

---

---

## МЕТОДИ ТА СИСТЕМИ ОПТИКО-ЕЛЕКТРОННОЇ І ЦИФРОВОЇ ОБРОБКИ ЗОБРАЖЕНЬ ТА СИГНАЛІВ

---

---

УДК 004.738.5:004.415.2

А.О. ПРИЛПА, Г.Є.ФІЛАТОВА

### АДАПТИВНІ ПОЛІТИКИ ОПТИМІЗАЦІЇ ТА ДЕГРАДАЦІЇ КОНТЕНТУ ДЛЯ ВЕБЗАСТОСУНКІВ В УМОВАХ НЕСТАБІЛЬНИХ МЕРЕЖ

*Національний технічний університет “Харківський політехнічний інститут”  
61002, вул. Кирпичова 2, м. Харків, Україна a.prilipa82@gmail.com*

**Анотація.** Розглянуто проблему забезпечення ефективної роботи комп’ютерних систем у вигляді PWA вебзастосунків в умовах обмежених і нестабільних мереж, де поряд із швидкістю критичною є завершеність операцій. Запропоновано підхід на основі адаптивних політик, у якому стан мережі визначає вибір стратегій для шляхів читання даних і запису операцій та керує кешуванням, офлайн-режимом і відкладеним виконанням. Проведено експериментальне порівняння базового варіанта та варіанта з адаптивними політиками за сценаріями з різними мережевими профілями. Отримані результати демонструють зниження затримок, зменшення мережевої вартості та підвищення завершеності операцій, зокрема в офлайн-умовах, що підтверджує ефективність запропонованого підходу для комп’ютерних систем, орієнтованих на роботу в умовах мережевих обмежень.

**Ключові слова:** комп’ютерна система; вебзастосунок; PWA; адаптивні політики; мережеві обмеження; кешування; офлайн-режим; завершеність операцій.

**Abstract.** The paper addresses the problem of ensuring the effective operation of computer systems implemented as PWA web applications under constrained and unstable network conditions, where operation completability is critical alongside performance. An adaptive policy-based approach is proposed, in which the network state determines the selection of strategies for data read paths and operation write paths, and governs caching, offline mode, and deferred execution. An experimental comparison of the baseline variant and the variant with adaptive policies was conducted using scenarios with different network profiles. The obtained results demonstrate reduced latencies, lower network cost, and improved operation completability, particularly under offline conditions, confirming the effectiveness of the proposed approach for computer systems designed to operate under network constraints.

**Keywords:** computer system; web application; PWA; adaptive policies; network constraints; caching; offline mode; operation completability.

**DOI: 10.31649/1681-7893-2026-51-1-7-17**

#### ВСТУП

З розвитком вебзастосунків змінюється характер взаємодії користувача з вебсистемами, що висуває нові вимоги до їх надійності та поведінки в умовах нестабільної мережі. У таких системах дедалі більшого значення набуває задача завершення операції, а не лише відображення сторінки. У робочих умовах можливі переривання мережі, високі затримки, втрати пакетів. У таких умовах типова оптимізація, орієнтована на Web Vitals, не гарантує завершення дії користувача. Додатково виникає розрив між метриками на рівні сторінки та результатом на рівні сценарію, оскільки частина критичних подій відбувається після первинного рендеру. Для прикладних вебзастосунків важливо забезпечити керовану поведінку для шляху читання даних та шляху запису операцій, включно з повторними спробами, постановкою в чергу та відкладеною синхронізацією. Тому цільова оцінка має спиратися на завершеність операцій та час до досягнення консистентного стану, а не лише на показники інтерактивності чи середні затримки запитів.

---

---

## МЕТОДИ ТА СИСТЕМИ ОПТИКО-ЕЛЕКТРОННОЇ І ЦИФРОВОЇ ОБРОБКИ ЗОБРАЖЕНЬ ТА СИГНАЛІВ

---

---

Service Worker розглядається як програмований мережевий посередник, що перехоплює запити та дозволяє реалізовувати кероване кешування і офлайн-поведінку на рівні вебзастосунку [1]. На практиці для шляху читання даних найчастіше застосовуються стратегії з пріоритетом кешу (cache-first), з пріоритетом мережі (network-first) та стратегія повернення кешованих даних із фоновою ревалідацією (stale-while-revalidate). Вони задають пріоритет мережі або кешу і можуть комбінуватися з попереднім кешуванням та маршрутизацією [2], однак не враховують змінний стан мережі та не забезпечують адаптивного керування поведінкою застосунку. Формально ці механізми спираються на моделі HTTP кешування та правила валідності і протермінування відповідей [3].

Дослідження PWA показують, що кешування істотно впливає на час завантаження та споживання ресурсів. Зокрема, оцінено вплив стану кешу на час завантаження сторінки для реальних PWA, що підтверджує значущість кеш-політик [4]. Окремо розглянуто вплив Service Worker на енергоефективність у різних мережевих умовах, що важливо для постановок з обмеженими мережами та мобільними пристроями [5]. Водночас у прикладних сценаріях ці стратегії зазвичай залишаються статичними, що ускладнює адаптацію до переходів між мережевими станами та контроль вартості повторів.

Для шляху запису даних поширеним є підхід локальної черги, коли операції запису накопичуються локально, а потім відправляються при відновленні зв'язку. Стандартизованою основою для відкладеної доставки виступає Background Sync, який дозволяє реєструвати задачі синхронізації у Service Worker і виконувати їх у фоновому режимі, коли браузер вважає умови прийнятними [6]. Для сценаріїв з регулярною актуалізацією розглядається Periodic Background Sync, що задає періодичні фонові синхронізації з урахуванням мережевих умов [7].

Критичними аспектами для офлайн-черг є семантика доставки «щонайменше один раз» (at-least-once), потреба в ідемпотентності серверних операцій, а також дедуплікація і кореляція спроб. Для відтворюваних експериментів і аналізу потрібні механізми телеметрії, що пов'язують подію, постановку в чергу, повторні відправки та фінальне підтвердження. Окремим практичним питанням є керування темпом синхронізації, яке пов'язане з обмеженням повторів та бюджетами трафіку [8].

