evaluating the effectiveness of the method are defined. The results indicate the feasibility of the agent-based approach for extending coverage and improving the resource efficiency of dynamic testing, and outline the direction of further experimental research.

Keywords: dynamic application security testing, large language models, agent-based systems, OpenAPI, BOLA, IDOR, test automation.

Надійшла до редакції (Received): 27.07.2026
Прийнята до друку (Accepted): 08.09.2026
Опубліковано онлайн (Available online): 15.09.2026

УДК 004.75:004.056 DOI: 10.31673/2409-7292.2026.033419

Прокопович-Ткаченко Дмитро Ігорович
кандидат технічних наук, доцент, завідувач кафедри кібербезпеки та інформаційних технологій
Університет митної справи та фінансів, Дніпро, Україна
ORCID: 0000-0002-6590-3898
E-mail: prokopovich-tkachenko@umsf.dp.ua

Пономарьов Ігор Володимирович
кандидат технічних наук, доцент кафедри електронних обчислювальних машин
Дніпровський національний університет імені Олеся Гончара, Дніпро, Україна
ORCID: 0009-0009-7139-2885
E-mail: ponomarov_i@365.dnu.edu.ua

Бабенко Костянтин Євгенович
аспірант кафедри комп'ютерних наук та інформаційних технологій, факультету фізики, електроніки та комп'ютерних систем
Дніпровський національний університет імені Олеся Гончара, Дніпро, Україна
ORCID: 0009-0009-0945-7078
E-mail: babenko-k@365.dnu.edu.ua

ОПЕРАЦІЙНЕ ВИЗНАЧЕННЯ ТА МЕТРИКИ НАСКРІЗНОЇ ЗАТРИМКИ БЕЗПЕЧНОГО ДОСТУПУ ДО СТАНУ БЛОКЧЕЙН-СИСТЕМ ПЕРШОГО РІВНЯ

У статті досліджено проблему операційного визначення затримки, яку спостерігає кінцевий користувач програмованої блокчейн-системи першого рівня між поданням підписаної транзакції та першим підтвердженим читанням оновленого стану через RPC-інтерфейс. Актуальність зумовлена тим, що метрики консенсусної фінальності не охоплюють індексацію, публікацію стану та клієнтське виявлення, а тому можуть занижувати оцінку доступності сервісу і тривалості використання застарілого стану. Мета роботи – формалізувати користувацько-видиму затримку T_visible як окремий об'єкт вимірювання та запропонувати відтворювану систему супровідних показників. Методика охоплює причинну декомпозицію конвеєра на етапи приймання, пакетування, консенсусу, виконання, фіксації, індексації, публікації та виявлення; формальний предикат видимості сертифікованого стану; розмежування правосторонньо цензурованих спостережень і конкуруючих кінцевих подій; оцінювання квантилів і відокремлення дискретизаційної складової опитування. У чисельному експерименті, параметри якого відкалібровано за опублікованими вимірюваннями конвеєрних блокчейнів, медіанний спостережуваний розрив між фінальністю та першим читанням становив 168–241 мс, або до 27 % повної затримки, а консервативний індикатор ймовірності застарілої відповіді для затримки 200 мс досягав 0,69. Паралельний конвеєр з раннім оприлюдненням стану скоротив медіану T_visible з 889 до 686 мс і частку порушень порогу 1 с – з 17,7 % до 0,2 %. Наукова новизна полягає у поєднанні предиката видимості, метрик розриву видимості, застарілої відповіді та порушення порогу в єдиному протоколі вимірювання. Практичне значення – можливість коректно порівнювати сценарії, обґрунтовувати SLO та оцінювати доступність і цілісність даних, що надаються користувачеві.

Ключові слова: наскрізна затримка, видимість стану, блокчейн-система, RPC-інтерфейс, кіберстійкість, застаріла відповідь, якість обслуговування.

Вступ і постановка проблеми

Сучасні блокчейн-системи першого рівня конкурують за пропускною здатністю та затримкою, що визначають придатність платформ для платіжних, інфраструктурних і кіберфізичних застосувань. У науковій літературі та індустріальних звітах переважають метрики окремих компонентів конвеєра, насамперед візантійсько-стійкого консенсусу. Архітектура Zaptos показує, що консенсус не є єдиним джерелом затримки, і вимірює шлях від подання транзакції до підтвердження її фіксації [1]. Однак кінцевий користувач взаємодіє через RPC-інтерфейс і бачить результат лише після індексації стану, його реплікації до обслуговувальних вузлів, проходження черги запитів і клієнтського циклу виявлення. Для захисту інформації цей часовий розрив важливий як вікно отримання застарілого стану: він впливає на доступність сервісу, цілісність даних, що використовуються для прийняття рішень, і коректність SLO. Відсутність узгодженого операційного визначення користувацько-видимої затримки унеможливлює коректне порівняння сценаріїв та призводить до систематично оптимістичних оцінок.

Аналіз останніх досліджень і публікацій

Дослідження зменшення затримки зосереджені переважно на консенсусі: PBFT [2], HotStuff [3] та Jolteon/Ditto [4] оптимізують кількість раундів, складність повідомлень і адаптацію до мережевих умов. DAG-орієнтовані Bullshark [5], Narwhal/Tusk [6], Mysticeti [7], Shoal++ [8] і Shoal [9], а також ланцюговий Moonshot [10] скорочують шлях упорядкування за високої пропускної здатності. Теоретичні та прикладні засади реплікованого стану представлено у Streamlet [11], SBFT [12], моделі часткової синхронності [13] і моделі візантійських відмов [14]. На рівні цілісних систем продуктивність та фінальність досліджено для Algorand [15], Hyperledger Fabric [16], Red Belly [17], Sui Lutris [18] і Porygon [19], тоді як Block-STM [20] прискорює паралельне виконання. Захист сертифікованого стану спирається на BLS-підписи [21] і дерева Меркла [22]. У цих роботах кінцевою подією зазвичай є впорядкування або фіксація на вузлі; готовність RPC-стану та клієнтське виявлення не утворюють окремого об’єкта вимірювання.

Невирішені частини проблеми

У проаналізованих працях не виявлено узгодженого визначення першого перевіреного RPC-читання конкретної версії стану. Кінцева подія змінюється від упорядкування блоку до фіксації на вузлі і зазвичай не охоплює готовність RPC-сервісу та клієнтське виявлення. Не розрізняються праве цензурування наприкінці вікна спостереження і конкуруючі кінцеві події – відхилення чи закінчення строку дії транзакції. Крім того, періодичне опитування вносить дискретизаційну складову, а криптографічно коректний доказ старого стану не гарантує його актуальності. Тому потрібна система, що розділяє блокчейн-фінальність, готовність серверного стану, клієнтське виявлення та політику перевірки свіжості.

