https://doi.org/10.31891/2307-5732-2026-365-7
УДК 004.75:004.9

ГНАТЮК ВІКТОР
Державний університет «Київський авіаційний інститут»; ДержНДІ технологій кібербезпеки
https://orcid.org/0000-0002-4916-7149
e-mail: viktor.hnatiuk@npp.kai.edu.ua
ЗАНДЕР КОСТЯНТИН
Державний університет «Київський авіаційний інститут»
https://orcid.org/0009-0006-4944-9249
e-mail: 8390983@stud.kai.edu.ua

МЕТОДИ ПІДВИЩЕННЯ РЕАКТИВНОСТІ В ІНТЕРАКТИВНИХ ІНТЕРНЕТ-СИСТЕМАХ

У статті розглянуто методи підвищення реактивності інтерактивних Інтернет-систем у процесі надання послуг. Проаналізовано чинники, що впливають на середній час реакції та дисперсію затримок відповідей. Запропоновано представлення реального трафіку запитів як фрактального дискретного квазістаціонарного випадкового процесу з передбачуваними статистичними характеристиками та складовою непередбачуваних сплесків. Розроблено модель пакетного Інтернет-трафіка, метод його обслуговування засобами реалізації телекомунікаційних протоколів, а також структурно-функціональну модель інтерактивної системи як системи масового обслуговування. Запропоновано метод формування трафіка звернень, який розділяє потік на передбачувану і непередбачувану складові та перетворює нестаціонарний трафік у квазістаціонарний, що дозволяє оптимізувати параметри обслуговування й нейтралізувати вплив сплесків. Результати можуть бути використані для підвищення стабільності та якості роботи прикладних Інтернет-систем в умовах нерівномірного навантаження.

Ключові слова: *інтерактивна Інтернет-система, реактивність, час відгуку, дисперсія затримок, пакетний трафік, система масового обслуговування.*

GNATYUK VIKTOR
State University "Kyiv Aviation Institute"; ICTIP
ZANDER KOSTIANTYN
State University "Kyiv Aviation Institute"

METHODS OF INCREASING REACTIVITY IN INTERACTIVE INTERNET SYSTEMS

This paper addresses the problem of improving the reactivity of interactive Internet systems in the context of service delivery. Reactivity is considered as a key quality metric defined by the average system response time and the variance of response delays under given usage conditions and resource constraints. A comprehensive analysis of influencing factors is presented, including network transmission delays, queuing times, and server-side processing times. It is proposed to model the real request traffic as a fractal discrete quasi-stationary stochastic process with predictable statistical parameters (mean and variance) combined with an unpredictable component representing burst traffic. This approach enables the system to be treated as a nonlinear dynamic system operating under sudden, non-predictable excitations, requiring optimal real-time parameter control to maintain stability within a specified burst range.

A packet Internet traffic model is developed, along with a method for its processing using telecommunication protocol implementations in server hardware. The paper further introduces a structural-functional model of the interactive system represented as a queuing system (mass service system) that separates incoming traffic into predictable and unpredictable components. For the predictable component, parameter optimization is carried out to meet target reactivity values, while the unpredictable bursts are mitigated through load balancing across processing servers.

Additionally, a request traffic shaping method is proposed, capable of converting a non-stationary traffic flow into a sequence of quasi-stationary segments with limited variance. The method employs a buffer-switching mechanism and an adaptive speed control system to smooth traffic fluctuations, reduce burst impact, and maintain service stability without excessive resource allocation. The proposed models and methods can be applied to the design and operation of interactive Internet systems that require high responsiveness and reliability under conditions of irregular and bursty traffic patterns. The results are relevant for improving Quality of Service (QoS) in a wide range of networked applications, including web-based platforms, cloud services, and real-time interactive systems.

Keywords: *interactive Internet system, reactivity, response time, delay variance, packet traffic, queuing system.*

Стаття надійшла до редакції / Received 11.02.2026
Прийнята до друку / Accepted 11.03.2026
Опубліковано / Published 28.05.2026

Постановка проблеми

Інтерактивні Інтернет-системи, які забезпечують надання послуг користувачам у режимі запит–відповідь, мають функціонувати з високими показниками реактивності навіть в умовах нерівномірного та пульсуючого трафіка звернень. Реактивність у цьому контексті визначається середнім часом відгуку системи на запити клієнтів та стабільністю цього часу, тобто величиною дисперсії затримок відповідей. Погіршення цих показників призводить до зниження якості обслуговування (QoS), втрати користувачів та зниження ефективності використання ресурсів.

Сучасні умови експлуатації інтерактивних інтернет-систем характеризуються наявністю складних і непередбачуваних сценаріїв навантаження, серед яких значну частку становлять різкі сплески (burst) трафіка, що можуть виникати випадковим чином і мати значну амплітуду. Традиційні методи оптимізації роботи систем орієнтовані на середні значення навантаження і не враховують непередбачувану складову трафіка, внаслідок чого

в пікові моменти відбувається різке зростання часу відгуку та збільшення втрат запитів.

Складність задачі підвищення реактивності інтерактивних інтернет-систем полягає в тому, що реальний потік запитів доцільно розглядати як суперпозицію двох компонент: передбачуваної квазістаціонарної складової з визначеними статистичними параметрами; непередбачуваної складової у вигляді сплесків трафіка з невідомими характеристиками.

Для підтримання заданих значень реактивності необхідно здійснити параметричну оптимізацію системи обслуговування запитів щодо передбачуваної складової трафіка, а вплив непередбачуваної — мінімізувати за допомогою механізмів адаптивного управління та згладжування потоку. При цьому важливо забезпечити мінімальні витрати обчислювальних і мережевих ресурсів, зберігаючи стабільність функціонування системи у широкому діапазоні змін навантаження.

Таким чином, проблема полягає у розробці моделей і методів, які дозволять: формалізувати вплив окремих чинників на показники реактивності інтерактивних Інтернет-систем; синтезувати моделі трафіка, що відображають як передбачувану, так і непередбачувану складові; розробити методи обслуговування та формування трафіка, які зменшують негативний вплив сплесків на реактивність і забезпечують стабільну якість обслуговування в умовах нестабільного навантаження.

Аналіз останніх джерел

Останні роки характеризуються переходом від суто мережевих оптимізацій до наскрізних (end-to-end) підходів, що об’єднують транспорт/протоколи, архітектуру розміщення сервісів і фронтенд-рішень. На транспортному рівні ключовою віхою стало впровадження QUIC і HTTP/3, які завдяки Zero-RTT/1-RTT встановленню з’єднань, мультиплексуванню без head-of-line blocking та інтегрованому шифруванню знижують мережеву латентність і підвищують стійкість при втраті пакетів [1–3]. Паралельно стандартизована схема пріоритизації ресурсів для HTTP (EPS) дала змогу точніше керувати порядком доставки критичних об’єктів сторінки; експериментальні роботи для HTTP/3 показали підвищення QoE за рахунок зваженої інкрементальної доставки контенту, що фіксується інструментами Lighthouse [4, 5]. Для мультимедійних сценаріїв активно досліджуються низьколатентні транспорти поверх HTTP/3 (WebTransport) та методики тестування HTTP/3-стримінгу в реалістичних умовах [6, 7].

