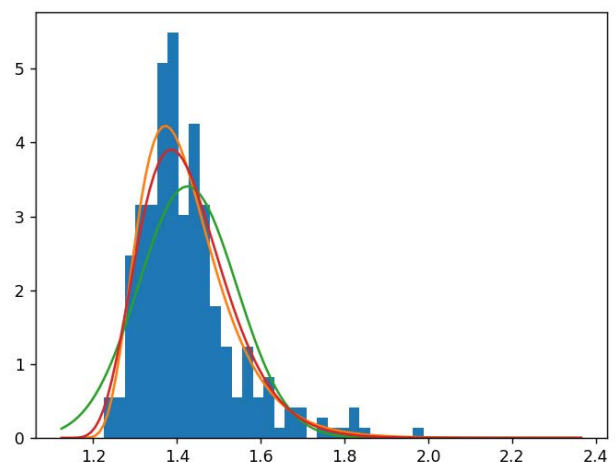


Рис. 2. Гістограма вимірів затримок, графік нормального розподілу(зелений), гамма-розподілу(червоний), та логарифмічного нормального розподілу(оранжевий)

Таким чином, наочно видно несиметричність гістограми та більш точна відповідність альтернативних розподілів. Для графічного порівняння відповідності двох різних розподілів як правило використовуються графіки Quantile-Quantile, або QQ-графіки [6-9, 15]. На такому графіку, кожна точка має координати, що відповідають певному квантилю для першого і другого розподілів. Для розподілів, наведених як множина вимірів, якщо кількість вимірів однакова, для цього достатньо відсортувати елементи цих множин в порядку зростання та, для кожного номера, поставити точку з координатами, що відповідають значенням елементів з цим номером. Для розподілів, наведених у вигляді функції щільності розподілу, цю функцію можна використати для обрахунку квантилів.

Результуючий графік порівнюється з прямою. Якщо порівняні розподіли однакові, точки графіка будуть лежати на прямій $y=x$. Якщо вони лежать на іншій прямій, це відповідає лінійній залежності двох розподілів. На рис.3 наведено порівняння графіків QQ-графіків для двох різних гіпотез розподілів. Більш точна відповідність лінії відповідає більшій точності гіпотези розподілу.

Крім цього, дослідження даних вказало на наявність аномально великих викидів в перших вимірах вибірки. Це пояснюється затримкою системи для виконання одноразових дій (виділення пам'яті, кешування і т.д.). Ці значення не є суттєвими для мети цієї роботи (порівняння різних форматів даних) і були відкинуті перед обчисленнями.

Хоча точний закон розподілу не підтверджено, використання обчислених значень (середнє, середньоквадратичне відхилення) є допустимим, з урахуванням вказаних вище особливостей. Побудова моделі випадкового процесу затримки і використання цієї моделі для підтвердження від-

повідності цього процесу одному з видів розподілів, є одним з напрямів подальшого дослідження.

Слід зазначити, що дослідження виконано в локальному середовищі, тому результати відображають сумарні накладні витрати формату передачі та серверної обробки, а не абсолютну мережеву затримку в реальних IoT-каналах. У розподілених системах із віддаленими вузлами різниця між підходами може змінюватися залежно від пропускної здатності каналу та частоти оновлення телеметрії.

Висновки. У роботі проведено експериментальне дослідження впливу формату передачі телеметричних даних на затримку обробки HTTP-запитів у серверній системі. Розроблено програмну архітектуру, що моделює процес передачі GNSS/NMEA повідомлень між клієнтським та серверним модулями.

За результатами експерименту встановлено, що передача сирого NMEA-повідомлення забезпечує мінімальні накладні витрати та найменшу затримку обробки. Формати передачі даних, що передбачають додаткову серіалізацію структури (JSON, TOON), створюють додаткові обчислювальні витрати на стороні клієнта та сервера.

За результатами досліджень не підтвердився нормальний закон розподілу у отриманих вибірках, що суперечить результатам [12], цей факт обґрунтований у вищевикладеному матеріалі. Також було виявлено, що розподіл у отриманих вибірках не відповідає відомим законам розподілу і як майбутній напрям дослідження передбачається отримання формули для закону розподілу у вибірках.