Мета та завдання дослідження

Мета статті – розробити операційне визначення та систему метрик наскрізної затримки безпечного доступу користувача до оновленого стану програмованої блокчейн-системи першого рівня. Об’єкт дослідження – процеси оброблення транзакцій і надання сертифікованого стану через RPC. Предмет дослідження – операційні визначення, статистичні оцінки та правила вимірювання користувацько-видимої затримки. Завдання:

1) визначити початкову й кінцеву події, предикат видимості та політику перевірки свіжості;

2) побудувати декомпозицію критичного шляху для послідовної й паралельної організації;

3) розмежувати праве цензурування та конкуруючі результати і сформувати супровідні показники;

4) проілюструвати методику чисельним експериментом. Науковий внесок полягає у формалізації першого перевіреного RPC-читання конкретної версії стану та розділенні готовності серверного стану і клієнтського виявлення.

© Прокопович-Ткаченко Д.І., Пономарьов І.В., Бабенко К.Є. Операційне визначення та метрики наскрізної затримки безпечного доступу до стану блокчейн-систем першого рівня. Сучасний захист інформації, 3(67), 91–103. https://doi.org/10.31673/2409-7292.2026.033419

Матеріали та методи дослідження

Модель системи

Розглядається блокчейн-система з клієнтами, повними вузлами та валідаторами, географічно розподіленими у мережі. Прийнято модель n = 3f + 1 валідаторів, серед яких не більше f є візантійськими; мережа частково синхронна [13, 14]. Консенсус формує доказ фінальності відповідно до політики конкретної платформи. Де це передбачено протоколом, доказ може використовувати агреговані BLS-підписи [21], а автентифікація стану – дерева Меркла [22]. Виконання транзакцій є детермінованим і може бути паралельним [20]. Модель охоплює конвеєрні архітектури класу Aptos [1, 23] та, після адаптації політики фінальності й RPC-методу, платформи з прямішою взаємодією клієнтів із валідаторами [24].

Модель загроз і властивості безпеки

Зловмисник може контролювати частину валідаторів або недовірчий RPC-вузол, затримувати публікацію, повторно надавати старий, але криптографічно коректний стан, чи збільшувати чергу обслуговування. Тому доказ Меркла сам по собі не гарантує свіжості: корінь стану має бути пов’язаний з актуальним checkpoint, незалежно отриманим легким клієнтом або від кворуму джерел. Політика прийняття також фіксує статус транзакції, висоту чи версію стану та допустимий результат виконання. Вимірювальна транзакція повинна змінювати унікальний ключ або перевірятися за квитанцією, щоб уникнути тривіальної істинності предиката для порожньої чи перезаписаної зміни.

Події життєвого циклу транзакції

Основну метрику вимірюють монотонним годинником клієнта: t_sub – момент передавання підписаної транзакції; t_read – момент першого успішного перевіреного читання; t_fin^(c) – момент отримання і перевірки клієнтом доказу фінальності. Події t_ord і внутрішня t_fin з журналів вузлів є діагностичними: їх дозволено зіставляти з клієнтською шкалою лише після оцінювання зсуву годинників і звітування похибки синхронізації. t_ready позначає найраніший момент, коли коректний RPC-запит міг би повернути прийнятий стан. Перелік подій наведено в табл. 1.

Таблиця 1

Події вимірювальної моделі та їхня спостережність

Подія	Зміст	Точка спостереження	Доступність для клієнта
t_{sub}	Подання підписаної транзакції	Клієнтський застосунок	пряма
t_{acc}	Приймання до мемпула	Відповідь RPC-вузла	пряма
t_{ord}	Упорядкування блоку консенсусом	Журнали валідаторів	опосередкована
t_fin^(c)	Отримання і перевірка доказу фінальності	Клієнт / легкий клієнт	пряма
t_ready	Готовність стану до коректної RPC-відповіді	RPC-вузол / трасування	опосередкована
t_{read}	Перше підтверджене читання стану	Клієнтський застосунок	пряма

Операційне визначення

Наскрізною користувацько-видимою затримкою транзакції x називається величина:

$$T_{\text{visible}}(x) = t_{\text{read}}(x) - t_{\text{sub}}(x)\ . \tag{1}$$

Відповідно до (1), момент першого перевіреного читання визначається за (2) через предикат видимості V(x, t) і множину моментів опитування T_poll:

$$t_{\text{read}}(x) = \min\{t \in \mathcal{T}_{\text{poll}} : V(x,t) = 1\}\ , \tag{2}$$

$$V(x,t) = 1 \Leftrightarrow F(B) = 1 \wedge R_x(B) \in \mathcal{A} \wedge h_{\text{rpc}}(t) \geq h(B) \wedge \Pi_{\text{rpc}}(t,B) = 1\ . \tag{3}$$

© Прокопович-Ткаченко Д.І., Пономарьов І.В., Бабенко К.Є. Операційне визначення та метрики наскрізної затримки безпечного доступу до стану блокчейн-систем першого рівня. Сучасний захист інформації, 3(67), 91–103. https://doi.org/10.31673/2409-7292.2026.033419

Безпечна політика видимості

У предикаті (3) F(B) означає перевірку політики фінальності блоку B; R_x(B) – квитанцію транзакції, статус якої належить наперед визначеній множині A; h_rpc(t) – висоту або версію стану, доступного через RPC; П_rpc(t, B) – перевірку доказу читання відносно актуального checkpoint. Для транзакції з відкотом допустимий статус фіксується окремо; для вимірювання зміни стану застосовується унікальний ключ. Така політика не ототожнює наявність старого коректного доказу з актуальністю стану.

Цензурування та конкуруючі події

Спостереження поділяються на успішні, правосторонньо цензуровані наприкінці вікна W та завершені конкуруючою подією – остаточним відхиленням або закінченням строку дії транзакції. Для неінформативного правого цензурування застосовується добутково-граничний оцінювач Каплана-Маєра [25], тоді як відхилення не прирівнюється до цензурування і його частота оцінюється окремо; за потреби спільного аналізу використовується функція кумулятивної інцидентності [26]:

$$\hat{F}(t) = 1 - \prod_{t_{(j)} \le t} \left(1 - \frac{d_j}{n_j}\right). \quad (4)$$

У формулі (4) t_(j) – упорядковані моменти виявлення, d_j – кількість виявлень у момент t_(j), n_j – кількість транзакцій під ризиком. Навіть мала частка цензурування може впливати на p99, тому один і той самий оцінювач слід застосовувати послідовно, а кількість успішних, цензурованих і відхилених операцій – звітувати окремо.

