Технічні науки

УДК 004.41:004.65

Лукашов Олександр Валентинович

магістр комп'ютерної інженерії,

асистент кафедри інженерії програмного забезпечення

Київський національний університет імені Тараса Шевченка

Lukashov Oleksandr

Master of Computer Engineering,

Assistant at the Department of Software Engineering

Taras Shevchenko National University of Kyiv

Шабатин Павло Євгенійович

магістр комп'ютерних наук,

асистент кафедри інженерії програмного забезпечення

Київський національний університет імені Тараса Шевченка

Shabatin Pavlo

Master of Computer Science,

Assistant at the Department of Software Engineering

Taras Shevchenko National University of Kyiv

DOI: https://doi.org/10.25313/2520-2057-2026-8-12107

НЕПЕРЕРВНА ВИБІРКОВА ПЕРЕВІРКА ДОСТУПНОСТІ ДАНИХ У СХОВИЩАХ ЗАГАЛЬНОГО ПРИЗНАЧЕННЯ
CONTINUOUS DATA AVAILABILITY SAMPLING IN GENERAL-PURPOSE STORAGE

Анотація. *Вибіркова перевірка доступності даних (DAS) дозволяє легким клієнтам переконуватися, що дані фактично зберігаються в*

мережі. Наявні формалізації DAS моделюють одноразову перевірку блоку; для сховищ загального призначення, де дані живуть роками, формальної моделі немає. Запропоновано модель неперервного DAS: деградація кодованого об'єкта та вибіркові запити — конкуруючі пуасонівські процеси. Виведено вираз для ймовірності невиявленої втрати, порогову умову q та межу запитів клієнта j*(ε).*

Ключові слова: *вибіркова перевірка доступності даних, децентралізовані сховища, надлишкове кодування, довговічність даних, пуасонівські процеси, бюджет вибірки.*

Summary. *Data availability sampling (DAS) allows light clients to verify that data is actually stored in the network. Existing formalizations model a one-time check of a block; for general-purpose storage, where data lives for years, no formal model exists. We propose a continuous DAS model: degradation of an encoded object and client sampling queries are competing Poisson processes. We derive the probability of undetected data loss, a threshold condition q*, and a per-client query bound j*(ε).*

Key words: *data availability sampling, decentralized storage, erasure coding, data durability, Poisson processes, sampling budget.*

Вступ. Децентралізовані сховища загального призначення – Filecoin, Storj, Walrus, Codex – захищають дані надлишковим кодуванням: об'єкт розбивається на блоки за схемою (n, k), і для відновлення достатньо будь-яких k < n блоків [1]. Проте кодування саме по собі не гарантує, що блоки *фактично зберігаються*: вузол може втратити дані внаслідок збою або зловмисно приховувати їх, відповідаючи на службові повідомлення, але не видаючи вміст. Для власника даних цю проблему розв'язують доведення зберігання – proofs of retrievability (PoR) [2] та provable data possession (PDP)

[3]; існують і публічно верифіковані варіанти [4], проте всі вони вимагають підготовки автентифікаторів власником під час завантаження й аудиту кожного файлу окремо – тобто не дають *стороннього* легкого клієнта без попереднього налаштування, здатного перевірити доступність довільних чужих даних.

Альтернатива – вибіркова перевірка доступності даних (DAS), запропонована для блокчейн-масштабування [5]: клієнт запитує випадкові блоки кодованих даних, і успіх невеликої кількості запитів дає високу впевненість у доступності цілого. DAS має формальні означення та доведення безпеки [6], розширення з локальним відновленням [7] та детальні симуляційні дослідження для Ethereum [8]. Однак уся ця лінія моделює *одноразову* подію: блок розповсюджено, клієнти перевірили його доступність, протокол завершився. У сховищі загального призначення постановка інша: об'єкт зберігається місяці й роки, вузли неперервно вибувають через плинність (churn), а приховування блоку небезпечне не саме по собі, а тим, що воно *залишається невиявленим*, поки надлишковість деградує. Системи, що вже застосовують вибіркові перевірки до довгоживучих даних – Codex із сублінійною вибіркою [9] та Walrus зі storage challenges [10], – обирають частоту перевірок евристично, без аналітичного зв'язку з довговічністю даних. Найближчий розгорнутий механізм неперервної перевірки – Filecoin Proof-of-Spacetime [11] – керується доводжувачем: провайдер сам періодично публікує доведення зберігання; у DAS натомість ініціатива належить клієнтам, і саме зв'язок частоти клієнтських перевірок із довговічністю даних у жодній із цих систем не аналізується.