На рівні інфраструктури домінує тренд наближення обчислень до користувача: multi-access edge computing (MEC) і edge-cloud архітектури систематично демонструють зменшення затримок завдяки кращому розміщенню сервісів, offloading’у та міграції стану/сервісів [8–12]. Огляди ACM CSUR узагальнюють механізми offloading’у, кешування, алокації ресурсів і мобільності, що прямо впливають на реактивність у сценаріях IoT/5G/6G [8, 11]. Окремі праці фокусуються на мінімізації e2e-латентності через спільну оптимізацію комунікацій/обчислень/кешування [9], а також на швидкій міграції стану для безперервної реакції сервісів при мобільності клієнтів [12].

На фронтенді наукова й прикладна спільнота зміщує акцент від суто «швидкого старту» до стабільної реактивності протягом усього життєвого циклу взаємодії користувача з інтерфейсом. У 2024 р. Interaction to Next Paint (INP) офіційно замінив FID у складі Core Web Vitals і став основним метричним орієнтиром саме для реактивності інтерфейсу [13–16]. Це стимулювало дослідження/практики щодо усунення «довгих задач» головного потоку, пріоритизації завантаження критичних ресурсів (fetchpriority/«Priority Hints»), а також оптимізації рендерингу за рахунок пропуску оф-скрін-вмісту (CSS content-visibility) [17–21]. Результати прикладних досліджень показують, що точне застосування пріоритизації ресурсів може помітно знижувати LCP/INP для великих проєктів, тоді як content-visibility дає кратні виграші для довгих сторінок за рахунок відкладеного рендерингу [19–21].

У роботі [22] запропоновано фреймворк продуктивного тестування веб-додатків, що базується на реактивності. Суть підходу — автоматичне формування тестових сценаріїв на основі веб-логів. Система витягує шаблони поведінки користувачів на сервері, застосовує метрики з боку клієнта, і на основі цих даних створює сценарії тестування із застосуванням еволюційних алгоритмів. Такий підхід спрямований на зниження витрат і підвищення ефективності тестування реакції статичних функцій у реальному веб-застосунку.

Діяс та співавт. в 2018 році представили підхід до розробки IoT-систем, що поєднує реактивне програмування та моделювання [23]. Метою є зменшення складності життєвого циклу IoT-систем, що характеризуються великомасштабністю, гетерогенністю та динамічною архітектурою. Модельно-орієнтований підхід дозволяє абстрагувати низькорівневу реалізацію, спрощуючи управління системами-of-systems.

Робота [24] зосереджена на підвищенні продуктивності UI оновлень завдяки поєднанню реактивного програмування, Virtual DOM та централізованої системи управління станом. Запропоновано механізм вибіркового оновлення тільки тих UI-компонентів, чиї дані змінилися. Такий підхід зменшує навантаження на браузер, підвищує швидкість відгуку і підтримує складну логіку взаємодії користувача без надлишкових перерисовок.

У роботі [25] проведено аналіз продуктивності SPA-додатків, що використовують Virtual DOM і реактивні UI. Дослідження показує, що хоча створення структури Virtual DOM займає більше часу на етапі завантаження, подальші операції над UI значно ефективніші, ніж при роботі з нативними методами. Реактивне програмування у JavaScript демонструє потенціал для створення масштабованих веб-додатків.

Узагальнюючи, сучасні підходи до підвищення реактивності інтерактивних Інтернет-систем є багаторівневими: (1) на транспорті — QUIC/HTTP/3 та коректна пріоритизація ресурсів; (2) на архітектурі — розміщення на краю мережі з підтримкою динамічної міграції/кешування; (3) на фронтенді — оптимізація

головного потоку, рендерингу та управління критичним шляхом завантаження під метрики INP/LCP. Водночас публікації підкреслюють важливість контексту: ефект від конкретної техніки (напр., пріоритизації чи інкрементальної доставки) залежить від структури сторінки, патернів взаємодії користувачів і мережевих умов. Саме комбінація зазначених методів, керована реальними метриками (RUM) і профілюванням, наразі розглядається як найефективніша стратегія досягнення стабільної низької латентності реакції.

Мета статті — розробити та обґрунтувати методи підвищення реактивності інтерактивних Інтернет-систем в умовах нерівномірного та пульсуючого трафіка, засновані на моделюванні потоку звернень як фрактального дискретного квазістаціонарного процесу з урахуванням передбачуваних і непередбачуваних складових, а також на синтезі механізмів обслуговування та формування трафіка, що забезпечують оптимізацію параметрів обслуговування та нейтралізацію впливу сплесків на час відгуку й стабільність роботи системи.

Виклад основного матеріалу

Визначення чинників, що впливають на реактивність інтерактивних систем. Однією з цілей дослідження є розробка методів побудови інтерактивних прикладних Інтернет-систем, здатних стабільно забезпечувати високі показники реактивності в реальних умовах експлуатації. Основними показниками прийнято: τ_{Ξ} — середнє значення часу реакції системи на клієнтські запити при заданих умовах і ресурсних обмеженнях; $D_{(\tau)}$ — дисперсія затримок відповідей τ відносно їх середнього значення.

Реактивність системи визначається сукупністю чинників:

$$\tau_{\Xi}=\tau_{кп1}+\tau_{чрг}+\tau_{пс}+\tau_{ко1}+\tau_{кп2}+\tau_{соз}+\tau_{ко2}, \quad (1)$$

де $\tau_{кп1}$ — передавання PDU із запитом клієнта від АРМ до вузла пошукових серверів ПС, включно з обробкою на цьому вузлі; $\tau_{чрг}$ — очікування запиту в черзі до обробки визначеним пошуковим сервером; $\tau_{пс}$ — обробка запиту пошуковим сервером ПС; $\tau_{ко1}$ — передавання PDU з відповіддю ПС до АРМ клієнта, включно з обробкою на його обладнанні; $\tau_{кп2}$ — передавання PDU із запитом до сервера обробки звернень (СОЗ), включно з обробкою на його обладнанні; $\tau_{соз}$ — обробка запиту на сервері СОЗ; $\tau_{ко2}$ — передавання PDU з відповіддю СОЗ до АРМ клієнта, включно з обробкою на його обладнанні.

Завдання полягає у вимірюванні цих складових та розробці методу оптимізації параметрів і структури системи для підвищення реактивності, особливо за умов значних пульсацій (Burst) непрогнозованого трафіка та зменшення їх негативного впливу на стабільність роботи.

Реактивність інтерактивної системи значною мірою залежить від характеристик потоку клієнтських запитів, адекватна модель якого в літературі практично відсутня. У даній роботі реальний трафік розглядається як фрактальний дискретний квазістаціонарний випадковий процес із довільною густиною ймовірності та визначеними математичним очікуванням і дисперсією, на який накладаються різкі випадкові сплески (Burst), статистично непередбачувані.

За таких умов система розглядається як нелінійна динамічна система, що функціонує при різких збуреннях і потребує оптимального керування параметрами в реальному часі для забезпечення сталості при мінімальних витратах на пропускну спроможність серверного обладнання.

