

ОСОБЛИВОСТІ ОБМІНУ ПОВІДОМЛЕННЯМИ В РЕЖИМІ РЕАЛЬНОГО ЧАСУ

Черних І.А., Лановий О.Ф.

e-mail: ihor.chernykh@nure.ua, oleksiy.lanovyuy@nure.ua

Харківський національний університет радіоелектроніки, каф. ПІ
м. Харків, Україна

This research provides a comparative analysis of WebSocket and Long Polling technologies for real-time message exchange in front-end applications. The study investigates architectural features, performance characteristics, error handling mechanisms, and reliability of both approaches. Research findings indicate that WebSocket provides better performance for systems with high-frequency updates, while Long Polling offers advantages in compatibility and implementation simplicity for less demanding scenarios.

Виклики, що пов'язані з організацією ефективної обробки даних в реальному часі, стають критично важливими в умовах стрімкого розвитку подій. З розвитком односторінкових додатків (SPA) та зростанням вимог до інтерактивності традиційна модель запит-відповідь (HTTP Request-Response) часто не відповідає сучасним вимогам до швидкості оновлення даних. Це спонукає розробників звертатися до альтернативних технологій, таких як WebSocket та Long Polling, які пропонують різні підходи до організації обміну повідомленнями між клієнтом та сервером.

Актуальність дослідження зумовлена необхідністю формування чітких критеріїв та рекомендацій щодо вибору оптимального підходу до організації обміну повідомленнями у Front-end додатках. Особливої важливості це питання набуває в контексті розробки систем, які потребують обробки великої кількості подій у реальному часі, таких як чати, системи сповіщень, онлайн-ігри та платформи для колаборації.

WebSocket забезпечує двонаправлений повнодуплексний канал зв'язку через TCP-з'єднання, дозволяючи серверу активно надсилати дані клієнту без додаткових запитів. Long Polling базується на стандартному HTTP-протоколі та імітує push-повідомлення через механізм тривалих запитів, що робить його більш сумісним з існуючою інфраструктурою. Вибір між цими технологіями залежить від багатьох факторів: вимог до швидкодії, масштабованості системи, специфіки даних, що передаються, та обмежень інфраструктури.

Метою роботи є проведення порівняльного аналізу ефективності використання технологій WebSocket та Long Polling для обробки сповіщень на стороні Front-end додатків, визначення їх переваг та недоліків у різних сценаріях використання.

Аналіз архітектурних особливостей показав, що WebSocket працює за принципом встановлення початкового HTTP рукоштовування з подальшим переходом з'єднання у режим постійного повнодуплексного каналу

зв'язку. Це забезпечує можливість обміну даними в обох напрямках без створення нових запитів та включає вбудовані механізми контролю стану з'єднання через ping/pong фрейми.

Long Polling, в свою чергу, базується на утриманні HTTP-запиту відкритим до появи нових даних або спрацювання тайм-ауту. При появі даних сервер відповідає клієнту, після чого клієнт одразу формує новий запит, забезпечуючи постійний канал комунікації. Такий підхід імітує двосторонню комунікацію, хоча він базується на стандартному HTTP-запиті.

З метою проведення детального аналізу обох підходів при організації одночасної обробки значної кількості повідомлень, в роботі було виконано порівняльний аналіз продуктивності, який довів, що WebSocket забезпечує значно менші затримки при передачі даних. При тестуванні з навантаженням у 1000 одночасних користувачів, середня затримка для WebSocket складала 50-100 мс, тоді як для Long Polling цей показник становив 200-300 мс. Ця різниця особливо помітна в додатках, що вимагають частого обміну даними.

WebSocket демонструє нижчі накладні витрати після встановлення з'єднання, оскільки не потребує багаторазового створення нових HTTP-запитів. Long Polling генерує більше службового трафіку через необхідність постійної передачі HTTP-заголовків при кожному запиті. При аналізі ефективності використання серверних ресурсів виявлено, що WebSocket зазвичай ефективніше використовує серверні ресурси завдяки підтримці єдиного довготривалого з'єднання на відміну від множинних з'єднань при Long Polling.