Для реалізації адаптивної поведінки необхідно є оцінка стану мережі. Як джерело оцінки мережі для вебзастосунків часто використовується браузерний інтерфейс Network Information API (об'єкт navigator.connection). Попри наявність практичної реалізації цього підходу в частині браузерів, його застосування залишається обмеженим через неоднорідну підтримку, відмінності між реалізаціями та обмежену точність окремих показників у реальних умовах. Тому дані navigator.connection доцільно розглядати не як самодостатнє джерело істини, а як один із вхідних сигналів для оцінки стану мережі [9, 10]. Для підвищення точності оцінки мережі в роботі запропоновано власний клієнтський профайлер мережевих характеристик, який виконується безпосередньо у браузері та поєднує стандартні сигнали Web Performance API з легковаговою моделлю TinyML [9]. Це дає змогу локально класифікувати стан мережевого з'єднання, зменшити залежність від обмежених браузерних індикаторів і забезпечити основу для подальшої адаптації поведінки вебзастосунку. У доповіді про Network Quality Estimation у Chrome описано підхід до оцінювання якості мережі та експозицію оцінок через API, що підтримує адаптацію поведінки під умови обмеженого мережевого з'єднання [11]. Для коректного вимірювання впливу мережі на метрики вебзастосунків також показано, що результати можуть суттєво залежати від браузера, а мережеві затримки мають прямиий вплив на часові метрики [12].

Дослідження вебзастосунків на великих вибірках показують, що досвід користувача визначається не лише середніми затримками. Це робить суто мережеві метрики для оцінювання результату недостатніми [13]. Окремо підкреслюється потреба в коректних, масштабованих методах вимірювання Quality of Experience (QoE). Метрики часу завантаження можуть не давати повної оцінки QoE. Для ширшого аналізу потрібні процедури збору даних та валідації метрик [15 - 18].

Попри наявність окремих рішень у сфері PWA, існуючі підходи залишаються фрагментованими та не забезпечують комплексної адаптації до змінного стану мережі. Зокрема, відсутні узгоджене керування шляхами читання і запису даних, системний контроль повторних спроб та їх вартості, а також метрики, що дозволяють оцінювати завершеність операційних сценаріїв в умовах нестабільного або відсутнього з'єднання. У роботі запропоновано перехід від «успіху HTTP запиту» до «успіху операції», тобто до метрик завершеності сценарію. Такий підхід узгоджується з практиками QoE, де вимірювання пов'язуються з рівнем користувацької задачі, а не лише з транспортними подіями протоколів [12, 14]. Це створює підставу для нових метрик, у яких об'єктом оцінювання є завершення операції з урахуванням постановки в чергу та відкладеної синхронізації.

## ПОСТАНОВКА ЗАДАЧІ

У роботі розглядається вебзастосунок для реєстрації та контролю відвідувачів подій. Шлях читання даних включає завантаження списків учасників, деталей події та довідкових даних. Шлях запису операцій включає операцію реєстрації присутності, що створює або оновлює запис на сервері. Для операцій запису істотними є повтори, дедуплікація та консистентність стану після відновлення мережі. Для оцінювання ефективності підходу розглядаються два варіанти реалізації. Варіант V0 – це класичний вебзастосунок без Service Worker. Він використовує прямі мережеві запити з типовими таймаутами клієнта. Кешування та повтори виконуються лише на рівні браузера і HTTP. Варіант V1 – це PWA з Service Worker та механізмом застосування політик. Реалізація застосовує адаптивні стратегії, офлайн-чергу, механізм обмеження марних спроб та бюджетоване попереднє завантаження.

Метою роботи є підвищення завершеності операцій в обмежених і нестабільних мережах при одночасному обмеженні вартості трафіку та кількості марних спроб. Для досягнення поставленої мети необхідно вирішати такі задачі:

- 1) формалізувати стан мережі та політики адаптації у вигляді конфігурації policy map;
- 2) інтегрувати шлях читання даних, шлях запису операцій, механізм обмеження марних спроб і деградацію контенту;
- 3) визначити метрики вартості та відновлення для сценаріїв дослідження;
- 4) побудувати відтворюваний протокол експериментів і експорт артефактів;
- 5) виконати порівняння V0 та V1 у сценаріях і підготувати основу для аналізу.

Для перевірки сформульованих задач введено такі гіпотези:

- адаптація на основі політик зменшує затримки виконання сценаріїв у V1 ;
- адаптація на основі політик у V1 підвищує завершеність сценаріїв порівняно з V0;
- у станах 2G та офлайн застосування постановки в чергу у V1 підвищує завершеність на рівні інтерфейсу;
- керовані таймаути та механізм обмеження марних спроб у V1 зменшують вартість марних спроб у обмеженій мережі.

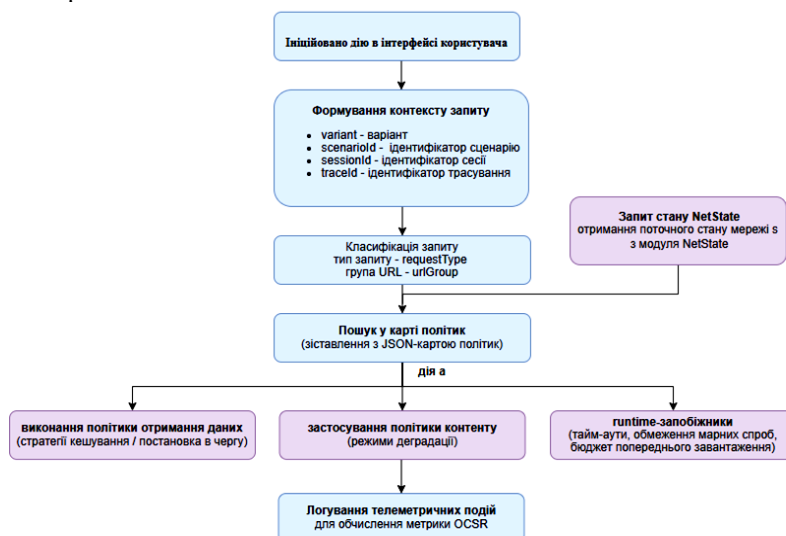
## МОДЕЛЬ АДАПТИВНИХ ПОЛІТИК ТА МЕХАНІЗМ ЇХ ЗАСТОСУВАННЯ

У роботі запропоновано модель адаптивного керування поведінкою вебзастосунку на основі політик, що визначають стратегії обробки запитів залежно від стану мережі. Ключовим елементом запропонованої моделі є відображення:

$$\pi : S \times R \rightarrow A,$$

де  $S$  – множина станів мережі;  $R$  – простір класифікації запитів;  $A$  – простір дій.