Отже, відкритим є базове запитання: *якою має бути інтенсивність вибірки, щоб виявлення втрат встигало за їхнім накопиченням?*

Внески цієї роботи: - Формальна модель неперервного DAS: втрата і приховування блоків та вибіркові запити клієнтів – конкуруючі пуасонівські

процеси; виявлення відсутнього блоку запускає відновлення. Модель поєднує DAS-постановку [6] з марківським аналізом надійності кодованих сховищ [12]. - Наближений замкнений вираз для ймовірності невиявленої втрати P_loss(q) як функції сукупного бюджету вибірки q та порогова умова $q^* \approx n(n-k)/(\delta \cdot t_r)$, за якої DAS-моніторинг наближається до ідеального монітора з допуском δ. - Межа $j^*(\varepsilon) = n \cdot \ln(1/\varepsilon)/(n-k+1)$ на кількість запитів окремого клієнта для ε-впевненості в доступності об'єкта, з наслідком: сукупне мережеве навантаження визначається лише q і не зростає з кількістю клієнтів за агрегації вибірок.

Формальна модель

Система та модель загроз

Розглядаємо один об'єкт, закодований за схемою (n, k): n блоків розміщено на n різних вузлах, для декодування достатньо будь-яких k. Стан об'єкта $s \in \{k-1, k, \ldots, n\}$ – кількість фактично доступних блоків; $s = k - 1$ – поглинаючий стан втрати. Блок стає недоступним двома шляхами з сумарною інтенсивністю λ на блок: (i) *чесна відмова* – вузол залишає мережу або втрачає дані; (ii) *приховування* – вузол відповідає на службові повідомлення, але не видає блок на запит. Ключова асиметрія: обидві події *невидимі* для системи, доки хтось не спробує отримати блок. На відміну від блокчейн-DAS, де супротивник – виробник блоку в момент розповсюдження [5], тут супротивником є підмножина вузлів зберігання протягом усього життя об'єкта.

Процес вибірки та виявлення

Клієнти колективно генерують вибіркові запити до об'єкта як пуасонівський процес із сукупною інтенсивністю q запитів за одиницю часу; кожен запит адресує рівномірно випадковий блок із n. Відповіддю слугує мала автентифікована комірка блоку (наприклад, лист Merkle-дерева з доведенням належності [9]), тож вартість запиту значно менша за розмір блоку. Запит до недоступного блоку завершується невдачею, що *виявляє*

дефект. Час до виявлення конкретного відсутнього блоку розподілений експоненційно з середнім n/q. Після виявлення система запускає відновлення блоку з решти ≥ k доступних; час відновлення має середнє t_r. Отже, ефективна інтенсивність усунення одного дефекту обчислюється за формулою (1):

$$\mu_{\text{DAS}} = \frac{1}{n/q + t_r}. \quad (1)$$

Строго кажучи, послідовність «виявлення, потім відновлення» має фазовий (phase-type) розподіл; заміна його одним експоненційним розподілом із середнім n/q + t_r – стандартне наближення марківських моделей надійності [12]. Зауважимо також консервативність: коли відсутніх блоків m > 1, вибірка влучає в *будь-який* із них з інтенсивністю q·m/n > q/n, тож використання μ_{DAS} для всіх станів дає нижню оцінку інтенсивності виявлення і, відповідно, верхню оцінку ймовірності втрати.

Зв'язка з відновленням: ланцюг народження-загибелі

Аналогічно до марківських моделей надійності [12], система описується ланцюгом зі станами {k−1, …, n}: перехід s → s−1 (нова недоступність) з інтенсивністю $\alpha_s = s \cdot \lambda$; перехід s → s+1 (виявлення + відновлення, s < n) з інтенсивністю μ_{DAS}. Модель навмисно протокол-агностична: конкретний механізм доведення (Merkle-доведення, KZG-зобов'язання [6], ZK-доведення [9]) впливає лише на розмір запиту та константи t_r, але не на структуру процесу.

