

ПОРІВНЯННЯ ЧАСУ ЗАВАНТАЖЕННЯ СТОРІНКИ ЗА ПРОТОКОЛАМИ HTTP1 та HTTP2

Перевозніков Сергій, Горобець Юрій

Вінницький національний технічний університет

Анотація

В даній доповіді, проведено порівняння протоколів HTTP1 і HTTP2, як з точки зору функціональних можливостей, так і з точки зору часу завантаження сторінки. Показано, що HTTP2 має менший час завантаження сторінки, ніж HTTP1. При використанні протоколу HTTP2 час завантаження сторінки залишається на 15% нижче, ніж HTTP1 навіть при збільшенні мережевої затримки. Крім того, встановлено, що протокол HTTP2 є більш стійким до втрати пакетів, ніж HTTP 1.

Abstract

This report compares HTTP1 and HTTP2, both in terms of functionality and page load time. HTTP2 is shown to have less page load time than HTTP1. When using HTTP2, page load time is 15% lower than HTTP1, even with an increase in network latency. In addition, HTTP2 is found to be more resilient to packet loss than HTTP 1.

Середній розмір веб-сторінки за останні 3 роки збільшився майже вдвічі [1]. Основна частина збільшення трафіку - це підвищення роздільної здатності відео та складність додатків для електронної комерції. Крім того, є прихована частина, яка виходить з оптимізації, що виконують веб-розробники. В якості прикладу, сегментування доменів (зберігання контенту в декількох піддоменах для збільшення числа паралельних підключень) збільшує пропускну здатність, доступну між браузером та веб-додатком, але додає багато накладних витрат на підключення, так як відбувається дублювання заголовків, файлів cookie, CSS-файлів та файлів JavaScript або апроксимації функції для прогнозування завантаженості веб-серверу [2].

HTTP1 був розроблений в 1990 році для перегляду HTML сторінок у веб-браузерах. Сучасні веб-додатки мають більш високі вимоги в плані інтерактивності і їм необхідно адаптуватися до надзвичайно мінливих мережевих умов. Однак при завантаженні веб-сайту, сторінка вимагає декількох запитів, ці запити не можуть бути дійсно розпаралелені. Навіть при конвеєризації HTTP/1.1, відповідь на запит N повинна чекати до тих пір, поки надсилається запит $n-1$. Блокування запиту є відомою проблемою HTTP протоколу.

Щоб подолати цю проблему, Google випробував новий протокол названий SPDY [3]. Ця ідея була повторно використана IETF, для використання в специфікаціях HTTP2. HTTP2 транспортує заголовки HTTP і дані в окремі фрейми. Стиснення заголовків HTTP2 оптимізує заголовки, в реальному часі, вирішуючи проблему повторень. Як і HTTP/1.1-конвеєризація, так і HTTP2 підтримують мультиплексування запитів в одному і тому ж TCP з'єднанні. У HTTP2 запити та відповіді пронумеровані. Це дозволяє чергувати відповіді паралельно і вирішує проблему блокування запитів. Також HTTP2 як і HTTP1.1 дозволяє використання через TLS шифрування або RSA шифрування [4]. В даній роботі, було проведено тестування продуктивності без використання TLS шифрування, через TCP протокол. При тестуванні використовувались mod_http2 та nhttp2 модулі.

Для порівняння продуктивності HTTP2 та HTTP1 виміряно час, необхідний для завантаження тієї ж середньої веб-сторінки (СВС) з HTTP1 і HTTP2, використовуючи одне і те ж апаратне, програмне забезпечення та характеристики мережі.

Перерозподіл компонентів середньої веб-сторінки (вміст та скрипти) для 16 доменів показано у таблиці 1.

Таблиця 1 – Планування запитів

| Домен\тип файлу | html | css | js | Img | flash | Інше |
|-----------------|------|-----|----|-----|-------|------|
| site1.com | 1 | 1 | 1 | | | |
| site2.com | 1 | 1 | 1 | | | |
| site3.com | | | 4 | | | 1 |
| site4.com | | | 4 | | | |
| site5.com | | | 4 | 3 | | |
| site6.com | 1 | 1 | | | | |
| site7.com | 1 | 1 | | | | |
| site8.com | 1 | 1 | 1 | | | 1 |
| site9.com | 1 | 1 | 1 | | | |
| site10.com | 1 | | | | | 4 |
| site11.com | | | 4 | 3 | | 1 |
| site12.com | 1 | 1 | | | | |
| site13.com | 1 | | | | 1 | |
| site14.com | | | | 10 | | |
| site15.com | | | | 10 | | |
| site16.com | | | | 10 | | |

Домен 14, 15 та 16 отримують половину запитів для статичних зображень. 25% доменів отримують більше ніж 7 запитів.

В обох випадках, при використанні протоколів HTTP1 і HTTP2, клієнт відкриває унікальне TCP-з'єднання з кожним доменом і негайно надсилає всі HTTP запити до цього домену. Щоб уникнути будь-яких упереджень, клієнт деактивує будь-яку оптимізацію на рівні протоколу HTTP2 (HTTP2 PUSH_PROMISE та PRIORITY відключено).

Тестувальна платформа складається з HTTP-сервера, а HTTP-клієнт запитує сторінку на сервері. Важливо зауважити, що сервери реалізовані як віртуальні машини на одному комп'ютері.