Цільова функція параметричної оптимізації:

$$min \sum_{i=1}^{n} \Delta F_i, \quad (2)$$

де ΔF_i — продуктивність і-го пошукового сервера, n — їх кількість.

Обмеження:

$$\tau_{чрг} \leq \tau_{0чрг}, \tau_{пс} \leq \tau_{0пс}, P(t_0 > \tau_{0чрг}), \quad (3)$$

де $\tau_{0чрг}$, $\tau_{0пс}$ — максимально допустимі значення часу очікування у черзі та обробки; t_0 — час очікування.

Визначення характеристик квазістаціонарної складової трафіка та її моделювання в термінах теорії телетрафіка дозволяє ідентифікувати систему як певний клас систем масового обслуговування (СМО), підібрати параметри для підтримки заданих показників реактивності та оцінити дисперсію затримок $D(\tau)$, необхідну для забезпечення сталості.

Непередбачувані сплески трафіка знижують стабільність системи, оптимізованої під передбачувану складову. Для компенсації цього впливу пропонується постановка крайової задачі з використанням функціонала Беллмана для знаходження параметрів адаптивного керування, що мінімізують його значення.

Для реалізації підходу необхідно: синтезувати модель потоку запитів як суперпозицію квазістаціонарної складової та непередбачуваних сплесків; розробити метод обслуговування пакетного трафіка засобами телекомунікаційних протоколів серверів; створити структурно-функціональну модель системи як СМО; синтезувати метод формування трафіка, придатного для обслуговування прикладними засобами; опрацювати квазістаціонарну складову з мінімальними витратами ресурсів при дотриманні обмежень; розробити метод обробки сплесків трафіка для підтримки сталості системи у визначеному діапазоні.

Синтез моделей пакетного трафіка та методів їхнього обслуговування засобами реалізації телекомунікаційних протоколів серверного обладнання інтерактивних систем. Якщо розглядати структурно-функціональну схему інтерактивної системи, то можливо виділити наступні три види потоків інформації, характеристики котрих впливають на показники реактивності цієї системи: потоки сигналів, що розповсюджуються дуплексними фізичними каналами зв'язку між портами клієнтських АРМів та вузлового обладнання мережі Інтернет; потоки протокольних блоків даних (*PDU*) телекомунікаційних систем, що транспортують запити і звернення клієнтів та відповіді на них (тобто, Інтернет-трафік пакетованих даних); потоки звернень, запитів та відповідей на них, що оброблюються програмним забезпеченням прикладного рівня.

Оцінка впливу засобів передавання фізичних сигналів на реактивность інтерактивних систем. Щодо потоків сигналів через фізичні канали зв'язку слід зазначити, що проміжки часу передавання сигналів між вузлами мережі малі у порівнянні із проміжками часу $\tau_{кп1}$, $\tau_{ко1}$, $\tau_{кп2}$ та $\tau_{ко2}$, на протязі яких здійснюється передавання *PDU* через канали зв'язку і їхня обробка у вузлах мережі Інтернет. Так що, вплив цих потоків на величину показника реактивності інтерактивної системи τ_{Σ} слід вважати несуттєвим.

Модель мережного Інтернет-трафіка запитів та звернень. Зазвичай в цілях моделювання використовують такі закони розподілу потоку пакетів як експоненціальний (показовий) та ерлангівський (порядку k). Для моделювання пакетного трафіку нерідко застосовують розподіл Парето. В теорії телетрафіка розглядаються також властивості таких розподілів як регулярний (детермінований), довільний, довільний щодо незалежних інтервалів між запитами та гіперекспоненціальний. Вищезазначені закони розподілу витікають із узагальненого гама-розподілу, густина ймовірності якого має наступний вигляд:

$$f(t) = \frac{b^a}{\Gamma(a)} t^{a-1} e^{-bt}, \tag{4}$$

де $\Gamma(a)$ – гама-функція; a, b – параметри розподілу.

Для гама-розподілу, якщо узяти параметр $a \to \infty$, то інтервали часу між запитами τ_0 будуть вважатися детермінованими. Для гама-розподілу математичне очікування та дисперсія τ_0 визначаються як $M(\tau_0) = a/b$ та $D(\tau_0) = a/b^2$ відповідно, звідки $a = [M(\tau_0)]^2/D(\tau_0)$ та $b = M(\tau_0)/D(\tau_0)$.

Вираз (4) перетворюється на показовий (експоненціальний) розподіл при $a = 1$. Зокрема, густина розподілу ймовірностей проміжків часу між запитами (зверненнями) $f(z)$ згідно показового закону розподіляється як

$$f(z) = \frac{dF(z)}{dz} = \lambda e^{-\lambda z}, z \geq 0; \tag{5}$$

λ – інтенсивність потоку.

А математичне очікування та дисперсія для показового розподілу визначають як

$$M(z) = 1/\lambda; D(z) = 1/\lambda^2. \tag{6}$$

Основна властивість експоненціального (показового) закону - відсутність післядії. Тому для моделювання потоку запитів доцільно використати саме показовий закон розподілу.

Імовірність появи рівно k запитів на протязі часу t для зазначеного вище потоку визначається за формулою Пуассона:

$$P_k(t) = \frac{(\lambda t)^k}{k!} e^{-\lambda t}. \tag{7}$$

А розподіл проміжків часу Δt між запитами у такому потоці підкоряється експоненціальному закону:

$$P(\Delta t \leq z) = F(z) = 1 - e^{-\lambda z}, \tag{8}$$

Важливим узагальненням показового закону є закон розподілу Ерланга. Якщо у якості параметра a у виразі (4) взяти будь-яке ціле число, то щільність розподілу Ерланга k-го порядку буде мати наступний вигляд:

$$f(t) = \frac{\lambda(\lambda t)^{k-1}}{(k-1)!} e^{-\lambda t}, t \geq 0, \tag{9}$$

де $\lambda = 1/M(\tau_0)$ – параметр даного закону розподілу, $M(\tau_0)$ – математичне очікування тривалості проміжків між запитами τ_0, $M(\tau_0) = k/\lambda$, $D(\tau_0) = k/\lambda^2$.

Іншим узагальненням показового розподілу, що іноді використовують для моделювання телефонних викликів, є гіперекспоненціальний розподіл порядку k:

$$f(t) = \sum_{i=1}^{k} p_i \lambda_i e^{-\lambda_i t}; \sum_{i=1}^{k} p_i = 1; p_i \geq 0. \tag{10}$$

Для моделювання багатьох видів пакетного трафіка, що мають фрактальні властивості, знайшов використання розподіл Парето, оскільки ймовірнісним характеристикам цього розподілу притаманні так звані «важкі хвости». Функція розподілу Парето має наступний вигляд:

$$f_П(t) = \frac{\alpha}{\beta}\left(\frac{\beta}{\beta+t}\right)^{\alpha+1}, \alpha > 0. \tag{11}$$

Параметр β у розподілі (11) характеризує мінімально можливе значення випадкової величини t. Параметр α визначає середнє значення $[\alpha/(\alpha-1)] \cdot \beta$ і дисперсію випадкової величини. Коли $\alpha \leq 2$, розподіл має нескінченну дисперсію, а при $\alpha \leq 1$ він має ще й нескінченне середнє значення.