Загальний процес застосування політик, включаючи формування контексту, класифікацію запиту та вибір дії, наведений на рис. 1.



Рисуюнок 1 – Графічна схема робочого процесу рушія політик

# МЕТОДИ ТА СИСТЕМИ ОПТИКО-ЕЛЕКТРОННОЇ І ЦИФРОВОЇ ОБРОБКИ ЗОБРАЖЕНЬ ТА СИГНАЛІВ

Простір станів мережі формується на основі класифікації мережевих умов і задається змінною  $NetState$ , що відповідає  $s \in S$ . Переходи між станами  $NetState$  проілюстровано на рис. 2. Класифікатор мережевого стану, побудований на основі клієнтського TinyML-профайлера, формує неперервні оцінки стану мережі. Для зменшення впливу короточасних коливань їх попередньо стабілізують. Множина станів визначається як  $S = \{offline, 2G, 3G, 4G, broadband\}$ , причому стани впорядковані за зростанням якості зв'язку (від  $offline$  до  $broadband$ ). Для забезпечення стабільності переходів застосовується обмеження частоти оновлення та мінімального часу перебування у стані.

Формально простір дій  $A$  задається як  $A = A_{fetch} \times A_{content} \times \Theta$ , де  $A_{fetch} = \{cacheonly, cacheFirst, networkFirst, staleWhile Re validate, tryThenEnqueue, enqueueOnly\}$  – множина стратегій отримання даних;  $A_{content} = \{full, lite, ultralite\}$  – множина режимів деградації контенту;  $\Theta$  – простір параметрів, які визначають таймаути, режим черги та параметри механізму обмеження марних спроб, а також бюджет попереднього завантаження і режим деградації контенту. Тоді для заданого контексту запиту дія визначається як  $a = \pi(s, r) \in A$ .

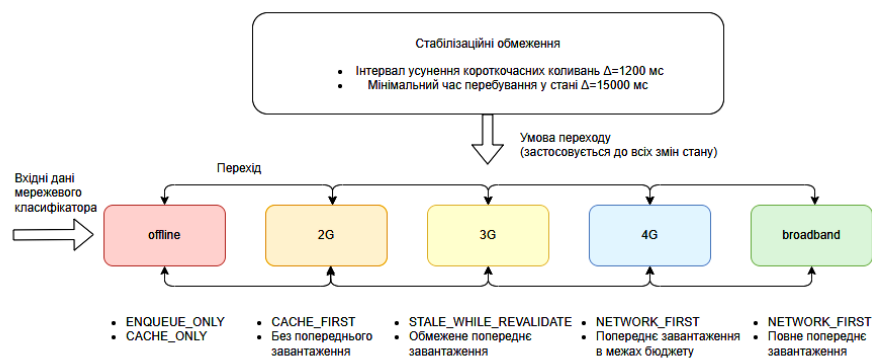


Рисунок 2 – Графічна схема  $NetState$

Простір класифікації запитів  $R$  формується як множина ознак запиту та включає тип запиту  $requestType$ , групу ресурсу  $urlGroup$ , а також додаткові атрибути контексту (наприклад, сценарій або варіант виконання), що використовуються для вибору політики.

Механізм застосування політик  $\pi$  включає формування контексту запиту, його класифікацію, визначення стану мережі та вибір відповідної дії згідно з картою політик (табл. 1).

Таблиця 1 – Фрагмент карти політик (policy map) у табличній формі

Стан мережі $s \in S$	Клас запиту / умова $r \in R$	Дія політики $a \in A$	Параметри $\theta \in \Theta$
4G	API_READ_EVENTS	STALE_WHILE_REVALIDATE	ttl=120s; staleBudget=600s
2G	API_READ_EVENTS	CACHE_FIRST	ttl=600s; staleBudget=86400s
offline	API_READ_EVENTS	CACHE_ONLY	-
4G	API_WRITE_CHECKIN	TRY_THEN_ENQUEUE	timeout=3000ms
3G	API_WRITE_CHECKIN	TRY_THEN_ENQUEUE	timeoutMs=2000; preferEnqueue=true
offline	API_WRITE_CHECKIN	ENQUEUE_ONLY	-
4G	MEDIA_AVATAR	CACHE_FIRST	ttl=604800s
2G	MEDIA_AVATAR	CACHE_ONLY	
4G	screen=attendeesList	content: mode=full	fields=full; pageSize=50; search=debounced; autoRefresh=true; avatars=lazy
3G	screen=attendeesList	content: mode=lite	fields=lite; pageSize=30; search=debounced; autoRefresh=false; disableAnimations=true; avatars=tapToLoad
offline	screen=attendeesList	content: mode=ultraLite	fields=min; pageSize=15; source=cacheOnly

Модель застосовується окремо для шляху читання даних і шляху запису операцій, які мають

---

---

## МЕТОДИ ТА СИСТЕМИ ОПТИКО-ЕЛЕКТРОННОЇ І ЦИФРОВОЇ ОБРОБКИ ЗОБРАЖЕНЬ ТА СИГНАЛІВ

---

---

різні вимоги до продуктивності та надійності. Для шляху читання політика визначає стратегію отримання даних залежно від стану мережі. У сприятливих умовах пріоритет надається актуальності даних (*network-first, stale-while-revalidate*), у обмежених – використанню кешу та зменшенню затримок, а в стані *offline* застосовується стратегія *cache-only*. На відміну від шляху читання, шлях запису орієнтований на забезпечення завершеності операцій навіть за нестабільного або відсутнього з'єднання. Для цього використовується локальна черга операцій, у яку додаються запити з подальшою відкладеною синхронізацією. Операція вважається прийнятою в момент її постановки в чергу, а її завершення досягається після успішної синхронізації із сервером.

Динаміка черги може бути наближено описана співвідношенням

$$\frac{dQ(t)}{dt} \approx \lambda(t) - \mu(s, a, t),$$

де  $\lambda(t)$  – інтенсивність надходження операцій;  $\mu(s, a, t)$  – інтенсивність їх обробки. Стійкість системи забезпечується за умови  $\mu(s, a, t) > \lambda(t)$ .