Причинна декомпозиція конвеєра

Наскрізний шлях розкладається на етапи приймання, пакетування, консенсусу, виконання, фіксації, індексації, публікації та виявлення (табл. 2). Для уникнення подвійного обліку загальна метрика поділяється на блокчейн-складову T_blockchain, залишковий серверний шлях D_srv і клієнтське виявлення D_disc:

$$T_{\text{visible}} = T_{\text{blockchain}} + D_{\text{srv}} + D_{\text{disc}}, \quad D_{\text{srv}} = D_{\text{idx}} + D_{\text{pub}}. \quad (5)$$

Таблиця 2

Декомпозиція наскрізного конвеєра доставлення стану

Етап	Початкова подія	Кінцева подія	Домінантний ресурс	Символ
Приймання	подання транзакції	підтвердження мемпула	мережа	D_{acc}
Черга пакетування	приймання	включення до пакета/блоку	–	D_{que}
Консенсус	пропозиція блоку	доказ упорядкування	мережа	D_{con}
Виконання	упорядкування	новий стан обчислено	процесор	D_{exe}
Фіксація	сертифікація стану	стан збережено	введення-виведення	D_{cmt}
Індексація	фіксація	індекси RPC оновлено	введення-виведення	D_{idx}
Публікація RPC	індексація	стан доступний для читання	мережа	D_{pub}
Виявлення	доступність стану	перше успішне читання	–	D_{disc}

Послідовна організація

Коли кожний етап починається після завершення попереднього, критичний шлях є сумою тривалостей, включно з чергою пакетування:

$$T_{\text{seq}} = D_{\text{acc}} + D_{\text{que}} + D_{\text{con}} + D_{\text{exe}} + D_{\text{cmt}} + D_{\text{idx}} + D_{\text{pub}} + D_{\text{disc}}. \quad (6)$$

Паралельна організація

Коли виконання, фіксація та індексація перекриваються у часі з консенсусом, використовується залишок критичного шляху, а не проста сума всіх етапів:

© Прокопович-Ткаченко Д.І., Пономарьов І.В., Бабенко К.Є. Операційне визначення та метрики наскрізної затримки безпечного доступу до стану блокчейн-систем першого рівня. Сучасний захист інформації, 3(67), 91–103. https://doi.org/10.31673/2409-7292.2026.033419

$$T_{\text{par}} = D_{\text{acc}} + D_{\text{que}} + D_{\text{con}} + \max\left(D_{\text{exe}} + D_{\text{cmt}} + D_{\text{idx}} - \Omega, 0\right) + D'_{\text{pub}} + D_{\text{disc}} . \quad (7)$$

У формулі (7) Ω – сумарне перекриття виконання, фіксації та індексації з вікном консенсусу; D'_pub – залишкова публікація після підтвердження фінальності. Значення Ω залежить від реалізації та має оцінюватися за трасуванням. Формули (6) і (7) є сценарними моделями критичного шляху, а не універсальними тотожностями для всіх L1-платформ.

Відокремлення складової виявлення

За незалежної рівномірної фази опитування з інтервалом τ додаткова затримка виявлення розподілена рівномірно на [0, τ], звідки отримуємо (8):

$$\mathrm{E}[D_{\text{disc}}] = \frac{\tau}{2}, \quad \mathrm{Var}[D_{\text{disc}}] = \frac{\tau^2}{12} . \quad (8)$$

Оскільки спостережувана величина є сумою системної складової T_sys та D_disc, віднімання τ/2 у (9) є лише поправкою першого порядку, а не повною деконволюцією. Її похибка може досягати τ/2; припущення про рівномірну фазу потрібно забезпечувати рандомізацією початку опитування або перевіряти емпірично:

$$\hat{q}_{\text{sys}}(p) \approx q_{\text{obs}}(p) - \frac{\tau}{2} . \quad (9)$$

Оцінювання черги RPC

Закон Літтла пов'язує середню кількість запитів N, інтенсивність λ та повний час перебування W. Для відокремлення очікування в черзі W_q потрібен також середній час обслуговування W_s:

$$N = \lambda W, \quad W_q = W - W_s . \quad (10)$$

Отже, вираз (10) використовується як діагностична перевірка балансу потоку, але сам по собі не визначає внесок черги в D_pub без вимірювання часу обслуговування.

Супровідні показники

Система розрізняє серверну готовність, клієнтське виявлення та їхню спостережувану суму, як визначено в (11):

$$G_{\text{ready}} = t_{\text{ready}} - t_{\text{fin}}^{(c)}, \quad D_{\text{disc}} = t_{\text{read}} - t_{\text{ready}}, \quad G_{\text{obs}} = G_{\text{ready}} + D_{\text{disc}} . \quad (11)$$

Імовірність застарілої відповіді для запиту, надісланого через фіксовану затримку Δ після клієнтсько-перевіреної фінальності, визначається за (12):

$$P_{\text{stale}}(\Delta) = P\{G_{\text{ready}} > \Delta\} . \quad (12)$$

Для запитів, рівномірно розміщених у [0, τ], потрібне усереднення P{G_ready > u} за u. Оскільки в наявній імітаційній вибірці t_ready не журналювався окремо, у результатах наведено консервативний проксі-індикатор P{G_obs > Δ}. Частота порушень порогу обслуговування L визначається за (13):

$$\varepsilon(L) = P\{T_{\text{visible}} > L\} . \quad (13)$$

Вплив конкуруючих відхилень ураховується окремою ймовірністю P_rej та очікуваним штрафом повторного подання:

$$\mathrm{E}[\Delta T] = P_{\text{rej}}\left(T_{\text{retry}} + \mathrm{E}[T'_{\text{visible}}]\right) . \quad (14)$$

У виразі (14) T_retry – інтервал до повторного подання, T'_visible – затримка повторної спроби. Квантилі рівня p оцінюються порядковими статистиками:

$$\hat{q}(p) = T_{(\lceil pn \rceil)}, \quad p \in \{0.50, 0.95, 0.99\} . \quad (15)$$

У чисельній ілюстрації нижче подано точкові оцінки (15). Для платформного benchmark необхідні незалежні повторення, чисельники подій і довірчі інтервали. Порівняння меж поширених метрик наведено в табл. 3.

© Прокопович-Ткаченко Д.І., Пономарьов І.В., Бабенко К.Є. Операційне визначення та метрики наскрізної затримки безпечного доступу до стану блокчейн-систем першого рівня. Сучасний захист інформації, 3(67), 91–103. https://doi.org/10.31673/2409-7292.2026.033419

Таблиця 3

Порівняння метрик затримки блокчейн-систем

Метрика	Початкова подія	Кінцева подія	Непокриті етапи	Джерела
Затримка впорядкування консенсусу	пропозиція блоку	доказ упорядкування	усі, крім консенсусу	[2–4]
Затримка фіксації валідатора	подання до валідатора	фіксація стану	індексація, публікація, виявлення	[3, 5, 9]
Наскрізна затримка повного вузла	приймання повним вузлом	фіксація на повному вузлі	індексація, публікація, виявлення	[1, 18]
Час до фінальності (TTF)	подання транзакції	фінальність блоку	індексація, публікація, виявлення	[15, 27]
$T_{visible}$ (пропонована)	подання транзакції	перше підтверджене читання	—	ця стаття

Часову структуру обох сценаріїв ілюструє рис. 1: у послідовній організації етапи утворюють неперервний критичний шлях, тоді як у паралельній виконання, фіксація та індексація частково перекриваються у часі з консенсусом; після підтвердження фінальності залишаються лише неперекриті серверні етапи та клієнтське виявлення.