Для проведення тестування, було використано такі інструменти:

- **HTTP-клієнт** повинен мати можливість відправляти і HTTP1 і HTTP2 незашифрований трафік, було вибрано h2load.
- **DNS сервер** - локальний DNS-сервер вирішує ім'я 16 доменів веб-серверів (domain1.com до домен16.com). Він працює на тій же машині, що і HTTP сервери.
- **Мережа** - Клієнти та сервери підключені напряму в мережі Ethernet 1Gbps. TCP розмір - це завжди 65535 байт, тоді як MTU - 1500 байт. Порушення мережі (затримка та втрачені пакети) згенеровано командою Linux tc.
- **Веб-сервер** - повинен підтримувати і HTTP1 і HTTP2. Використано HTTP-сервер Apache версії 2.4.17 з модулем mod_http2.
- Клієнти працюють на ОС Linux Debian. Кожен домен має власний сервер Apache. Вони запущені в VirtualBox (VM), яка розміщена на ПК з 8Gb оперативної пам'яті та з процесором Intel Core i7 з 8 ядрами.

Результати тестування.

На рисунку 1 зображено варіацію середнього часу завантаження сторінки використовуючи HTTP1 та HTTP2 протокол, в залежності від збільшення затримки у мережі. Середнє значення обчислюється за всіма вимірюваннями для всіх значень втрати пакетів.

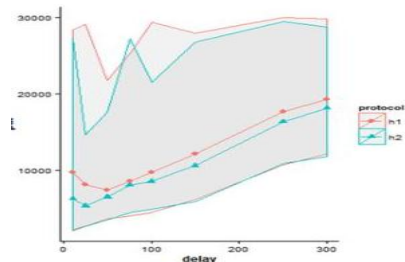


Рисунок 1. Середній час завантаження сторінки використовуючи HTTP1 та HTTP2 протокол.

На рисунку 2 представлено співвідношення часу завантаження сторінки використовуючи HTTP1 та HTTP2 протокол для втрати 1% пакетів. Коли затримка в мережі 10 мс, HTTP2 завантажує сторінку в 2 рази швидше, ніж HTTP/1.1.

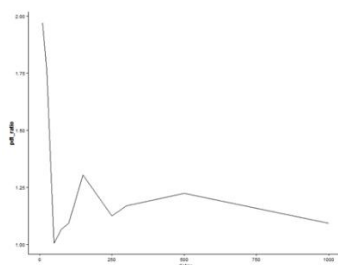


Рисунок 2. співвідношення часу завантаження сторінки для втрати 1% пакетів.

На рисунку 3, зображено час завантаження сторінки в залежності від втрати пакетів, для протоколів HTTP/1.1 та HTTP/2.

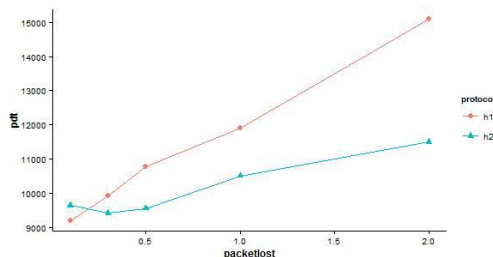


Рисунок 3. час завантаження сторінки в залежності від втрати пакетів, для протоколів HTTP/1.1 та HTTP/2.

Рисунок показує, що для протоколів HTTP/1.1 та HTTP/2 час завантаження сторінки збільшується в міру збільшення втрати пакетів, що є логічним. Крім того, показано, що швидкість завантаження сторінки по протоколу HTTP2 відносно протоколу HTTP1 зменшується під час збільшення втрати пакетів. Для підтвердження цього результату ми обчислюємо середній показник часу завантаження сторінки для протоколів HTTP1 та HTTP2 для фіксованої втрати пакетів для всіх вимірювань і всіх значень затримки мережі. Результат - представлений на рисунку 4. Це показує, що HTTP2 є більш стійким до втрати пакетів, ніж HTTP1.

Висновок

У роботі проведено порівняння HTTP1 та HTTP2 протоколів з точки зору функціональності, зокрема щодо проблеми блокування заголовків HTTP1. Виконано дослідження часу завантаження сторінки використовуючи протоколи HTTP1 та HTTP2 за

допомогою одного і того ж апаратного, програмного забезпечення та характеристик мережі, використовуючи затримку мережевих пакетів та втрат пакетів. Виходячи з отриманих результатів, можна зробити висновок, що НТТР2 завжди має менший час завантаження сторінки, ніж НТТР1. Час завантаження сторінки НТТР2 збільшується повільніше ніж НТТР1, під час збільшення мережевої затримки, залишаючись на 15% нижчим порівняно з НТТР1. Час завантаження сторінки НТТР2 також збільшується повільніше ніж НТТР1, протягом збільшення втрати пакетів. В останньому випадку показано, що два протоколи мають різну поведінку, підкреслюючи набагато кращу стійкість НТТР2 до втрати пакетів.

Отже, результати підтверджують переваги переведення веб-серверів на використання протоколу НТТР2.

Список використаних джерел

1. Webpages Are Getting Larger Every Year, and Here's Why it Matters [Електронний ресурс]. – Режим доступу: <https://royal.pingdom.com/webpages-are-getting-larger-every-year-and-heres-why-it-matters/>
2. Озеранський В.С., Мартиненко М.О., «Застосування апроксимації функції при прогнозуванні завантаженості сервера» Матеріали конференції «XLVIII Науково-технічна конференція підрозділів Вінницького національного технічного університету (2019)», Вінниця, 2019. [Електронний ресурс]. Режим доступу: <https://conferences.vntu.edu.ua/index.php/all-fitki/index/pages/view/view/zbirn2019> Дата звернення: грудень. 2019
3. SPDY protocol [Електронний ресурс]. – Режим доступу <https://tools.ietf.org/html/draft-mbelshe-httpbis-spy-00>
4. Перевозніков С.І. Криптографічний захист цифрової інформації / М.Л. Благодир, С.І. Перевозніков // Тези XLV науково-технічної конференції професорсько-викладацького складу, співробітників та студентів університету, 21–23 березня, 2018. – Вінниця: ВНТУ, 2018