Для обмеження марних спроб використовується механізм порогового блокування запитів, який переводить систему в режим обмеження активності при перевищенні допустимого рівня помилок. Попереднє завантаження (*prefetch*) виконується за умови дотримання бюджету, що залежить від стану мережі, що дозволяє обмежити додаткове навантаження. Для узгодження якості інтерфейсу та вартості запитів застосовується деградація контенту, яка визначає обсяг переданих даних залежно від стану мережі.

Таким чином, запропонована модель забезпечує узгоджене керування обробкою запитів, що дозволяє підвищити завершеність операцій і зменшити витрати ресурсів у нестабільних мережевих умовах.

Для комплексного оцінювання ефективності запропонованої моделі введено систему метрик, яка охоплює затримку, вартість та надійність. Для кожного класу запитів фіксуються затримки  $p_{50}$ ,  $p_{95}$ ,  $p_{99}$ , виміряні як інтервал від моменту ініціювання запиту до отримання відповіді або виникнення помилки. Відповіді, надані з кешу, реєструються окремо як *cacheLatency*. Це дає змогу ізолювати вплив стратегії кешування від мережевої затримки.

Для оцінювання мережевої вартості використовується вектор показників  $\vec{C}(\pi) = (\text{bytes}, \text{requests}, \text{retries})$ , де *bytes* – обсяг переданих даних; *requests* – кількість мережевих запитів; *retries* – кількість повторних спроб. Для порівняння варіантів застосовується скаляризована вартість

$$C_{\Sigma}(\pi) = w_b \cdot \text{bytes} + w_r \cdot \text{requests} + w_t \cdot \text{retries},$$

де  $w_b, w_r, w_t \geq 0$  – вагові коефіцієнти, причому  $w_b + w_r + w_t = 1$ , що забезпечує нормування показника вартості. Показник вартості використовується для аналізу компромісів між надійністю та ефективністю.

На відміну від часових та ресурсних метрик, які характеризують окремі запити, для оцінювання поведінки системи на рівні сценарію використовується показник завершеності операцій. Основною метрикою для оцінювання є показник успішного завершення операцій (*Operation Completion Success Rate*):

$$OCSR = \frac{N_{\text{success}}}{N_{\text{start}}},$$

де  $N_{\text{start}}$  – кількість ініційованих операцій;  $N_{\text{success}}$  – кількість успішно завершених операцій.

Така метрика відображає результат виконання операцій з точки зору досягнення кінцевого стану. Однак у сценаріях з нестабільним з'єднанням частина операцій може бути відкладена шляхом постановки в чергу. З урахуванням цього вводиться розширена метрика на рівні інтерфейсу:

$$OCSR_{UI} = \frac{N_{\text{success}} + N_{\text{queued}}}{N_{\text{start}}},$$

де  $N_{\text{queued}}$  – кількість операцій, що були прийняті системою та поставлені в чергу для подальшої обробки.

Для оцінювання відкладеного завершення операцій після відновлення з'єднання використовується метрика успішної синхронізації у часовому вікні  $W$ :

---

---

## МЕТОДИ ТА СИСТЕМИ ОПТИКО-ЕЛЕКТРОННОЇ І ЦИФРОВОЇ ОБРОБКИ ЗОБРАЖЕНЬ ТА СИГНАЛІВ

---

---

$$OCSR_{SYNC}(W) = \frac{N_{sync\_ok}}{N_{start}},$$

де  $N_{sync\_ok}$  – кількість операцій, що були успішно синхронізовані з сервером протягом інтервалу часу  $W$  після відновлення з'єднання;  $N_{start}$  – кількість ініційованих операцій у відповідному сценарії.

Часове вікно  $W$  задає інтервал спостереження після відновлення з'єднання і визначає період, протягом якого враховується завершення відкладених операцій. Значення метрики залежить від ефективності механізму черги, параметрів синхронізації та стану мережі після відновлення. Ця метрика відображає здатність системи забезпечити завершення операцій, які були відкладені в офлайн-режимі.

Усі три метрики потребують телеметричних подій рівня окремої операції ( $op_{start}$ ,  $op_{end}$ ,  $sync_{start}$ ,  $sync_{end}$ ), пов'язаних між собою унікальним ідентифікатором  $operationId$ . Перед обчисленням метрик виконується дедуплікація повторних спроб.

Час досягнення узгодженості визначається як  $T_{consistency}^{(i)} = t_{ack}^{(i)} - t_{start}^{(i)}$ , де  $t_{ack}$  – часова мітка серверного підтвердження для операції. Час відновлення визначається як  $T_{recovery}^{(i)} = t_{recovery}^{(i)} - t_{offline}^{(i)}$ , що вимірює інтервал від моменту переходу в офлайн-стан до моменту, коли інтерфейс користувача сигналізує про успішне відновлення з'єднання.

Для узагальненого порівняння ефективності політик вводиться відносний показник, що враховує компроміс між приростом завершеності та зміною мережевої вартості:

$$E_{\pi} = \frac{\Delta OCSR}{\Delta C_{\Sigma}},$$

$$\text{де } \Delta OCSR = OCSR_{\pi_1} - OCSR_{\pi_0}; \Delta C_{\Sigma} = C_{\Sigma}(\pi_1) - C_{\Sigma}(\pi_0).$$

Цей показник дає змогу оцінити, наскільки приріст успішності завершення операцій досягається відносно зміни мережевої вартості між порівнюваними варіантами системи.

Таким чином, запропонована модель задає не лише правило вибору дії  $a = \pi(s, r)$ , а й основу для кількісного оцінювання наслідків такого вибору.

### РЕЗУЛЬТАТИ ЕКСПЕРИМЕНТАЛЬНОГО ДОСЛІДЖЕННЯ

Експериментальне дослідження спрямоване на перевірку сформульованих гіпотез щодо впливу адаптивних політик на затримку, мережеву вартість та завершеність операцій. У експерименті порівнюються базовий варіант V0 та варіант V1 із механізмом адаптивних політик. Для експериментів використано мережеві профілі, що відповідають множині станів  $S = \{offline, 2G, 3G, 4G, broadband\}$ , які впорядковані за зростанням якості зв'язку та моделюють перехід від стабільного з'єднання до його повної відсутності. Нехай  $Sc = \{Sc1, \dots, Sc6\}$  – множина сценаріїв експерименту (табл. 2). Аналіз проводився за метриками  $p_{95}$ ,  $C_{\Sigma}$ ,  $OCSR$ ,  $OCSR_{UI}$  та  $OCSR_{SYNC}(W)$ . Для кожного з варіантів (V0 та V1), кожного мережевого профілю для сценаріїв Sc1–Sc3 виконано по 5 запусків. Для сценаріїв Sc4–Sc6 також виконано по 5 запусків із заздалегідь визначеними переходами між мережевими станами.