Межі бюджету вибірки

Ймовірність невиявленої втрати

Визначимо P_{loss} як ймовірність того, що *епізод деградації* – період від першої недоступності в повному стані n до повернення в n або поглинання – завершиться безповоротною втратою (станом k−1). Перший перехід n → n−1 відбувається без конкуренції з відновленням: у повному стані нема чого виявляти. Далі втрата вимагає ще n−k послідовних

недоступностей до першого завершення циклу «виявлення + відновлення» (разом n−k+1 відмов від n до k−1). У станах s ∈ {k, …, n−1} ймовірність переходу вниз становить s·λ/(s·λ + μ_DAS) ≈ s·λ/μ_DAS (наближення чинне при μ_DAS >> n·λ), і перемноження по станах s = n−1, …, k дає формулу (2):

$$P_{\text{loss}}^{\text{DAS}}(q) \approx \frac{(n-1)!}{(k-1)!} \cdot \left(\lambda \cdot \left(\frac{n}{q} + t_r\right)\right)^{n-k}. \qquad (2)$$

Вираз (2) явно розділяє два доданки вікна вразливості: затримку *виявлення* n/q, керовану бюджетом вибірки, та затримку *відновлення* t_r, задану інфраструктурою. При q → ∞ вираз збігається до ймовірності втрати системи з ідеальним монітором (миттєвим виявленням), для якої вікно дорівнює лише t_r.

Порогова умова q*

Ідеальний монітор виявляє недоступність миттєво (границя q → ∞), тож P_loss^ideal = ((n−1)!/(k−1)!)·(λ·t_r)^{n−k}. Визначимо надлишок ризику R(q) = P_loss^DAS(q) / P_loss^ideal = (1 + n/(q·t_r))^{n−k}; комбінаторний коефіцієнт скорочується. Вимагаючи R(q) ≤ 1 + δ для допуску δ << 1 і розкладаючи за малим n/(q·t_r), отримуємо формулу (3):

$$\boxed{q^* \approx \frac{n \cdot (n-k)}{\delta \cdot t_r}} \qquad (3)$$

– мінімальний сукупний бюджет вибірки, за якого DAS-моніторинг втрачає не більше ніж частку δ довговічності порівняно з миттєвим виявленням. Практичні наслідки: q* зростає лінійно з n при фіксованому запасі n−k (за пропорційної надлишковості n−k ∝ n зростання квадратичне), лінійно з n−k (більший запас означає, що ризик визначається глибшим хвостом добутку, чутливішим до затримки виявлення), і обернено пропорційний t_r – у системах зі швидким відновленням вузьким місцем стає саме виявлення, і вибірка мусить бути інтенсивнішою. Оскільки розклад утримує лише перший порядок за δ, формула трохи занижує

потрібний бюджет (похибка $O(\delta^2)$): наприклад, $R(q^*) \approx 1.105$ замість цільового 1.1 при $\delta = 0.1$, $n-k = 20$).

Впевненість окремого клієнта

Розглянемо тепер легкого клієнта, який разово перевіряє доступність об'єкта. Щоб зробити об'єкт невідновним, супротивник має приховати щонайменше $n-k+1$ блоків, тобто частку $\beta_0 = (n-k+1)/n$. Клієнт, що надсилає j незалежних рівномірних запитів, не влучає в жоден прихований блок з ймовірністю $(1-\beta_0)^j \leq \exp(-j\cdot\beta_0)$. Вимога, щоб ця ймовірність не перевищувала ε, дає формулу (4):

$$j^*(\varepsilon) = \frac{n\cdot\ln(1/\varepsilon)}{n-k+1}. \quad (4)$$

За формулою (4) для типової конфігурації $n = 30$, $k = 10$ та $\varepsilon = 10^{-9}$ достатньо $j^* \approx 30$ запитів; оскільки запит повертає малу автентифіковану комірку, а не блок цілком (§2.2), сумарний обсяг завантаження становить незначну частку розміру об'єкта – у цьому й полягає практичний сенс DAS для сховищ. Межа консервативна через експоненційну оцінку: точна геометрична умова $(1-\beta_0)^j \leq \varepsilon$ дає $j \approx 18$ за тих самих параметрів. Зауважимо розподіл ролей: межа $j(\varepsilon)$ *захищає* окремого* клієнта від приховування «тут і зараз», тоді як умова q^* з §3.2 гарантує *системну* довговічність у часі; перша не залежить від λ, друга – акумулює запити всіх клієнтів, тож за агрегації вибірок (наприклад, через gossip-поширення результатів [8], яке, втім, додає власне припущення про чесність ретрансляції) сукупне навантаження мережі визначається лише q^*, а не кількістю клієнтів.