Дискретним аналогом розподілу Парето є дзета-розподіл

$$P\{t = k\} = ck^{-(n+1)}, \text{ де } c = \left[\sum_{k=1}^{\infty} k^{-(n+1)}\right]^{-1}, \tag{12}$$

щодо якого є справедливим наступне твердження: статистичні моменти випадкових величин з номерами порядку, що рівні або більші ніж n є нескінченними.

В теорії телетрафіку розглядаються й інші види розподілів випадкових величин (такі як рівномірний, нормальний, логаріфмічно-нормальний, Вейбула та ін.), але підстав для їхнього використання для моделювання трафіка запитів в інтерактивних системах у наявних публікаціях не виявлено.

Результати аналізу існуючих моделей пакетного трафіка показують, що потоки *PDU* на вводі серверної частини обладнання інтерактивної системи доцільно представити у вигляді так званої ON/OFF – моделі. Згідно цієї моделі потік *PDU* із зверненнями клієнтів уявляється як часова послідовність випадковим чином накладених один на одного фрагментів пакетованих даних, що просуваються каналами Інтернет між цим обладнанням та АРМами клієнтів інтерактивної системи. ON/OFF – процес розглядається як послідовні відрізки фрактального процесу – одного із різновидів дискретного фрактального точкового процесу (Fractal Point Process – FPP).

Об'єднання фрагментів в агрегований потік фрагментів утворюється виходячи із умови, що клієнти діють в системі незалежно один від одного. Кожен окремий фрагмент такого потоку розглядається як часовий відрізок FPP, який з фізичної точки зору являє часову послідовність пакетів з даними, що генерується конкретним активним клієнтом на протязі одного сеансу його взаємодії з інтерактивною системою. Типова процедура цієї взаємодії передбачає обмін «пачками» пакетів між клієнтом і сервером. При цьому проміжки часу, коли клієнт генерує запити і передає їх в канал у вигляді певним чином сформованих послідовностей пакетів кінцевої довжини, змінюються на проміжки часу, коли клієнт не виявляє активності, а очікує, поки серверна частина прикладної програми формує відповіді на ці запити і передає їх в канал у вигляді кінцевих послідовностей пакетів на адресу цього клієнта. Отже, у цій моделі трафік розглядається як комбінація джерел, які його генерують. Кожне джерело має наступну структуру. В ON – періоди часу джерело може генерувати пакети, при цьому всередині одного періоду пакети просуваються з однаковими інтервалами між ними. Після ON – періоду настає OFF-період, коли джерело не генерує пакети. Тривалість ON – та OFF-періодів є випадковою величиною, яка має кінцеве математичне очікування та безкінечну дисперсію. Структуру фрагмента доцільно моделювати у вигляді дискретного фрактального ON/OF-процесу, що відображає ситуацію, коли клієнти у випадкові проміжки часу (так звані, ON-періоди) генерують пакети з даними запитів, після яких йдуть періоди очікування відповідей, коли клієнти не генерують пакети (OF-періоди). Реалістично вважати, що всередині ON-періодів часові інтервали між пакетами є незмінними, тривалість яких залежить від характеристик обладнання доступу клієнта до мережі Інтернет. Проте тривалість ON-періоду і, от же, кількість пакетів у «пачці», що передаються на протязі цього періоду, доцільно вважати випадковою величиною. Випадковою величиною слід вважати і тривалість OF-періодів. Якщо статистичні характеристики названих вище випадкових величин вказують на існування масштабної інваріантності, то модель фрагменту розглянутої вище структури можливо класифікувати як самоподібну.

Розробка методу обслуговування пакетного трафіку засобами реалізації телекомунікаційних протоколів серверного обладнання інтерактивних систем. На обробку пакетного трафіка засобами серверного обладнання інтерактивної системи потрібен час, тривалість якого має бути оцінена з точки зору його впливу на показники реактивності цієї системи. Тому роботу цих засобів представимо як систему масового обслуговування (СМО) з дискретним часом $t \in \{..., -1, 0, 1, 2, ...\}$. СМО має моделювати роботу телекомунікаційних протоколів чотирьох нижніх рівнів взаємодії відповідно до класифікатора семирівневої моделі OSI ISO. Будемо вважати, що серверне обладнання має багатопортове уведення. Через ці порти просуваються потоки пакетів із запитами від клієнтів. Тобто, маємо v обслуговуючих пристроїв (v каналів обслуговування) та буферну пам'ять, яка може бути використана для організації черг пакетів у випадках, коли обслуговуючі пристрої знаходяться у стані «зайнято».

Теоретичні засади СМО широко висвітлені у численних публікаціях. Прийнято користуватися символічними позначеннями класів СМО у вигляді A/B/C/D/E/F, де: A – закон розподілу потоку запитів (звернень) на їхнє обслуговування серверним обладнання інтерактивної системи; B – закон розподілу часу обслуговування запитів (звернень); C – кількість елементів серверного обладнання, що опрацьовують запити (звернення); D – ємність буферної пам'яті, що призначена для створення черг у доступі до ресурсів серверного обладнання; E – кількість потоків запитів (звернень), тобто джерел навантаження на СМО (або порядок обслуговування черг – прямий, інверсний або випадковий); F – пріоритетність обслуговування.

Позначення характеристик кожного із наведених вище елементів прийнятого класифікатора, що у сукупності утворюють певний клас СМО, надано у табл.1.

Таблиця 1

Характеристики СМО

Характеристика СМО	Умовне позначення
Позначення закону розподілу моментів появи запитів (звернень) на обслуговування у трафіку запитів (звернень) (тобто, позначення характеристики першого елементу A класу СМО, що є об'єктом розгляду) або позначення закону розподілу часу обслуговування (тобто, позначення характеристики другого елементу B класу СМО, що є об'єктом розгляду):	
- експоненціальний (показовий)	M
- регулярний (детермінований)	D
- ерлангівський (порядку k)	E_k
- довільний	G
- довільний щодо незалежних інтервалів між запитами	GI
- гіперекспоненціальний	H_K
Кількість елементів обслуговуючого серверного обладнання (тобто, позначення характеристики третього елементу C класу СМО, що є об'єктом розгляду)	v, V
Ємність накопичувача запитів (звернень), що утворюють чергу (тобто, позначення характеристики четвертого елементу D класу СМО, що є об'єктом розгляду)	r; k;
Порядок обслуговування:	d
прямий	d_1
інверсний	d_2
випадковий	d_3
Кількість джерел навантаження на СМО	m
Пріоритетність обслуговування:	f; p;
- при постановці запитів у чергу (f^0 - обслуговування без пріоритетів)	f^i
- при звільненні запитів із черги (f_0 - обслуговування без пріоритетів)	f_i

Будемо вважати, що СМО обслуговує ON/OFF трафік $Y = (\dots, Y_{-1}, Y_0, Y_1, \dots)$, який утворюється АРМами клієнтів, що розглядаються в якості незалежних джерел генерації пакетів. Припустимо, що кожне джерело на інтервалах активності утворює потік пакетів з постійною швидкістю $R \in N$. Якщо вважати, що s-те джерело у довільний момент часу t_s почне генерувати пакети тривалістю 1 одиниця часу, то на протязі інтервалу «ON"» тривалістю τ_s джерело зможе згенерувати $\theta_s(\tau_s)$ пакетів. Якщо вважати, що кількість джерел, для яких $t_s = t$, складає ε_t, то загальна кількість утворених пакетів Y_t від ε_t джерел у момент *t* буде дорівнювати:

$$Y_t = \sum_{s \in Z} \theta_s(t - t_s + 1), t \in Z \quad .(13)$$

Дана модель СМО передбачає, що інтервали активності τ_s утворюють послідовність незалежних і однаково розподілених величин $\theta_s(i) \in Z$, де $\theta_s(i)$ - кількість пакетів у момент $t_s + i - 1$.