Телеметрія збиралася для кожного запуску у вигляді узгодженого потоку подій інтерфейсу, запитів, черги та синхронізації, що забезпечує відтворюваність результатів. Для перевірки гіпотези щодо зниження затримок проаналізовано часові характеристики виконання сценаріїв у варіантах V0 і V1 (табл. 3).

Варіант V1 демонструє нижчі значення  $p_{95}$  для всіх мережевих станів і сценаріїв. Найбільший ефект зафіксовано в обмежених мережах. Для сценарію Sc1 в умовах 3G значення  $p_{95}$  зменшується з 816.12 мс до 27.43 мс, що відповідає скороченню на 96,6 %. Це зумовлено переходом від стратегії network-first до cache-first, що обслуговує запит із кешу з фоновією ревалідацією. Подібна закономірність спостерігається також у інших сценаріях: затримки шляхів запису в умовах 3G зменшуються на 78,2 % та 80,2 % відповідно. Це зумовлено поєднанням керованих тайм-аутів і обмеження марних спроб. В умовах широкосмугового з'єднання покращення у межах 27–41 %, однак залишаються суттєвими. Це пояснюється попереднім наповненням кешу. Отримані результати підтверджують гіпотезу про зниження затримок при використанні адаптивних політик.

## МЕТОДИ ТА СИСТЕМИ ОПТИКО-ЕЛЕКТРОННОЇ І ЦИФРОВОЇ ОБРОБКИ ЗОБРАЖЕНЬ ТА СИГНАЛІВ

Таблиця 2 – Перелік сценаріїв  $Sc = \{Sc1, \dots, Sc6\}$  з коротким описом кроків

Сценарій	Кроки сценарію
Sc1: Навігація та читання даних	Відкрити список подій → відкрити деталі події → відкрити список учасників
Sc2: Одна операція реєстрації	Відкрити сторінку події → відкрити список учасників → зареєструвати одного учасника → перевірити результат в інтерфейсі користувача та статус синхронізації.
Sc3: Серія операцій реєстрації	Відкрити список учасників → зареєструвати серію учасників → зафіксувати динаміку черги та час досягнення узгодженості $T_{consistency}$
Sc4: Офлайн-реєстрація та подальша синхронізація	Перейти в <i>offline</i> → зареєструвати учасника → відновити з'єднання → дочекатися доставлення операції з черги та серверного підтвердження → зафіксувати $T_{consistency}$ та $OCSR_{SYNC}(W)$
Sc5: Деградація мережі під час сесії	Почати виконання у стані 4G → виконати кроки сценарію Sc1 → здійснити перехід 4G → 2G → <i>offline</i> → 4G → повторити кроки читання даних і реєстрації учасника → зафіксувати час відновлення $T_{recovery}$ , а також поведінку механізму обмеження марних спроб і черги
Sc6: Офлайн-серія реєстрацій та відкладена синхронізація	Перейти в <i>offline</i> → відкрити список учасників → зареєструвати кількох учасників → підтвердити постановку операцій у чергу → відновити з'єднання → дочекатися пакетного доставлення операцій та підтвердження синхронізації

Для оцінювання мережевої вартості використовується вектор показників  $\vec{C}(\pi)$  та скаляризована вартість  $C_{\Sigma}(\pi)$ . Спостерігається суттєва відмінність у розподілі переданих байтів між V0 і V1 (рис. 3): у варіанті V1 значна частка запусків демонструє близький до нуля обсяг трафіку, що зумовлено застосуванням стратегій *cache-only* та блокуванням запитів механізмом обмеження марних спроб в умовах обмеженого з'єднання. Отриманий розподіл підтверджує зменшення складових мережевої вартості.

Таблиця 3 – Порівняння затримки  $p_{95}$  для V0 і V1 по сценаріях

Сценарій	Профіль	$p_{95\_V0}$ , мс	$p_{95\_V1}$ , мс	$\Delta p_{95}$ , мс	$\Delta p_{95}$ , %
Sc1	<i>broadband</i>	42.18	30.54	11.64	27.6
	4G	66.85	20.96	45.89	68.6
	3G	816.12	27.43	788.69	96.6
Sc2	<i>broadband</i>	25.73	15.2	10.53	40.9
	4G	26.21	12.77	13.44	51.3
	3G	13.07	2.85	10.22	78.2
Sc3	<i>broadband</i>	24.99	15.2	9.79	39.2
	4G	20.81	13.83	6.98	33.5
	3G	14.48	2.88	11.60	80.2

У сценаріях із нестабільною мережею під час сесії (Sc5) зафіксовано суттєве зниження мережевих витрат за збереження функціональної надійності (понад 110 кБ у середньому для сценарію Sc5). Це дає підстави стверджувати, що ефект зниження вартості не є рівномірним: для стабільних мережевих профілів він помірний, тоді як в умовах деградації мережі або за наявності механізму обмеження марних спроб – значний. Таким чином, гіпотеза щодо зниження мережевої вартості підтверджується, причому величина ефекту залежить від типу сценарію.