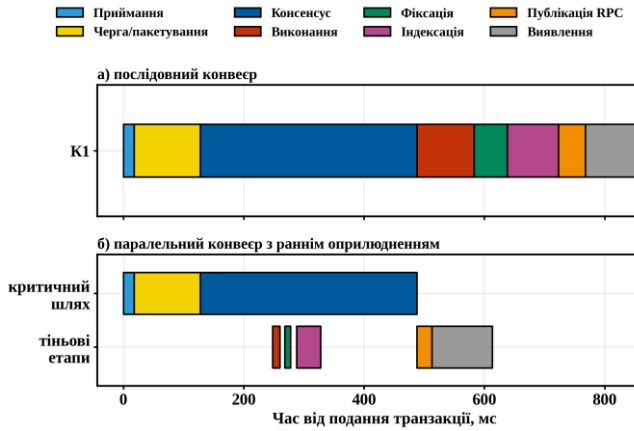

Рис. 1. Часова діаграма етапів наскрізного конвеєра: а) послідовна організація; б) паралельна організація з раннім оприлюдненням стану

Умови чисельної апробації

Ілюстративні дані згенеровано стохастичною моделлю (Python 3.11, NumPy 1.26, зерно 42) з логнормальними тривалостями етапів і рівномірними чергами. Медіанні параметри відкалібровано за опублікованими вимірюваннями 100-валідаторної конфігурації [1]; обсяг вибірки – 4000 транзакцій на сценарій, τ = 200 мс, W = 10 с. Числові результати ілюструють застосування протоколу, але не є незалежною валідацією чи benchmark конкретної платформи. Для повного відтворення пакет супровідних матеріалів має містити параметри масштабу і кореляцій розподілів, код, згенеровану вибірку та кількість успішних, цензурованих і відхилених операцій.

Результати імітаційного експерименту

Сценарії імітаційної моделі

Порівнювалися три сценарії: K1 – послідовний конвеєр за (6); K2 – гіпотетичний конвеєр з випереджальним виконанням і скороченою фіксацією; K3 – паралельний конвеєр за (7) з

частковим суміщенням індексації та фіксації. Параметри чисельного експерименту наведено в табл. 4.

Таблиця 4

Параметри імітаційної моделі

Параметр	Значення	Коментар
Кількість валідаторів / повних вузлів, од.	100 / 30	калібрування за [1]
Медіанна затримка впорядкування, мс	360	3 раунди консенсусу [4, 10]
Інтервал пакетування, мс	200	черга (0…220 мс)
Медіана виконання блоку (К1/К2/К3), мс	95 / 30 / 12	паралельне виконання [20]
Медіана фіксації (К1/К2/К3), мс	55 / 35 / 10	оптимістична фіксація [1]
Медіана індексації (К1/К2/К3), мс	85 / 85 / 40	суміщення з фіксацією у К3
Інтервал опитування τ, мс	200	формули (8), (9)
Обсяг вибірки на сценарій, тр.	4000	вікно W = 10 с

Розподіли модельної наскрізної затримки

Емпіричні функції розподілу T_visible для трьох сценаріїв наведено на рис. 2. Медіана становить 889 мс для К1, 799 мс для К2 та 686 мс для К3; p95 – відповідно 1090, 991 і 872 мс; p99 – 1174, 1067 і 948 мс (табл. 5). У межах моделі паралелізація скорочує медіану на 23 %, а p99 – на 19 % щодо К1. Ці відносні значення є результатами одного чисельного сценарію і не переносяться безпосередньо на публічні мережі.

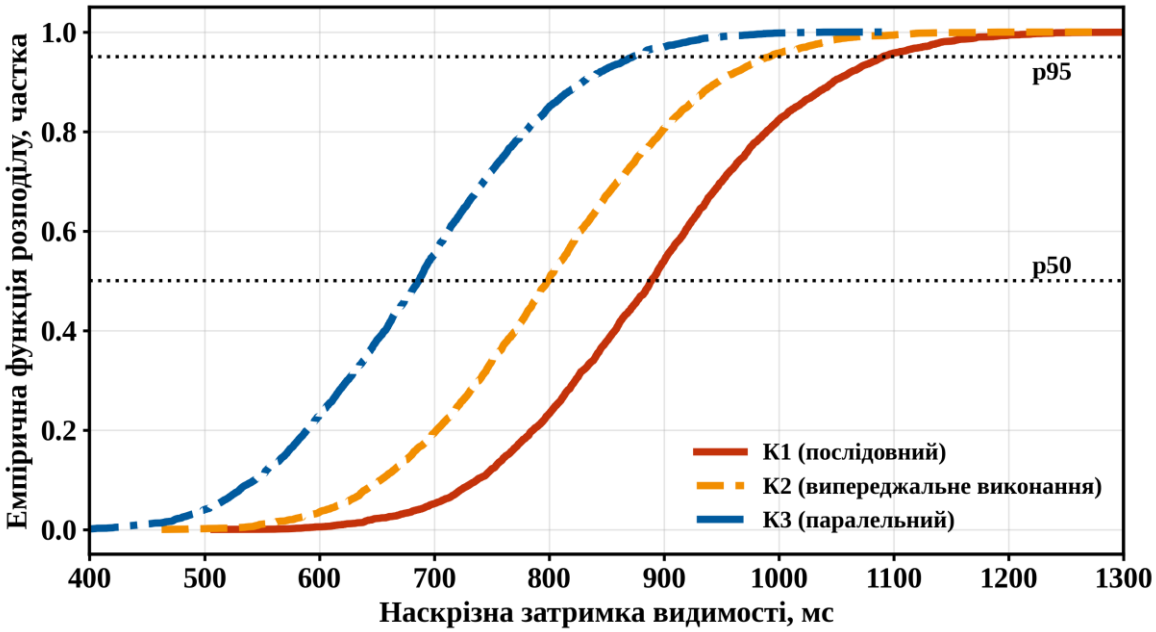

Рис. 2. Емпіричні функції розподілу модельної користувацько-видимої затримки для сценаріїв К1-К3

Таблиця 5

Квантилі модельної наскрізної затримки T_visible та її системної складової, мс

Сценарій	p50	p95	p99	Середнє	p50 системне, за (9)
К1 (послідовна)	889	1090	1174	890	789
К2 (випереджальне виконання)	799	991	1067	801	699
К3 (паралельна)	686	872	948	686	586

Декомпозиція затримки

На рис. 3 наведено суму медіан окремих модельних компонентів для К1. Консенсус має найбільшу медіану (358 мс), а сума медіан позаконсенсусних етапів і черги демонструє вагомий внесок серверного та клієнтського шляху. Сума медіан компонентів загалом не дорівнює медіані їхньої суми, тому рисунок використовується лише як описова декомпозиція, а квантилі T_visible беруться безпосередньо з емпіричного розподілу в табл. 5.