Отримані результати можуть бути використані під час проектування IoT-систем моніторингу, у яких важливими є мінімальна затримка передачі телеметрії та ефективність обробки GNSS-даних.

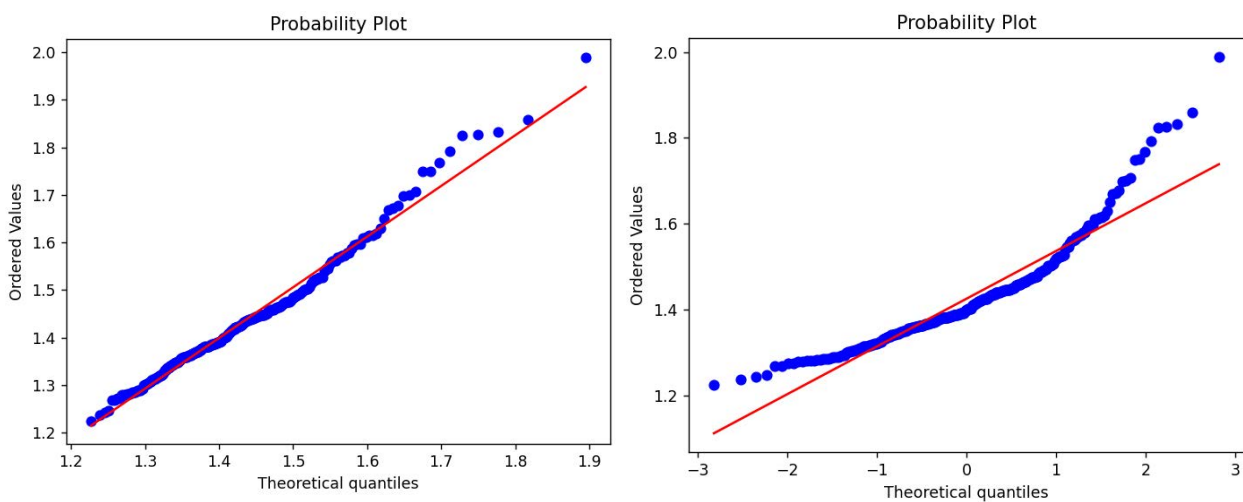


Рис. 3. QQ-графіки для логарифмічного нормального розподілу(зліва) та нормального розподілу(справа)

Список літератури:

1. Bray T. The JavaScript Object Notation (JSON) Data Interchange Format : RFC 8259. Internet Engineering Task Force (IETF), 2017. 16 с. URL: <https://datatracker.ietf.org/doc/html/rfc8259> (дата звернення: 07.03.2026).
2. Fielding R. та ін. Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing : RFC 7230. Internet Engineering Task Force (IETF), 2014. 89 с. URL: <https://datatracker.ietf.org/doc/html/rfc7230> (дата звернення: 07.03.2026).
3. Bormann C., Hoffman P. Concise Binary Object Representation (CBOR): RFC 8949. [Чинний від 2020-12]. Internet Engineering Task Force (IETF), 2020. URL: <https://datatracker.ietf.org/doc/html/rfc8949> (дата звернення: 07.03.2026).
4. NMEA 0183 Standard for Interfacing Marine Electronic Devices. Version 4.30. National Marine Electronics Association, 2023. URL: <https://www.nmea.org/nmea-0183.html> (дата звернення: 07.03.2026).
5. Al-Fuqaha A. та ін. Internet of Things: A Survey on Enabling Technologies, Protocols and Applications. IEEE Communications Surveys & Tutorials. 2015. Vol. 17, № 4. С. 2347-2376. URL: <https://ieeexplore.ieee.org/document/7123563> (дата звернення: 07.03.2026).
6. Mele A. Django 5 By Example: Build powerful and reliable Python web applications from scratch. 5th ed. – Birmingham: Packt Publishing, 2024. 822 p.
7. Горбачук В. М., Кушлик-Дивульська О. І. Теорія ймовірностей та математична статистика: підручник. Київ : КПІ ім. Ігоря Сікорського, 2023. 351 с.
8. Озеранський В. С., Крилик Л. В., Шевчук О. Ф. Теорія ймовірності та математична статистика для здобувачів спеціальності 122 «Комп'ютерні науки»: електронний лабораторний практикум. Вінниця : ВНТУ, 2025. 113 с.
9. Донченко В. С., Сидоров М. В.-С. Теорія ймовірностей та математична статистика для соціальних наук : навч. посіб. Київ : Київський університет, 2015. 400 с.
10. Token-Oriented Object Notation (TOON) Specification. Version 3.0. 2025. URL: <https://toonformat.dev/> (дата звернення: 20.03.2026).
11. Banks A., Gupta R. MQTT Version 3.1.1 : OASIS Standard. OASIS Open, 2014. 81 p. URL: <https://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html> (дата звернення: 20.03.2026).
12. Катін П.Ю., Похиленко О.А., . Електрон. моделювання. 2021. Т. 43. № 2, с. 51-67 URL: https://web.archive.org/web/20210716145023id_/https://www.emodel.org.ua/images/em/43-2/43-2-5.pdf (дата звернення: 20.03.2026).
13. Sasaki Y., Yokotani T. Performance Evaluation of MQTT as a Communication Protocol in IoT. Advances in Technology Innovation. 2019. Vol. 4, № 1. P. 21-29.
14. Kadam S., Ghule S. Comparative Analysis of MQTT and HTTP Protocols for IoT Applications. International Journal of Innovative Research in Science, Engineering and Technology. 2023.
15. Огірко О. І., Галайко Н. В. Теорія ймовірностей та математична статистика: навч. посіб. Львів: ЛьВДУВС, 2017. 292 с.