## МЕТОДИ ТА СИСТЕМИ ОПТИКО-ЕЛЕКТРОННОЇ І ЦИФРОВОЇ ОБРОБКИ ЗОБРАЖЕНЬ ТА СИГНАЛІВ

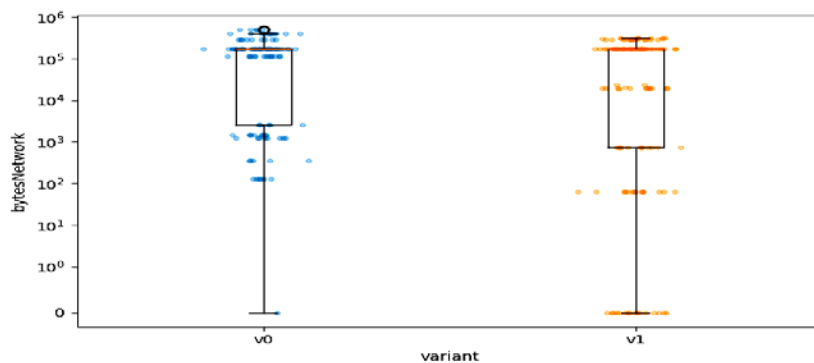


Рисунок 3 – Розподіл байтів для V0 і V1

Для оцінювання завершеності операцій проаналізовано значення метрик  $OCSR$ ,  $OCSR_{UI}$  та  $OCSR_{SYNC}(W)$  за різних мережевих умов (табл. 4). Значення  $\Delta OCSR = 1$  означає перехід від незавершених до завершених операцій.

Таблиця 4 – Значення метрик  $OCSR$ ,  $OCSR_{UI}$ ,  $OCSR_{SYNC}(W)$  для V0 і V1.

Сценарій	Профіль	$\Delta OCSR$	$\Delta OCSR_{UI}$	$\Delta OCSR_{SYNC}(W)$	$\Delta C$
Sc4	4G	0	0	0	-9874
	offline	1	1	1	-
Sc5	4G	0	0	0	-110093
	3G	0	0	0	-112482
Sc6	4G	0	0	0	-11337
	offline	1	1	1	-

За стабільних мережевих умов обидва варіанти забезпечують повну завершеність операцій ( $\Delta OCSR = 0$ ). Відмінність між варіантами за цих умов проявляється лише на рівні мережевої вартості. Зокрема, V1 зменшує  $\Delta C_{\Sigma}$  на 9.8 кБ у сценарії Sc4/4G та більш ніж на 110 кБ у сценаріях Sc5/4G і Sc5/3G. Отримані результати підтверджують, що механізм політик забезпечує ідентичну завершеність операцій за меншої мережевої вартості.

Ключовий результат отримано в офлайн-умовах, де відмінність між варіантами є принциповою. Для сценаріїв Sc4 і Sc6 в профілі *offline*:

- у V0 операції є незавершеними ( $OCSR = 0$ ,  $OCSR_{UI} = 0$ ,  $OCSR_{SYNC}(W) = 0$ ), оскільки операції не можуть бути виконані без мережевого з'єднання;
- у V1 операції гарантовано завершуються ( $OCSR = 1$ ,  $OCSR_{UI} = 1$ ,  $OCSR_{SYNC}(W) = 1$ ) завдяки відкладеному виконанню та подальшій синхронізації.

Показник  $E_{\pi}$  використовується для узагальненого оцінювання ефективності політик як співвідношення приросту завершеності до зміни мережевої вартості. Значення  $E_{\pi}$ , наведені в табл. 5, свідчать про диференційований характер ефективності залежно від типу запиту.

Для операцій запису (API\_WRITE\_CHECKIN) спостерігається високий ефект:  $E_{\pi} = 0,94$  у станах із наявним з'єднанням і  $E_{\pi} = 1$  в офлайн-стані. Це зумовлено поєднанням адаптивної постановки операцій у чергу, керованих тайм-аутів і механізму обмеження марних спроб.

Для операцій читання (API\_READ\_EVENTS) ефект є помірним:  $E_{\pi} = 0,28 - 0,36$  за наявності з'єднання та зростає до  $E_{\pi} = 0,98$  в офлайн-умовах. Це відображає ефективність адаптивного кешування при доступному з'єднанні та визначальну роль кешу за його відсутності.

Для статичних ресурсів (STATIC) ефект є мінімальним у мережевих станах ( $E_{\pi} = 0,08 - 0,12$ ) і зростає до  $E_{\pi} = 0,96$  в офлайн-режимі. Це пояснюється тим, що більшість оптимізацій для таких ресурсів уже реалізується на рівні браузера, тоді як внесок політик стає суттєвим лише за відсутності

# МЕТОДИ ТА СИСТЕМИ ОПТИКО-ЕЛЕКТРОННОЇ І ЦИФРОВОЇ ОБРОБКИ ЗОБРАЖЕНЬ ТА СИГНАЛІВ

мережі.

Отримані результати підтверджують гіпотезу про підвищення завершеності операцій і демонструють, що ефективність адаптації залежить від типу операції та мережевого стану.

Таблиця 5 – Значення  $E_{\pi}$  для основних класів запитів

Профіль	Тип запиту	$E_{\pi}$	Профіль	Тип запиту	$E_{\pi}$	Профіль	Тип запиту	$E_{\pi}$
<i>broadband</i>	API_READ_EVENTS	0.34	4G	STATIC	0.12	2G	API_WRITE_CHECKIN	0.94
<i>broadband</i>	API_WRITE_CHECKIN	0.94	3G	API_READ_EVENTS	0.36	2G	STATIC	0.08
<i>broadband</i>	STATIC	0.084	3G	API_WRITE_CHECKIN	0.94	<i>offline</i>	API_READ_EVENTS	0.98
4G	API_READ_EVENTS	0.28	3G	STATIC	0.08	<i>offline</i>	API_WRITE_CHECKIN	1
4G	API_WRITE_CHECKIN	0.94	2G	API_READ_EVENTS	0.36	<i>offline</i>	STATIC	0.96

Таким чином, отримані результати показують, що в нестабільних мережевих умовах перевага запропонованого підходу на основі політик проявляється не лише у зниженні затримок, а й у керованому забезпеченні завершеності операцій. У варіанті V1 зафіксовано стале зменшення затримок, а в частині сценаріїв досягнуто зниження мережевої вартості. Для офлайн-сценаріїв найбільш важливим є те, що відкладене виконання і подальша синхронізація дають змогу забезпечити завершення операції після відновлення зв'язку. Отже, адаптація на основі політик є ефективною не лише як засіб покращення часових метрик взаємодії, а і як механізм підвищення надійності вебзастосунку.

## ВИСНОВКИ

У роботі запропоновано адаптивний підхід до функціонування PWA вебзастосунку в умовах обмежених і нестабільних мереж із акцентом на завершеність операцій, а не лише на швидкодію інтерфейсу. Проведене експериментальне дослідження показало, що використання адаптивних політик забезпечує систематичне зниження затримок порівняно з базовим варіантом. Найбільш виражений ефект спостерігається у сценаріях Sc1–Sc3, де затримки зменшуються в усіх станах мережі, причому найбільший вигрaш досягається в обмежених умовах.