Висновки. Ми запропонували першу, наскільки нам відомо, формальну модель вибіркової перевірки доступності даних як *неперервного* процесу для довгоживучих об'єктів у децентралізованих сховищах загального призначення – на противагу одноразовій блокчейн-постановці наявних формалізацій [6; 7]. Ключові результати – наближений замкнений вираз $P_{loss}(q)$, порогова умова $q^* \approx n(n-k)/(\delta\cdot t_r)$ та клієнтська межа $j^*(\varepsilon)$ –

пов'язують бюджет вибірки з довговічністю даних і дають аналітичну основу для калібрування частоти вибіркових перевірок у системах на кшталт Codex [9] і Walrus [10], де сьогодні її обирають евристично.

Обмеження. Модель передбачає незалежні відмови, пуасонівські процеси вибірки та відмов, рівномірну вибірку блоків і чесну доставку результатів запитів; корельовані відмови, атаки на мережевий рівень вибірки та стратегічного супротивника, що адаптує приховування до спостережуваної інтенсивності запитів, залишаються поза межами розгляду. **Майбутня робота:** валідація моделі симуляцією та на трасах відмов реальних мереж; нерівномірна вибірка, зважена за популярністю і віком об'єктів; ігрова постановка проти адаптивного супротивника; інтеграція з локальним відновленням у дусі [7].

Література

1. Li C. et al. SoK : Decentralized Storage Network. *High-Confidence Computing*. 2024. Vol. 4, No. 3. Art. 100239. DOI: 10.1016/j.hcc.2024.100239.

2. Juels A., Kaliski B. S. PORs : Proofs of Retrievability for Large Files. *14th ACM Conference on Computer and Communications Security (CCS '07)* : proceedings, Alexandria, 2007. P. 584–597.

3. Ateniese G. et al. Provable Data Possession at Untrusted Stores. *14th ACM Conference on Computer and Communications Security (CCS '07)* : proceedings, Alexandria, 2007. P. 598–610.

4. Shacham H., Waters B. Compact Proofs of Retrievability. *ASIACRYPT 2008* : LNCS. Springer, 2008. Vol. 5350. P. 90–107.

5. Al-Bassam M. et al. Fraud and Data Availability Proofs: Detecting Invalid Blocks in Light Clients. *Financial Cryptography and Data Security (FC '21)* : LNCS. Springer, 2021. Vol. 12675. P. 279–298.

6. Hall-Andersen M., Simkin M., Wagner B. Foundations of Data Availability Sampling. *IACR Communications in Cryptology*. 2025. Vol. 1, No. 4. Art. 34. DOI: 10.62056/a09qudhdj.

7. Boneh D. et al. Data Availability Sampling with Repair. *Cryptology ePrint Archive*. 2025. Paper 2025/1414. URL: https://eprint.iacr.org/2025/1414 (дата звернення: 07.07.2026).

8. Chaudhuri A. et al. On the Design of Ethereum Data Availability Sampling: A Comprehensive Simulation Study. *Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS 2024)* : proceedings, Berlin, 2024. P. 1–4.

9. Codex: Decentralised Durability Engine : whitepaper. *Codex Storage*. 2023. URL: https://docs.codex.storage/learn/whitepaper (дата звернення: 07.07.2026).

10. Danezis G. et al. Walrus: An Efficient Decentralized Storage Network. *arXiv*. 2025. arXiv : 2505.05370. URL: https://arxiv.org/abs/2505.05370 (дата звернення: 07.07.2026).

11. Filecoin : A Decentralized Storage Network : whitepaper. *Protocol Labs*. 2017. URL: https://filecoin.io/filecoin.pdf (дата звернення: 07.07.2026).

12. Greenan K. M., Plank J. S., Wylie J. J. Mean Time to Meaningless: MTTDL, Markov Models, and Storage System Reliability. *2nd USENIX Workshop on Hot Topics in Storage and File Systems (HotStorage '10)* : proceedings, Boston, 2010. P. 1–5. URL: https://www.usenix.org/legacy/event/hotstorage10/tech/full_papers/Greenan.pdf (дата звернення: 13.07.2026).