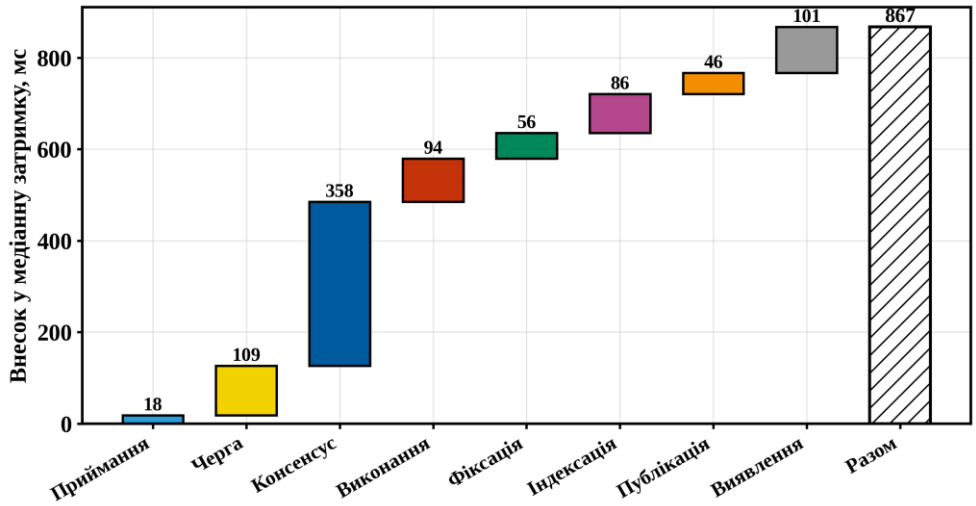

Рис. 3. Сума медіан модельних компонентів затримки для сценарію К1

Порівняння етапів між сценаріями за діапазоном p50–p99 наведено на рис. 4. У межах заданої імітаційної моделі для К3 діапазони виконання, фіксації та індексації є вужчими, тоді як розподіли консенсусу, черги та виявлення близькі. Це узгоджується із закладеним сценарієм оптимізації, але не є самостійним доказом коректності декомпозиції (5).

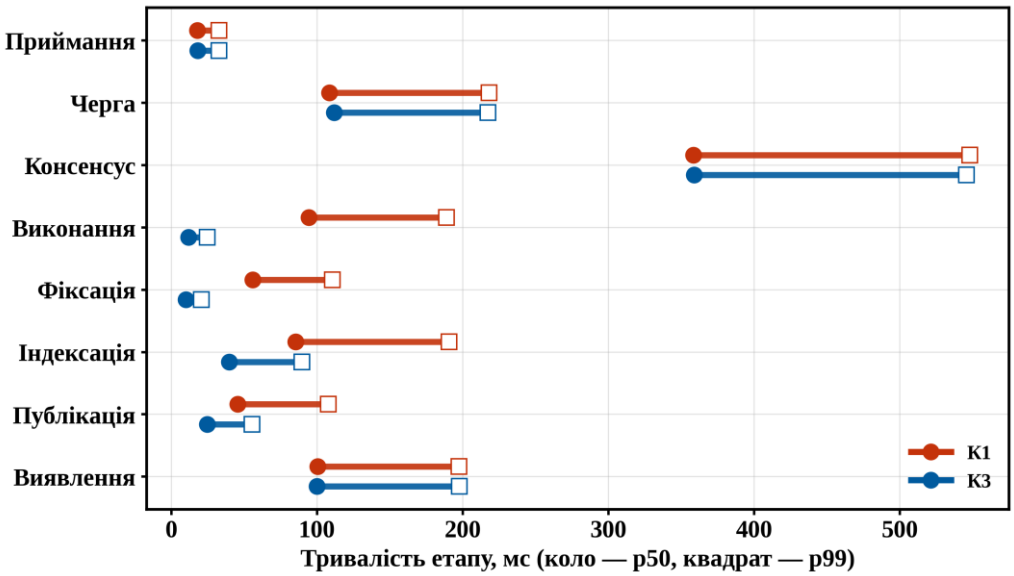

Рис. 4. Діапазони p50–p99 модельних тривалостей етапів для сценаріїв К1 і К3

Поведінка під навантаженням

На рис. 5 показано ілюстративне адитивне подання медіан тривалостей компонентів К3 за навантаження 1–20 тис. транзакцій за секунду. Найбільше зростають пакетувальна черга і впорядкування. Для етапів, що перекриваються у часі, висота стеку не трактується як фактичний критичний шлях; для цього застосовується сценарна модель (7).

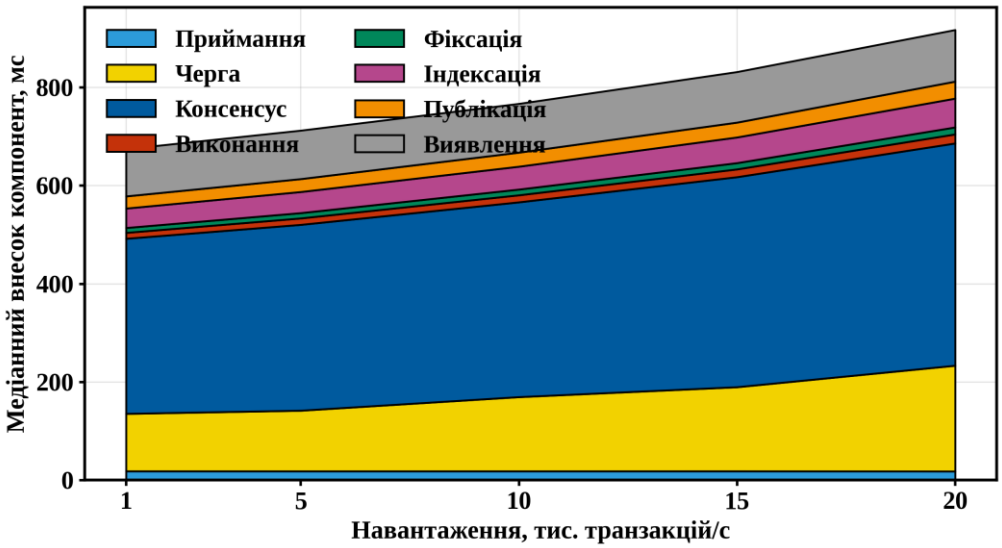

Рис. 5. Ілюстративне адитивне подання медіан компонентів К3 залежно від навантаження

Спостережуваний розрив і застаріла відповідь

У табл. 6 наведено G_obs, що включає серверну готовність і очікування наступного опитування. Його медіана становить 241 мс у К1 та 168 мс у К3. Оскільки t_ready не збережено окремо, показник P{G_obs > 200 мс} подано як консервативний проксі ймовірності застарілої відповіді; він становить 0,69 для К1 та 0,34 для К3. Частота порушення порогу T_visible > 1 с дорівнює 17,7 %, 4,3 % та 0,2 % для К1–К3 відповідно.