З точки зору масштабованості, Long Polling має перевагу в легкості інтеграції з існуючою HTTP-інфраструктурою та можливості використання стандартних механізмів балансування навантаження. WebSocket, хоча і забезпечує кращу продуктивність, може вимагати спеціальних рішень для масштабування та підтримки постійних з'єднань.

Особливу увагу в дослідженні приділено механізмам обробки помилок. WebSocket має вбудовані механізми виявлення та обробки помилок на рівні протоколу. Коли відбувається збій з'єднання, WebSocket API генерує подію onClose, яка містить код помилки та опис причини закриття з'єднання, що дозволяє точно визначити характер проблеми. Long Polling використовує стандартні HTTP-коди відповіді та часові обмеження запитів, що вимагає від розробника самостійної реалізації логіки відновлення з'єднання.

Проведені дослідження виявили важливу відмінність, пов'язану з підходом до виявлення розриву з'єднання. WebSocket використовує механізм heartbeat-повідомлень для активного моніторингу стану з'єднання, тоді як Long Polling покладається на тайм-аути HTTP-запитів, що може призводити до більшої затримки у виявленні проблем із з'єднанням.

В результаті проведеного дослідження було розроблено методологію порівняльного аналізу, яка включає оцінку продуктивності, надійності та масштабованості технологій. Основними критеріями оцінки є латентність, мережеві накладні витрати, пропускна здатність системи, ефективність використання серверних ресурсів, надійність при мережевих збоях, складність реалізації та сумісність з різними платформами.

Для проведення тестування спроектовано архітектуру тестових додатків з використанням React та TypeScript. Архітектура базується на модульній структурі, де мережевий шар реалізується по-різному для WebSocket та Long Polling версій, при цьому зберігаючи однаковий інтерфейс взаємодії з іншими компонентами системи. Система тестування включає генератор навантаження, емулятор мережевих умов та систему моніторингу для збору метрик у реальному часі.

Для забезпечення об'єктивності результатів всі тести проводяться в ідентичних умовах для обох технологій. Система тестування розгортається в ізольованому середовищі, що мінімізує вплив зовнішніх факторів на результати вимірювань. Це дозволяє отримати надійні дані для порівняльного аналізу ефективності WebSocket та Long Polling у різних сценаріях використання.

Результати проведених досліджень показують, що вибір між WebSocket та Long Polling повинен базуватися на конкретних вимогах проекту. WebSocket є більш ефективним для систем реального часу з високою частотою обміну даними, онлайн-ігор, чатів та інтерактивних додатків, де критичною є мінімальна затримка. Long Polling більш підходить для систем сповіщень з нерегулярними оновленнями, додатків з обмеженнями на використання WebSocket та середовищ, де важлива сумісність з широким спектром клієнтських платформ.

Проведені дослідження можуть служити основою для подальшої практичної реалізації та експериментального порівняння обох технологій у різних умовах використання. Розроблені методологія та архітектура тестових додатків дозволять отримати об'єктивні дані про ефективність WebSocket та Long Polling у конкретних сценаріях застосування та сформувані обґрунтовані рекомендації щодо їх використання.

Список використаних джерел:

1. Wang V., Salim F., Moskovits P. The Definitive Guide to HTML5 WebSocket. Apress, 2021. 164 p.
2. Pimentel V., Nickerson B. G. Web-Based Real-Time Communications: Past, Present and Future. Information Systems Research, 2023. Vol. 34(1). P. 289-308.
3. Мазурова О.О., Набока А.В., Широкопетлева М.С. Порівняльний аналіз технологій реального часу у веб-застосунках. Вісник ХНУРЕ. Серія: Інформаційні технології. 2023. № 2(6). С. 31-42.