Будемо вважати, що величини ε_t теж є незалежними і розподіленими згідно закону Пуасона з інтенсивністю λ, тобто:

$$P_r(\varepsilon_t = k) = \frac{(\lambda)^k}{k!} e^{-\lambda}, 0 < \lambda < \infty. \quad (14)$$

Два основних параметра СМО, які впливають на якість обслуговування пакетного трафіка, - це кількість каналів його обслуговування v (тобто, кількість фізичних портів серверного обладнання, що задіяні в обробці трафіка) та ємність буферної пам'яті b, де мають зберігатися пакети в черзі на обслуговування. За цих умов на часовому інтервалі $(t, t+1)$ СМО буде здатна обслужити не більше *v* пакетів, які потрапляють на її вхід або безпосередньо із мережі Інтернет або із буферу. Якщо поточну кількість пакетів, що потрапляють на вхід СМО на інтервалі $(t, t+1)$, позначити як Y_t, то величина відношення $\frac{Y_t}{R}$ буде мати пуасонівський розподіл із середнім значенням Y_t, що визначається наступним виразом:

$$\overline{Y_t} = \lambda R \overline{\tau}, \quad (15)$$

де λ −інтенсивність появи джерел пакетів; $\overline{\tau}$ - середня тривалість активного інтервалу джерела пакетів.

Отже, маючи на увазі дані табл.1, розглянута СМО відноситься до класу $Y/D/v/b$. Порядок обслуговування потоку є наступним: якщо кількість пакетів у системі в момент *t* до появи Y_t нових пакетів позначити як Z_t, то при умові $Y_t + Z_t \le b + v$ жоден пакет у момент *t* не втрачається; якщо $Y_t + Z_t > b + v$, то $Y_t + Z_t - b - v$ пакетів у момент *t* буде втрачатися.

Переповнення буферу у момент *t* станеться, якщо виникне подія $\{Y_t + Z_t - b - v > 0\}$, коли буде втрачено хоча б один пакет. Імовірність переповнення буфера $P_{П}$ для різних видів пакетного трафіка визначена у багатьох дослідженнях. Інтерес являє випадок розподілу тривалості інтервалів активності τ за законом Парето з параметром α, щодо якого імовірність переповнення буфера $P_{П}$ визначиться як

$$P_{П} \approx \gamma_0 b^{(1-\alpha)k}, \quad k = 1 + \left\lfloor v/R - \lambda\overline{\tau} \right\rfloor, \quad v > \lambda R\overline{\tau}, \quad (16)$$

де $\lfloor x \rfloor$ - ціла частина числа x ; γ_0 - деяка функція від *R*, *v*, λ та α .

Відповідно з даними досліджень для випадку *v=R=1* для значень $\alpha = 1{,}3 \div 1{,}8$ значення $\gamma_0 = 11 \div 23$.

Вираз (16) показує, що імовірність $P_{П}$ зменшується по ступеневому закону в залежності від кількості каналів обслуговування *v*. При цьому показник ступеню є пропорційним величині $(v - \lambda R\overline{\tau})$ − перевищенню кількості каналів обслуговування v над інтенсивністю навантаження $\lambda R\overline{\tau}$. Так що, усувати можливість переповнення буферу краще шляхом підвищення кількості каналів обслуговування, ніж збільшенням ємності буферу.

У СМО з розподілом тривалості інтервалів активності τ за законом Парето, середня довжина черги визначається згідно наступного виразу:

$$q = \frac{\rho^{\frac{1}{[2(1-H)]}}}{(1-\rho)^{\frac{H}{(1-H)}}}, \quad (17)$$

де ρ - коефіцієнт навантаження, H – параметр Херста.

У той же час середня довжина черги у СМО класу M/M/1 або M/D/1, де обслуговуються класичні види трафіка, визначається виразом

$$q = \frac{\rho^2}{2(1-\rho)}, \quad (18)$$

де ρ - коефіцієнт навантаження, *q* - середня довжина черги.

Із порівняння виразів (17) та (18) витікає висновок, що із зростанням міри самоподібності пакетного трафіка (тобто, значень параметру *H*) вимоги до ємності буферу суттєво зростають у порівнянні з вимогами щодо класичних видів трафіка. Збільшення ємності буферу дозволяє зберігати «зайві» пакети у більш довгих чергах, але негативно впливає на показники реактивності інтерактивної системи.

Таким чином, з аналізу представлених вище моделей можливо зробити наступні висновки:

1) Характеристики потоків PDU телекомунікаційних систем, що транспортують запити і звернення клієнтів та відповіді на них (тобто, інтернет-трафік пакетованих даних), у т.ч. і характеристики розглянутої вище ON/OF-моделі пакетного трафіка, суттєво впливають на величину показників $\tau_{кп1}$, $\tau_{ко1}$, $\tau_{кп2}$ та $\tau_{ко2}$. Отже, справедливим є твердження, що реактивність системи у певній мірі залежить від характеристик прийнятих телекомунікаційних технологій фізичного, канального та мережного рівнів, що визначають швидкість

передавання даних через канали зв'язку, а також від швидкодії процесорів, що здійснюють обробку інформації в операційному середовищі прикладної системи.

2) Для підтримки заданих значень показників реактивності інтерактивної системи адміністратори мережного обладнання на серверній стороні цієї системи мають обрати експериментальним шляхом значення величини ємності буферної пам'яті b та кількості каналів його обслуговування v. При цьому мати на увазі, що реактивність треба покращувати за рахунок збільшення кількості каналів обслуговування, а не ємності буфера.

3) Параметри обладнання проміжних вузлів Інтернет не можуть бути об'єктами для маніпулювання ні на етапі створення інтерактивної системи, ні на етапі її експлуатації. Вважається, що мережі Інтернет притаманна властивість часової прозорості, тобто здатності забезпечувати тривалість затримки, яка відповідає нормативній якості обслуговування. Для проектантів та експлуатантів ці параметри задаються умовами технічного завдання і регламентуються відповідними нормативними документами. Зокрема, усі найбільш вживані види потоків, що транспортуються каналами IP, згруповані за шістьома класами трафіка IP, яким відповідають шість класів обслуговування. Оскільки інтерактивні системи функціонують на основі встановлених IP-з'єднань (дейтаграмний режим роботи не використовується), то наведемо нормативні значення параметрів реактивності для нульового та першого класу обслуговування, тобто для режиму роботи із встановленням IP-з'єднань (табл.2).