Таблиця 6

Спостережувані супровідні показники доступності стану

Сценарій	G_obs p50, мс	G_obs p95, мс	Проксі P_stale(Δ=200 мс)	ε(L=1 с), %
К1 (послідовна)	241	352	0,69	17,7
К2 (випереджальне виконання)	239	347	0,69	4,3
К3 (паралельна)	168	262	0,34	0,2

Узагальнене зіставлення квантилів сценаріїв К1 і К3 подано на рис. 6. У цій імітаційній вибірці відносне скорочення становить 23 % за медіаною, 20 % за p95 та 19 % за p99. Стійкість такого співвідношення у хвості потребує незалежних повторень, довірчих інтервалів і аналізу чутливості до параметрів розподілів.

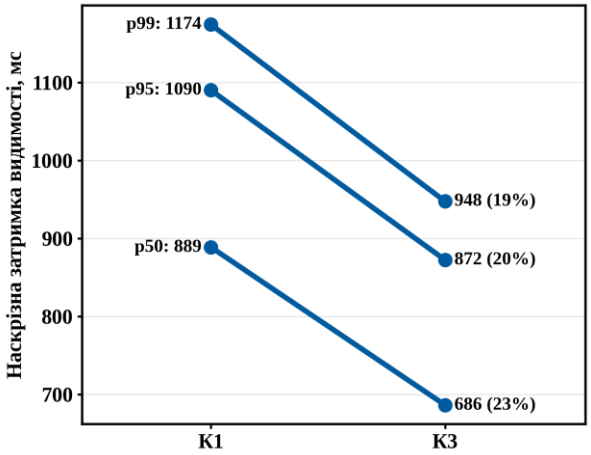

Рис. 6. Зіставлення квантилів модельної наскрізної затримки сценаріїв К1 і К3

Хвостову поведінку модельних розподілів у логарифмічному масштабі показано на рис. 7. Для К1 оцінка ймовірності перевищення порогу 1 с становить 1,8·10−1, для К2 – 4,3·10−2, для К3 – 2·10−3. У межах цієї вибірки показник ε(L) чіткіше за медіану розрізняє сценарії, однак для угод про рівень обслуговування слід також наводити кількість подій і довірчі інтервали.

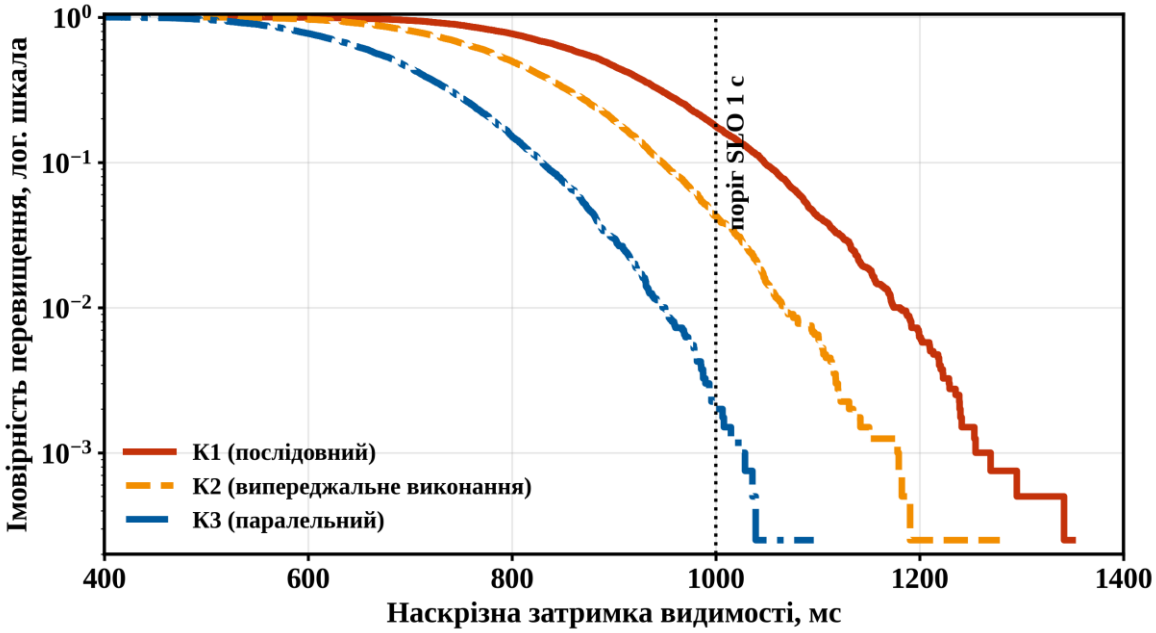

Рис. 7. Імовірність перевищення заданого рівня модельної затримки для сценаріїв К1-К3

Обговорення результатів

Чисельний експеримент ілюструє головну тезу: метрика, що завершується фіксацією на вузлі, не охоплює весь шлях до першого перевіреного RPC-читання. Це узгоджується зі спостереженням Zaptos про вагомість позаконсенсусних етапів [1], але запропонована схема додатково розділяє серверну готовність, клієнтське виявлення та перевірку свіжості. Для Porygon [19] механізм сертифікації впливає на момент, після якого RPC-вузол має право публікувати стан; для Sui Lutris [18] та інших платформ політика прийняття адаптується до конкретного швидкого шляху. Отримані числа слід трактувати як модельну ілюстрацію, а не як доказ переваги певної реалізації.

Розбіжності з наведеними в літературі значеннями пояснюються різними кінцевими подіями вимірювання. Вузлова метрика залишається корисною для локалізації затримок, однак міжплатформне порівняння коректне лише за однакових початкової події, політики фінальності, RPC-методу, способу перевірки доказу, географії клієнта й інтервалу опитування. Практично доцільно звітувати T_visible разом із G_ready, D_disc, часткою конкуруючих результатів і похибкою синхронізації.

Обмеження дослідження

Результати отримано на синтетичній вибірці, відкаліброваній переважно за [1]; повного коду, параметрів хвостів, незалежних повторень і довірчих інтервалів у вихідному наборі матеріалів немає. Отже, точні відсотки не є платформним benchmark. Перенесення на публічні мережі потребує врахування кешів, балансувальників, географії RPC-провайдерів і різних політик фінальності; D_idx може бути нульовим для RPC-методів без окремого індексатора. Зловмисний RPC може надати правильний доказ старого кореня, тому свіжість вимагає незалежного актуального checkpoint. Для натурного вимірювання G_ready потрібне інструментування серверної готовності або клієнтсько-спостережуваний доказ, а часові мітки різних вузлів мають бути приведені до однієї шкали з оцінкою похибки.

© Прокопович-Ткаченко Д.І., Пономарьов І.В., Бабенко К.Є. Операційне визначення та метрики наскрізної затримки безпечного доступу до стану блокчейн-систем першого рівня. Сучасний захист інформації, 3(67), 91–103. https://doi.org/10.31673/2409-7292.2026.033419

Висновки та практичні рекомендації

1. Запропоновано операційне визначення T_visible від подання транзакції до першого перевіреного RPC-читання конкретної версії стану. Політика видимості поєднує фінальність, квитанцію, версію та доказ, пов'язаний з актуальним checkpoint.