Показано, що перевага адаптивного підходу проявляється не лише у часових характеристиках, а й у зменшенні мережевої вартості, особливо в сценаріях із нестабільною мережею під час сесії. При цьому встановлено, що ефект зниження вартості залежить від типу сценарію та мережевого профілю.

Ключовим результатом роботи є забезпечення завершеності операцій в офлайн-сценаріях. Для сценаріїв Sc4 і Sc6 показано, що в умовах відсутності мережі базовий підхід не забезпечує завершення операцій, тоді як використання адаптивних політик дозволяє гарантовано досягти їх завершення після відновлення з'єднання. Це означає перехід від незавершених до завершених операцій і забезпечення консистентного стану системи.

Отримані результати підтверджують, що ефект адаптації є багатовимірним і залежить від типу сценарію: у стабільних умовах основний вигрaш проявляється у зменшенні вартості, тоді як в обмежених та офлайн-умовах – у підвищенні завершеності операцій.

Таким чином, запропонований підхід забезпечує узгоджене покращення затримок, мережевої вартості та завершеності операцій і є доцільним для застосування в комп'ютерних системах із критичними операціями запису в умовах нестабільного з'єднання. Подальші дослідження доцільно спрямувати на узагальнення запропонованого підходу у вигляді фреймворку адаптивних політик із відокремленням конфігурації адаптації від прикладної логіки.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. World Wide Web Consortium. Service Workers: W3C Candidate Recommendation Draft. 2026. Available from: URL: <https://www.w3.org/standards/history/service-workers/>
2. Google Chrome Developers. Strategies for service worker caching. Workbox documentation. Available from: URL: <https://developer.chrome.com/docs/workbox/caching-strategies-overview>
3. Fielding R, Nottingham M, Reschke J. HTTP Caching. RFC 9111. RFC Editor; 2022. Available from: URL: <https://www.rfc-editor.org/rfc/rfc9111>
4. Malavolta I, Chinnappan K, Jasmontas L, Gupta S, Soltany KAK. Evaluating the impact of caching on the energy consumption and performance of progressive web apps. In: MOBILESoft '20: Proceedings

- of the IEEE/ACM 7th International Conference on Mobile Software Engineering and Systems; 2020. New York: ACM; 2020. p. 109-19. doi: 10.1145/3387905.3388593.
5. Malavolta I, Procaccianti G, Noorland P, Vukmirovic P. Assessing the impact of service workers on the energy efficiency of progressive web apps. In: 2017 IEEE/ACM 4th International Conference on Mobile Software Engineering and Systems (MOBILESoft); 2017. Piscataway (NJ): IEEE; 2017. p. 35-45. doi: 10.1109/MOBILOSoft.2017.7.
  6. Web Incubator Community Group. Web Background Synchronization: WICG Specification. 2021. Available from: URL: <https://wicg.github.io/background-sync/spec/>
  7. Web Incubator Community Group. Web Periodic Background Synchronization: WICG Specification. 2020 May 28. Available from: URL: <https://wicg.github.io/background-sync/spec/PeriodicBackgroundSync-index.html>
  8. de Munk O, Scoccia GL, Malavolta I. The state of the art in measurement-based experiments on the mobile web. *Inf Softw Technol.* 2022;149:106944. doi: 10.1016/j.infsof.2022.106944
  9. Приліпа А. О., Філатова А. Є. Клієнтський TinyML профайлер мережевих характеристик у веббраузері. Системи управління, навігації та зв'язку. 2025;4(82):114-20. doi: 10.26906/SUNZ.2025.4.114
  10. MDN Web Docs. Navigator.connection property. Mozilla; 2024. Available from: URL: <https://developer.mozilla.org/en-US/docs/Web/API/Navigator/connection>
  11. World Wide Web Consortium. Network Quality Estimation in Chrome. 2020. Available from: URL: <https://www.w3.org/2020/02/05-web-networks-Network-Quality-Estimation-in-Chrome.pdf>
  12. Jahromi HZ, Delaney DT, Hines A. Quantifying the influence of browser, OS and network delay on time instant metric measurements for a web mapping application. In: 2019 IEEE 19th International Conference on Communication Technology (ICCT); 2019. Piscataway (NJ): IEEE; 2019. p. 1580-4. doi: 10.1109/ICCT46805.2019.8947014
  13. Rajiullah M, Lutu A, Khatouni AS, Fida M, Mellia M, Brunström A, et al. Web experience in mobile networks: lessons from two million page visits. In: WWW '19: The World Wide Web Conference; 2019 May 13-17; San Francisco, CA, USA. New York: ACM; 2019. p. 1532-43. doi: 10.1145/3308558.3313606
  14. Varvello M, Blackburn J, Naylor D, Papagiannaki K. EYEORG: a platform for crowdsourcing web quality of experience measurements. 2019. <https://arxiv.org/abs/1902.02865>
  15. Bisikalo, O.; Kharchenko, V.; Kovtun, V.; Krak, I.; Pavlov, S. Parameterization of the Stochastic Model for Evaluating Variable Small Data in the Shannon Entropy Basis. *Entropy* 2023, 25, 184.
  16. Інтелектуальні технології в медичній діагностиці, лікуванні та реабілітації: монографія / [С.В. Павлов, О.Г. Аврунін, С.М. Злепко, Є.В. Бодяньський та ін.]; за редакцією С. Павлова, О. Авруніна. – Вінниця: ПП «ТД «Едельвейс і К», 2019. – 260 с.
  17. Hoffeld T, Wunderer S, Beyer A, Hall A, Schwind A, Gassner C, et al. White paper on crowdsourced network and QoE measurements: definitions, use cases and challenges. Würzburg: University of Würzburg; 2020. <https://doi.org/10.25972/OPUS-20232>.
  18. Romanyuk, O., Zavalniuk, Y., Pavlov, S., Chekhmestruk, R., etc. (2023). NEW SURFACE REFLECTANCE MODEL WITH THE COMBINATION OF TWO CUBIC FUNCTIONS USAGE. *Інформатика, Автоматика, Pomiry W Gospodarce I Ochronie Środowiska*, 13(3), 101–106.