Таблиця 2

Нормативні значення параметрів реактивності для нульового та першого класу обслуговування

Назва параметру	Позначення параметру	Клас 0	Клас 1
Верхня межа щодо затримки пакетів, мс	$IPTD_{max}$	100	400
Верхня межа щодо варіації затримки, мс	$IPDV_{max}$	50	50
Поріг ймовірності перевищення $IPTD_{max}$	$P_{max}(IPTD_{max})$	$1x10^{-2}$	$1x10^{-2}$

Таким чином, при розрахунках величини τ_{Σ} є доцільним обрати наступні значення параметрів $\tau_{кп1}$, $\tau_{ко1}$, $\tau_{кп2}$ та $\tau_{ко2}$:

$$\tau_{кп1} = \tau_{ко1} = \tau_{кп2} = \tau_{ко2} = 100\ мс. \quad (19)$$

Отже, максимально можливий внесок компонентів мережного обладнання Інтернет у середнє значення проміжку часу реакції інтерактивної системи на запити клієнтів може складати не більше 400 мс. При цьому, потенційно можливе максимальне значення дисперсії поточних значень цих компонентів не може перевищувати 50 мс x 4 = 200 мс.

Із вище зазначеного витікає наступне твердження: щоб знизити величину показника реактивності τ_{Σ}, доцільно розробити метод побудови інтерактивної системи, що дозволяє оптимізувати характеристики, що впливають на швидкість обробки звернень клієнтів на прикладному рівні, зокрема оптимізувати характеристики схеми організації пошуку IP-адрес шуканих файлів з персональними даними клієнтів, а також узгодити характеристики потоків звернень та запитів з характеристиками інтерактивної системи. Іншими словами, оптимізацію здійснювати у напрямку мінімізації значень таких показників як $\tau_{чрг}$, $\tau_{пс}$, та $\tau_{соз}$. Як показують результати досліджень, саме ці об'єкти вносять основний вклад у величину реактивності інтерактивної прикладної системи. Тому завдання даного дослідження полягає у визначенні шляхів збільшення швидкості опрацювання запитів клієнтів до інтерактивних систем, та розробці відповідного метода, що базується на результатах такого визначення.

Синтез структурно-функціональної моделі інтерактивної системи як системи масового обслуговування. Оскільки у даній роботі реальний трафік запитів (звернень) представляється у вигляді суперпозиції передбачуваного випадкового процесу із відомою щільністю ймовірності та непередбачуваного процесу, що моделює різкі сплески трафіку, то мінімізацію показника реактивності інтерактивної системи доцільно здійснювати шляхом розробки відповідного методу опрацювання саме передбачуваної складової трафіка запитів. Підкреслимо, що у даному випадку мова йде не про потоки пакетів з клієнтськими даними, а про потоки фрагментів інформації, що відтворюють зміст звернень та запитів, на віртуальних портах прикладних програм, що призначені для опрацювання цих звернень та запитів. Потім після визначення оптимальних значень параметрів інтерактивної системи відносно передбачуваної складової трафіка звернень у даній роботі пропонується нейтралізувати негативний вплив непередбачуваних різких сплесків цього трафіка на сталість оптимізованої системи шляхом побудови відповідної динамічної адаптивної системи реагування на ці впливи. Зрозуміло, що адаптивне керування параметрами інтерактивної системи не підвищить її реактивність, але забезпечить стабільність її функціонування в умовах різких сплесків трафіка звернень.

Структурно-функціональна схема прийнятої моделі інтерактивній системи типу «запит/відповідь» показана на рис. 1.

Результати аналізу характеристик цієї моделі вказують на доцільність її представлення як системи масового обслуговування (СМО), що характеризується потоком заявок клієнтів на обслуговування з очікуваннями (з так званими неявними втратами, коли припускається потенціальна можливість перевищення максимального значення довжини черги у доступі до обслуговуючого пристрою) шляхом використання ресурсів наявного серверного обладнання - ресурсів лінійки пошукових серверів *ПС* та ресурсів визначеної множини серверів *СОЗ*, які опрацьовують запити клієнтів.

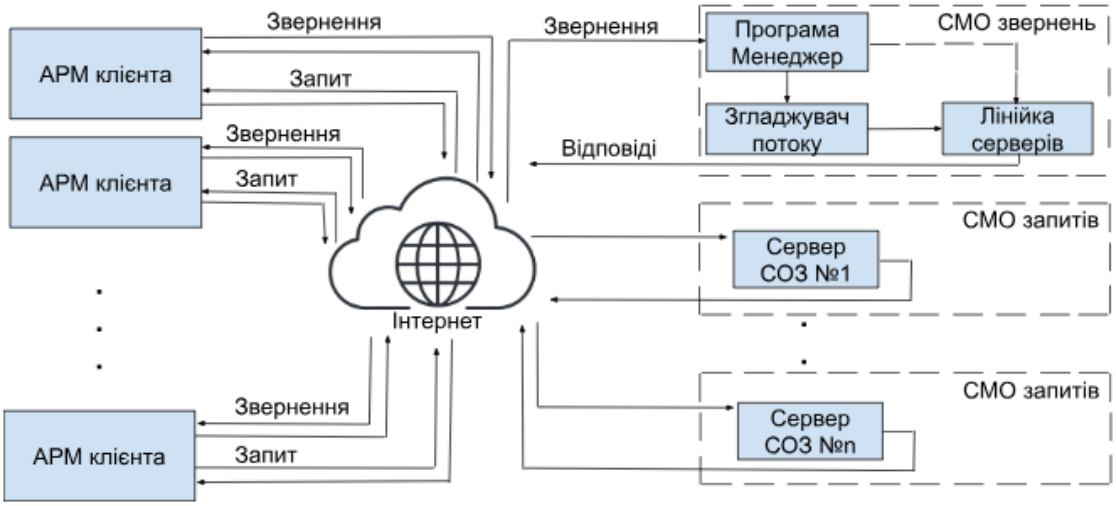

Рис. 1. Структурно-функціональна модель інтерактивної системи як СМО

Визначимо клас СМО, що має моделювати процес обробки звернень в інтерактивній Інтернет-системі для умов, коли в якості основних показників якості такої обробки розглядаються показники тривалості опрацювання звернень.

З урахуванням даних табл. 1 модель інтерактивної системи щодо обробки звернень на пошук IP-адрес необхідних клієнтам серверів *СОЗ* доцільно представити у наступному вигляді:

$$M / GI / v / r \le k / f_0^0 . \quad (20)$$

Запис $M / GI / v / r \le k / f_0^0$ означає повно доступну *v*-канальну СМО звернень клієнтів на пошук IP-адрес серверів *СОЗ* з файлами, що містять їхні персональні дані, і мають опрацьовувати їхні запити.