2. Уточнено декомпозицію критичного шляху: послідовний сценарій містить пакетувальну чергу, а для паралельного сценарію перекриті етапи враховуються через залишок критичного шляху. Окремо визначено серверну готовність і клієнтське виявлення.

3. Розмежовано правостороннє цензурування і конкуруючі результати; для SLO рекомендовано окремо звітувати успішні, цензуровані та відхилені операції, p50/p95/p99, G_ready, D_disc, P_stale(Δ) та ε(L).

4. Чисельна апробація ілюструє потенційний масштаб розриву між фінальністю і першим читанням, але через синтетичні дані не є незалежною валідацією. Перед платформним порівнянням необхідні відкритий код, повні параметри розподілів, кілька зерен, довірчі інтервали та натурне трасування.

У протоколі вимірювання слід фіксувати версію платформи й RPC-метод, політику фінальності, джерело checkpoint, інтервал і фазу опитування, географію клієнта, похибку синхронізації та статус кожної транзакції. Для недовірчих RPC-вузлів необхідно перевіряти не лише доказ включення, а й свіжість кореня стану. Подальші дослідження мають охопити натурну перевірку на EVM- і Move-платформах, WebSocket-підписки та сценарії навмисної затримки або повторного надання старого стану.

Перелік посилань

1. Xiang, Z., Li, Z., Arun, B., Zhang, T., & Spiegelman, A. (2025). Zaptos: Towards optimal blockchain latency. arXiv. https://doi.org/10.48550/arXiv.2501.10612.

2. Castro, M., & Liskov, B. (2002). Practical Byzantine fault tolerance and proactive recovery. ACM Transactions on Computer Systems, 20(4), 398–461. https://doi.org/10.1145/571637.571640.

3. Yin, M., Malkhi, D., Reiter, M. K., Gueta, G. G., & Abraham, I. (2019). HotStuff: BFT consensus with linearity and responsiveness. In Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing (pp. 347–356). ACM. https://doi.org/10.1145/3293611.3331591.

4. Gelashvili, R., Kokoris-Kogias, L., Sonnino, A., Spiegelman, A., & Xiang, Z. (2022). Jolteon and Ditto: Network-adaptive efficient consensus with asynchronous fallback. In Financial Cryptography and Data Security, FC 2022 (LNCS, Vol. 13411, pp. 296–315). Springer. https://doi.org/10.1007/978-3-031-18283-9_14.

5. Spiegelman, A., Giridharan, N., Sonnino, A., & Kokoris-Kogias, L. (2022). Bullshark: DAG BFT protocols made practical. In Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security (pp. 2705–2718). ACM. https://doi.org/10.1145/3548606.3559361.

6. Danezis, G., Kokoris-Kogias, L., Sonnino, A., & Spiegelman, A. (2022). Narwhal and Tusk: A DAG-based mempool and efficient BFT consensus. In Proceedings of the Seventeenth European Conference on Computer Systems (pp. 34–50). ACM. https://doi.org/10.1145/3492321.3519594.

7. Babel, K., Chursin, A., Danezis, G., Kokoris-Kogias, L., & Sonnino, A. (2023). Mysticeti: Low-latency DAG consensus with fast commit path. arXiv. https://doi.org/10.48550/arXiv.2310.14821.

8. Arun, B., Li, Z., Suri-Payer, F., Das, S., & Spiegelman, A. (2024). Shoal++: High throughput DAG BFT can be fast! arXiv. https://doi.org/10.48550/arXiv.2405.20488.

9. Spiegelman, A., Arun, B., Gelashvili, R., & Li, Z. (2025). Shoal: Improving DAG-BFT latency and robustness. In Financial Cryptography and Data Security, FC 2024 (LNCS, Vol. 14744, pp. 92–109). Springer. https://doi.org/10.1007/978-3-031-78676-1_6.

10. Doidge, I., Ramesh, R., Shrestha, N., & Tobkin, J. (2024). Moonshot: Optimizing chain-based rotating leader BFT via optimistic proposals. arXiv. https://doi.org/10.48550/arXiv.2401.01791.

11. Chan, B. Y., & Shi, E. (2020). Streamlet: Textbook streamlined blockchains. In Proceedings of the 2nd ACM Conference on Advances in Financial Technologies (pp. 1–11). ACM. https://doi.org/10.1145/3419614.3423256.

12. Gueta, G. G., Abraham, I., Grossman, S., Malkhi, D., Pinkas, B., Reiter, M., Seredinschi, D.-A., Tamir, O., & Tomescu, A. (2019). SBFT: A scalable and decentralized trust infrastructure. In 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (pp. 568–580). IEEE. https://doi.org/10.1109/DSN.2019.00063.

13. Dwork, C., Lynch, N., & Stockmeyer, L. (1988). Consensus in the presence of partial synchrony. Journal of the ACM, 35(2), 288–323. https://doi.org/10.1145/42282.42283.

14. Lamport, L., Shostak, R., & Pease, M. (1982). The Byzantine generals problem. ACM Transactions on Programming Languages and Systems, 4(3), 382–401. https://doi.org/10.1145/357172.357176.

© Прокопович-Ткаченко Д.І., Пономарьов І.В., Бабенко К.Є. Операційне визначення та метрики наскрізної затримки безпечного доступу до стану блокчейн-систем першого рівня. Сучасний захист інформації, 3(67), 91–103. https://doi.org/10.31673/2409-7292.2026.033419

15. Gilad, Y., Hemo, R., Micali, S., Vlachos, G., & Zeldovich, N. (2017). Algorand: Scaling Byzantine agreements for cryptocurrencies. In Proceedings of the 26th Symposium on Operating Systems Principles (pp. 51–68). ACM. https://doi.org/10.1145/3132747.3132757.

16. Androulaki, E., Barger, A., Bortnikov, V., et al. (2018). Hyperledger Fabric: A distributed operating system for permissioned blockchains. In Proceedings of the Thirteenth EuroSys Conference (Article 30, pp. 1–15). ACM. https://doi.org/10.1145/3190508.3190538.

17. Crain, T., Natoli, C., & Gramoli, V. (2021). Red Belly: A secure, fair and scalable open blockchain. In 2021 IEEE Symposium on Security and Privacy (pp. 466–483). IEEE. https://doi.org/10.1109/SP40001.2021.00087.

18. Blackshear, S., Chursin, A., Danezis, G., et al. (2024). Sui Lutris: A blockchain combining broadcast and consensus. In Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security (pp. 2606–2620). ACM. https://doi.org/10.1145/3658644.3670286.