#### REFERENCES

1. World Wide Web Consortium. Service Workers: W3C Candidate Recommendation Draft. 2026. Available from: URL: <https://www.w3.org/standards/history/service-workers/>
2. Google Chrome Developers. Strategies for service worker caching. Workbox documentation. Available from: URL: <https://developer.chrome.com/docs/workbox/caching-strategies-overview>
3. Fielding R, Nottingham M, Reschke J. HTTP Caching. RFC 9111. RFC Editor; 2022. Available from: URL: <https://www.rfc-editor.org/rfc/rfc9111>
4. Malavolta I, Chinnappan K, Jasmontas L, Gupta S, Soltany KAK. Evaluating the impact of caching on the energy consumption and performance of progressive web apps. In: MOBILESoft '20: Proceedings of the IEEE/ACM 7th International Conference on Mobile Software Engineering and Systems; 2020. New York: ACM; 2020. p. 109-19. doi: 10.1145/3387905.3388593.
5. Malavolta I, Procaccianti G, Noorland P, Vukmirovic P. Assessing the impact of service workers on the energy efficiency of progressive web apps. In: 2017 IEEE/ACM 4th International Conference on

---

---

## МЕТОДИ ТА СИСТЕМИ ОПТИКО-ЕЛЕКТРОННОЇ І ЦИФРОВОЇ ОБРОБКИ ЗОБРАЖЕНЬ ТА СИГНАЛІВ

---

---

- Mobile Software Engineering and Systems (MOBILESoft); 2017. Piscataway (NJ): IEEE; 2017. p. 35-45. doi: 10.1109/MOBILOSoft.2017.7.
6. Web Incubator Community Group. Web Background Synchronization: WICG Specification. 2021. Available from: URL: <https://wicg.github.io/background-sync/spec/>
  7. Web Incubator Community Group. Web Periodic Background Synchronization: WICG Specification. 2020 May 28. Available from: URL: <https://wicg.github.io/background-sync/spec/PeriodicBackgroundSync-index.html>
  8. de Munk O, Scoccia GL, Malavolta I. The state of the art in measurement-based experiments on the mobile web. *Inf Softw Technol.* 2022;149:106944. doi: 10.1016/j.infsof.2022.106944
  9. Prylipa A., Filatova A. TinyML network profiler in browser. *Systems of Control, Navigation and Communication.* 2025;4(82):114–120. doi: 10.26906/SUNZ.2025.4.114
  10. MDN Web Docs. Navigator.connection property. Mozilla; 2024. Available from: URL: <https://developer.mozilla.org/en-US/docs/Web/API/Navigator/connection>
  11. World Wide Web Consortium. Network Quality Estimation in Chrome. 2020. Available from: URL: <https://www.w3.org/2020/02/05-web-networks-Network-Quality-Estimation-in-Chrome.pdf>
  12. Jahromi HZ, Delaney DT, Hines A. Quantifying the influence of browser, OS and network delay on time instant metric measurements for a web mapping application. In: 2019 IEEE 19th International Conference on Communication Technology (ICCT); 2019. Piscataway (NJ): IEEE; 2019. p. 1580-4. doi: 10.1109/ICCT46805.2019.8947014
  13. Rajiullah M, Lutu A, Khatouni AS, Fida M, Mellia M, Brunström A, et al. Web experience in mobile networks: lessons from two million page visits. In: WWW '19: The World Wide Web Conference; 2019 May 13-17; San Francisco, CA, USA. New York: ACM; 2019. p. 1532-43. doi: 10.1145/3308558.3313606
  14. Varvello M, Blackburn J, Naylor D, Papagiannaki K. EYEORG: a platform for crowdsourcing web quality of experience measurements. 2019. <https://arxiv.org/abs/1902.02865>
  15. Bisikalo, O.; Kharchenko, V.; Kovtun, V.; Krak, I.; Pavlov, S. Parameterization of the Stochastic Model for Evaluating Variable Small Data in the Shannon Entropy Basis. *Entropy* 2023, 25, 184.
  16. Intellectual technologies in medical diagnostics, treatment and rehabilitation: monograph / [S.V. Pavlov, O.G. Avrunin, S.M. Zlepko, E.V. Bodianskyi and others]; edited by S. Pavlov, O. Avrunin. – Vinnytsia: PP “TD “Edelweiss and K”, 2019. – 260 p.
  17. Hoßfeld T, Wunderer S, Beyer A, Hall A, Schwind A, Gassner C, et al. White paper on crowdsourced network and QoE measurements: definitions, use cases and challenges. Würzburg: University of Würzburg; 2020. <https://doi.org/10.25972/OPUS-20232>.
  18. Romanyuk, O., Zavalniuk, Y., Pavlov, S., Chekhmestruk, R., etc. (2023). NEW SURFACE REFLECTANCE MODEL WITH THE COMBINATION OF TWO CUBIC FUNCTIONS USAGE. *Informatyka, Automatyka, Pomiarzy W Gospodarce I Ochronie Środowiska*, 13(3), 101–106.

Дата надходження: 20.03.2026

Дата прийняття до друку після рецензування: 21.04.2026

Дата публікації: 18.06.2026

Ця робота ліцензується відповідно до  
[Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/)

**ПРИЛІПА АРТЕМ ОЛЕГОВИЧ** – аспірант кафедри комп’ютерної інженерії та програмування, Національний технічний університет "Харківський політехнічний інститут", Харків, Україна;

**e-mail:** [Artem.Prylipa@cs.khpi.edu.ua](mailto:Artem.Prylipa@cs.khpi.edu.ua), <https://orcid.org/0009-0005-6633-8308>

**ГАННА ЄВГЕНІВНА ФІЛАТОВА** – доктор технічних наук, професор кафедри комп’ютерної інженерії та програмування, Національний технічний університет "Харківський політехнічний інститут", Харків, Україна; **e-mail:** [Hanna.Filatova@khpi.edu.ua](mailto:Hanna.Filatova@khpi.edu.ua), <https://orcid.org/0000-0003-1982-2322>

**Artem PRYLIPA, Hanna FILATOVA**  
**ADAPTIVE CONTENT OPTIMIZATION AND DEGRADATION POLICIES FOR WEB  
APPLICATIONS UNDER UNSTABLE NETWORK CONDITIONS**

National Technical University “Kharkiv Polytechnic Institute”