Прийнято наступні позначення: *v* – кількість *ПС* у лінійці пошукових серверів, яка обслуговує потік звернень; *M* - експоненціальний розподіл потоку звернень на увідному логічному порту програми Менеджер; *GI* - довільний закон розподілу часу обслуговування звернень пошуковими серверами щодо незалежних інтервалів між зверненнями; *r* – поточна кількість звернень у черзі; *k* – ємність буферної пам'яті на вході лінійки пошукових серверів; f_0^0 - обслуговування без пріоритетів.

Експоненціальний закон розподілу, що моделює потік звернень на ввідному логічному порту програми Менеджер, має ймовірнісні характеристики, що наведені у табл. 3. Параметр λ - інтенсивність потоку звернень. Параметр τ – тривалість акту обслуговування одного звернення.

Таблиця 3

Ймовірнісні характеристики

Функція розподілу	Щільність розподілу потоку звернень	Математичне очікування	Дисперсія потоку звернень
$1-e^{-\lambda t}$	$\lambda e^{-\lambda t}$	$1/\lambda$	$1/\lambda^2$

Зроблено припущення, що закон розподілу часу обслуговування звернень пошуковими серверами є довільним із заданими значеннями $b_\tau^{(1)}$- математичного очікування (середнього значення) часу обслуговування та $b_\tau^{(2)}$- другого початкового моменту функції розподілу τ, тобто дисперсії часу обслуговування.

Для СМО класу (20) якість обслуговування потоку звернень оцінюється за такими показниками:

– імовірність очікування $P\ (t_0 > 0)$, де t_0 – тривалість очікування;

– імовірність очікування більше припустимого часу $P\ (t_0 > t_{npn})$, яка пов'язана з оцінкою умовних втрат;

– середня тривалість очікування $\overline{t_0}$;

– середня довжина черги *q*;

– середня тривалість обслуговування звернення τ_0;

– ймовірність перевищення заданої довжини черги.

Реактивність даної моделі СМО визначається значеннями показників τ_0 та $\overline{t_0}$. Величина τ_0 залежить від характеристик СУБД, де зберігаються персональні дані клієнтів прикладної системи щодо IP-адрес серверів СОЗ, які їх обслуговують. Зрозуміло, що у рамках даного методу опрацювання трафіка показник τ_0 розглядається як задана величина, що має бути мінімізована відповідним вибором характеристик СУБД, зокрема вибором відповідного алгоритму пошуку IP-адрес серверів СОЗ.

З урахуванням вище зазначеного середня тривалість обробки одного звернення $\overline{t_e}$ у СМО класу (20) представляється наступним виразом:

$$\overline{t_e} \approx \overline{t_0} \cdot \frac{b_\tau^{(2)}}{2[b_\tau^{(1)}]}. \quad (21)$$

А середня довжина черги звернень на обслуговування визначиться як:

$$q = \overline{t_e} \cdot \lambda. \quad (22)$$

Синтез методу формування трафіка звернень, придатного для обслуговування прикладними засобами інтерактивних систем.

Загальні міркування щодо порядку формування трафіка звернень. Прикладне програмне забезпечення (ППЗ) інтерактивних систем реалізує моделі обслуговування трафіка звернень клієнтів, яке з точки зору забезпечення заданих показників реактивності цих систем має бути оптимізоване щодо характеристик моделей трафіка. Отже, ефективна робота засобів ППЗ є можливою лише тоді, коли характеристики трафіка звернень, що має опрацьовуватися цим ППЗ, будуть відповідати розглянутим моделям. Як показують результати експериментальних досліджень, реальний потік звернень клієнтів на вході лінійки пошукових серверів, які здійснюють процес опрацювання звернень на серверній стороні інтерактивної системи, не є стаціонарним. Та й розмах сплесків цього трафіка не є обмеженим. Оптимізація процесу управління таким трафіком за будь-якими критеріями є складним науково-технічним завданням, оскільки його стохастичні характеристики не піддаються прогнозуванню. Тому пошук шляхів перетворення нестаціонарних потоків у послідовності відрізків квазістаціонарного процесу з обмеженою дисперсією представляється актуальним завданням. Можливість такого перетворення витікає із очевидного припущення, що процедура згладжування пульсуючого потоку звернень шляхом усереднення на обраних певним чином часових інтервалах зменшить амплітуду та тривалість пульсацій і, тим самим, створить умови для формування трафіка, котрий за своїми характеристиками буде наближений до характеристик розглянутих вище моделей. Проте слід враховувати, що здійснення процедури згладжування потребує часу, тривалість якого негативно впливає на показники реактивності інтерактивної системи.

В даній роботі запропоновано метод перетворення реального потоку звернень в потоки з характеристиками, що наближені до характеристик розглянутих вище моделей. В основі цього метода лежить уявлення про реальний трафік звернень як дискретний квазістаціонарний випадковий процес із довільним видом щільності ймовірності та заданими величинами двох перших статистичних моментів, на який накладена складова у вигляді різких випадкових сплесків трафіка, статистичні характеристики котрих неможливо визначити та передбачити. Якщо таке уявлення відповідає дійсності, то є доцільним реальний потік звернень, що надходить на віртуальні порти пошукових серверів *ПС*, розділити на дві складові так, щоб одну складову потоку звернень, що просувається через буфер Згладжувача (рис. 1), можливо було моделювати у вигляді квазістаціонарних відрізків випадкового передбачуваного ON/OFF процесу із обмеженою дисперсією. А іншу складову потоку звернень розглядати як реалізації різких непередбачуваних сплесків трафіка. Оскільки середня інтенсивність потоку звернень, що утворюють сплески, набагато менша за середню інтенсивність основного потоку, то у результаті такого розділення виникає можливість оптимізувати параметри системи опрацювання звернень щодо показників її реактивності, а сплески розглядати як чинник, що негативно впливає на сталість оптимізованої системи. От же, формування першої складової трафіка має здійснюватися з урахуванням необхідності забезпечення заданих показників реактивності системи, а другої складової – з урахуванням необхідності мінімізації показника втрат звернень шляхом забезпечення сталості інтерактивної системи у заданому діапазоні значень коефіцієнту навантаження на серверне обладнання цієї системи.

Структурно-функціональна схема механізму формування увідного трафіка звернень. Згідно методу, що пропонується, механізм формування увідного трафіка звернень має виконувати функцію розділення потоку звернень на дві складові – передбачувану у вигляді послідовності відрізків квазістаціонарного процесу з обмеженою дисперсією та непередбачувану, що відтворює сплески цього трафіка, статистичні характеристики котрих не піддаються прогнозуванню. Структурно-функціональна схема реалізації такого механізму показана на рис. 2.

Як бачимо (рис. 2), реальний потік звернень через перемикач потоків *П1* просувається у буферну пам'ять стекового типу, що складається із двох рівнозначних частин – Буфер черги №1 (*БЧ1*) та Буфер черги №2 (*БЧ2*). Ці буфери призначені для утворення черг звернень. Звернення із увідного потоку послідовно у реальному часі потрапляють у черги через перемикач *П1*. Перемикач *П1* розбиває потік звернень на фрагменти однакової тривалості і здійснює їхнє передавання послідовно у часі по черзі то в один буфер, то в інший. Довжина фрагментів визначається, виходячи із заданих норм на показники реактивності інтерактивної системи. Отже, звернення, що входять до складу цих фрагментів, послідовно просуваються у чергах своїх буферів, заповнюючи ємність цих буферів на протязі проміжку часу між моментами переключення перемикача *П1*.