Katin P.Yu, Yatsenko D.V., Mishyn O.V. EXPERIMENTAL STUDY OF TRANSMISSION AND PROCESSING LATENCY OF GNSS DATA IN DIFFERENT DATA REPRESENTATION FORMATS FOR IOT SYSTEMS

This paper investigates the efficiency of transmitting GNSS navigation telemetry messages in Internet of Things (IoT) systems. The relevance of this research area is determined by the widespread use of navigation monitoring systems for transportation, robotic platforms, and other IoT applications in which the prompt transmission of coordinate and telemetry data to server-side processing systems is required. Navigation receivers generate messages according to the NMEA standard, which are transmitted as ASCII text strings and contain information about the object's coordinates, speed, course, and time of position determination.

In modern IoT systems, the transmission of such messages to the server side of the system is performed using application-layer network protocols, in particular HTTP. Navigation messages can be transmitted either in raw textual form or in structured data representation formats, such as JSON, or as parameters of HTTP requests. The use of different formats for representing telemetry messages may introduce additional overhead associated with data serialization, encoding, and syntactic parsing during their processing by the server system.

In this work, an experimental study of the influence of different GNSS message transmission formats on the delay of HTTP request processing in a server application was conducted. To conduct the experiment, a software testbed was developed that simulates the process of transmitting telemetry data from a client device to the server side of the system. An experimental procedure and data processing scheme were developed. The software testbed was implemented in the form of two components, namely: a web service based on the Django framework and a client module for generating HTTP requests using the Python requests library.

Within the scope of the study, a comparative analysis of several methods for transmitting telemetry messages was performed, including raw NMEA, application/json, application/x-www-form-urlencoded, multipart/form-data, and the experimental text-based format TOON. For each transmission method, a series of experimental measurements of HTTP request processing time was carried out, forming a statistical sample consisting of 300 observations. The results were processed using methods of mathematical statistics, including the calculation of the sample mean, minimum and maximum values, and the sample standard deviation. A study was conducted to determine the correspondence of the samples to known probability distribution laws.

The obtained results show that the transmission of raw NMEA messages provides the lowest request processing latency, since it does not require additional serialization operations or structural data transformations. The JSON and TOON formats demonstrate similar latency values, whereas the application/x-www-form-urlencoded and multipart/form-data formats introduce greater processing overhead. The results of this study can be applied in the design of IoT-based navigation monitoring systems where minimizing telemetry transmission delay and ensuring efficient processing of GNSS messages are important requirements.

Keywords: GNSS, NMEA, IoT, telemetry, HTTP, JSON, data transmission, transmission latency, navigation monitoring, mathematical statistics.

Дата першого надходження статті до видання: 21.03.2026

Дата прийняття статті до друку після рецензування: 20.04.2026

Дата публікації (оприлюднення) статті: 19.05.2026