19. Chen, W., Xia, D., Cai, Z., Dai, H.-N., Zhang, J., Hong, Z., Liang, J., & Zheng, Z. (2024). Porygon: Scaling blockchain via 3D parallelism. In 2024 IEEE 40th International Conference on Data Engineering (pp. 1944–1957). IEEE. https://doi.org/10.1109/ICDE60146.2024.00153.

20. Gelashvili, R., Spiegelman, A., Xiang, Z., Danezis, G., Li, Z., Malkhi, D., Xia, Y., & Zhou, R. (2023). Block-STM: Scaling blockchain execution by turning ordering curse to a performance blessing. In Proceedings of the 28th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (pp. 232–244). ACM. https://doi.org/10.1145/3572848.3577524.

21. Boneh, D., Lynn, B., & Shacham, H. (2001). Short signatures from the Weil pairing. In Advances in Cryptology—ASIACRYPT 2001 (LNCS, Vol. 2248, pp. 514–532). Springer. https://doi.org/10.1007/3-540-45682-1_30.

22. Merkle, R. C. (1988). A digital signature based on a conventional encryption function. In Advances in Cryptology—CRYPTO '87 (LNCS, Vol. 293, pp. 369–378). Springer. https://doi.org/10.1007/3-540-48184-2_32.

23. Aptos Foundation. (2022). The Aptos blockchain: Safe, scalable, and upgradeable Web3 infrastructure. https://aptosfoundation.org/whitepaper/aptos-whitepaper_en.pdf.

24. Yakovenko, A. (2018). Solana: A new architecture for a high-performance blockchain. https://solana.com/solana-whitepaper.pdf.

25. Kaplan, E. L., & Meier, P. (1958). Nonparametric estimation from incomplete observations. Journal of the American Statistical Association, 53(282), 457–481. https://doi.org/10.1080/01621459.1958.10501452.

26. Fine, J. P., & Gray, R. J. (1999). A proportional hazards model for the subdistribution of a competing risk. Journal of the American Statistical Association, 94(446), 496–509. https://doi.org/10.1080/01621459.1999.10474144.

27. Buterin, V., & Griffith, V. (2017). Casper the friendly finality gadget. arXiv. https://doi.org/10.48550/arXiv.1710.09437.

Dmytro Prokopovych-Tkachenko
Candidate of Technical Sciences, Associate Professor, Head of the Department of Cybersecurity and Information Technologies
University of Customs and Finance, Dnipro, Ukraine
ORCID: 0000-0002-6590-3898
E-mail: prokopovich-tkachenko@umsf.dp.ua

Igor Ponomarev
PhD, Associate Professor, Department of Electronic Computing Machines
Oles Honchar Dnipro National University, Dnipro, Ukraine
ORCID: 0009-0009-7139-2885
E-mail: ponomarov_i@365.dnu.edu.ua

Kostiantyn Babenko
PhD Student, Department of Computer Science and Information Technologies, Faculty of Physics, Electronics and Computer Systems
Oles Honchar Dnipro National University, Dnipro, Ukraine
ORCID: 0009-0009-0945-7078
E-mail: babenko-k@365.dnu.edu.ua

OPERATIONAL DEFINITION AND METRICS OF END-TO-END LATENCY FOR SECURE ACCESS TO THE STATE OF LAYER-1 BLOCKCHAIN SYSTEMS

The article addresses the operational definition of the latency observed by an end user of a programmable layer-1 blockchain between submitting a signed transaction and the first confirmed read of the updated state through an RPC interface. The problem is relevant because consensus-finality metrics omit state indexing, publication, and client-side

© Прокопович-Ткаченко Д.І., Пономарьов І.В., Бабенко К.Є. Операційне визначення та метрики наскрізної затримки безпечного доступу до стану блокчейн-систем першого рівня. Сучасний захист інформації, 3(67), 91–103. https://doi.org/10.31673/2409-7292.2026.033419

discovery; they can therefore underestimate service availability and the interval during which a user may receive stale state. The study aims to formalize user-visible latency T_visible as a separate measurement object and to establish a reproducible system of companion indicators. The method combines a causal decomposition of the pipeline into admission, batching, consensus, execution, commit, indexing, publication, and discovery stages; a formal visibility predicate for certified state; separation of right-censored observations from competing terminal events; quantile estimation; and separation of polling discretization from systemic latency. In a numerical experiment calibrated against published measurements of pipelined blockchains, the median observed gap between finality and the first read was 168–241 ms, or up to 27% of total latency, while a conservative stale-response indicator at a 200 ms delay reached 0.69. A parallel pipeline with early state publication reduced median T_visible from 889 to 686 ms and the 1-second threshold violation rate from 17.7% to 0.2%. The scientific contribution is an integrated measurement protocol that combines a state-visibility predicate with visibility-gap, stale-response, and service-level violation metrics. The practical value is the ability to compare scenarios consistently, justify SLOs, and assess the availability and integrity of data exposed to end users.

Keywords: end-to-end latency, state visibility, layer-1 blockchain, RPC interface, cyber resilience, stale response, quality of service.

Надійшла до редакції (Received): 30.07.2026
Прийнята до друку (Accepted): 08.09.2026
Опубліковано онлайн (Available online): 15.09.2026

УДК 004.056.5:004.65 DOI: 10.31673/2409-7292.2026.035620

Рзаєва Світлана Леонідівна
кандидат технічних наук, доцент, доцент кафедри комп'ютерних наук
Київський столичний університет імені Бориса Грінченка, Київ, Україна
ORCID: 0000-0002-7589-2045
E-mail: s.rzaieva@kubg.edu.ua

Бондарчук Андрій Петрович
доктор технічних наук, професор, завідувач кафедри комп'ютерних наук
Київський столичний університет імені Бориса Грінченка, Київ, Україна
ORCID: 0000-0001-5124-5102
E-mail: a.bondarchuk@kubg.edu.ua

Костюк Юлія Володимирівна
доктор філософії, доцент, завідувач кафедри інформаційної та кібернетичної безпеки імені професора Володимира Бурячка
Київський столичний університет імені Бориса Грінченка, Київ, Україна
ORCID: 0000-0001-5423-0985
E-mail: y.kostiuk@kubg.edu.ua

Рзаєв Дмитро Олександрович
старший викладач кафедри інформатики та системології
Київський національний економічний університет імені Вадима Гетьмана, Київ, Україна
ORCID ID: 0000-0002-7149-4971
E-mail: rzaiev@kneu.edu.ua

Складанний Павло Миколайович
кандидат технічних наук, доцент, завідувач кафедри інформаційної та кібернетичної безпеки імені професора Володимира Бурячка
Київський столичний університет імені Бориса Грінченка, Київ, Україна
ORCID: 0000-0002-7775-6039
E-mail: p.skladannyi@kubg.edu.ua

© Прокопович-Ткаченко Д.І., Пономарьов І.В., Бабенко К.Є. Операційне визначення та метрики наскрізної затримки безпечного доступу до стану блокчейн-систем першого рівня. Сучасний захист інформації, 3(67), 91–103. https://doi.org/10.31673/2409-7292.2026.033419