Виходи буферів також почергово підключаються до лінійки пошукових серверів *ПС* шляхом переключення перемикача *П2*. Перемикач *П2* працює синхронно з перемикачем *П1*, але у протифазі. Якщо, наприклад, перемикач *П1* підключає вхідний потік до *БЧ1*, то в цей момент перемикач *П2* відключає вихід цього буферу від лінійки *ПС* і підключає до цієї лінійки вихід *БЧ2*. Так що, на будь-якому поточному проміжку часу, що називається інтервалом усереднення, один із буферів заповнюється поточним фрагментом трафіка звернень, а з іншого буферу утворений на попередньому часовому інтервалі фрагмент трафіка подається на лінійку пошукових серверів *ПС*. На наступному інтервалі усереднення на лінійку *ПС* подається фрагмент вже з іншого буферу. Так що, коли один із буферів працює по входу (тобто, приймає у реальному часі та накопичує у черзі звернення із увідного потоку, утворюючи черговий фрагмент трафіку), то інший буфер працює по виходу (тобто, виштовхує у реальному часі накопичений фрагмент звернень через усереднювач швидкості потоку *УШП* на лінійку *ПС*).

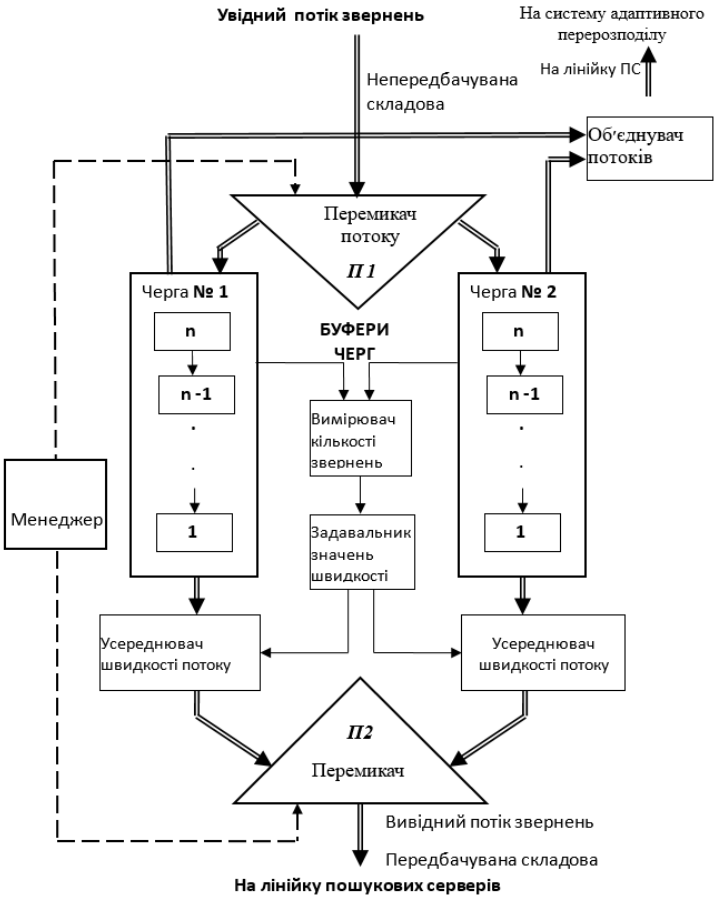

Рис. 2. Структурно-функціональна схема механізму формування увідного трафіка звернень

Звернемо увагу на те, що на уводи обох буферів просуваються фрагменти непрогнозованого нестаціонарного потоку, а на їхніх виходах утворюється послідовність фрагментів квазістаціонарного потоку, що представляється як випадковий передбачуваний ON/OFF процес із обмеженою дисперсією. От же, розглянута буферна конструкція виконує функцію перетворення фрагментів нестаціонарного процесу у фрагменти квазістаціонарного процесу. Для такого перетворення у склад структурно-функціональної схеми, що показана на рис.1, уведено три елементи: 1) *ВКЗ* - вимірювач кількості звернень, що накопичуються у кожній черзі на момент закінчення поточного часового інтервалу усереднення (тобто, кількість звернень, що утворюють один фрагмент потоку звернень); 2) *ЗЗШ* - задавальник значень швидкості усередненого потоку, тобто генератор сигналів, частота появи котрих змінюється в залежності від показників *ВКЗ*: 3) *УШП* - усереднювач швидкості потоку (по одному на виході кожної черги), що пропускає черговий фрагмент потоку звернень із швидкістю, що дорівнює швидкості усередненого потоку на поточному часовому інтервалі усереднення.

Перетворення нестаціонарного потоку на квазістаціонарні фрагменти здійснюється наступним чином. *ВКЗ* фіксує кількість звернень, що накопичені у буфері на момент закінчення чергового інтервала усереднення. *ЗЗШ* генерує потік еталонних сигналів із швидкістю, що визначається як результат ділення зафіксованої кількості накопичених у буфері звернень на тривалість інтервалу усереднення. *УШП* виштовхує із буферу накопичені у черзі звернення із швидкістю, що дорівнює усередненому значенню інтенсивності передбачуваної складової той ділянки увідного потоку звернень, що були накопичені у буфері на попередньомиу інтервалі усереднення. І все це працює у реальному часі.

Визначення параметрів буферу згладжувача. Визначимо параметри буферу згладжувача, за яких підпотік звернень, що формується у реальному часі на його вивідному порті, можливо було б моделювати ON/OFF процесом, якому притаманні наступні характеристики:

- інтервали активності τ_a представляються як послідовність статистично незалежних квазістаціонарних ділянок дискретного випадкового процесу;

- тривалість інтервалів активності τ_a обрана з урахуванням вимог щодо тривалості перебування звернень в черзі на обслуговування $\tau_{чрг}$, виходячи із необхідності виконання обмеження (2), а саме $\tau_{чрг} \le \tau^0_{чрг}$, де $\tau_{чрг}$ - тривалість затримки звернення у буфері Згладжувача; $\tau^0_{чрг}$ - максимально припустиме значення $\tau_{чрг}$;

- припустимий діапазон змін (розмах) середнього значення інтенсивності реалізацій випадкового процесу обрано з урахуванням вимог щодо припустимих значень коефіцієнту навантаження на лінійку пошукових серверів.

Потік звернень, що надходить з мережі, розподіляється між двома чергами, що утворюються в режимі почергового їхнього підключення/відключення до джерела/приймача трафіка під керуванням програми

Менеджер. Тобто створюються дві черги звернень, а реальний час розбивається на однакові інтервали активності, на котрих здійснюється усереднення трафіка. Режим переключення є наступним. Якщо на і-му інтервалі активності перший буфер підключається до увідного потоку звернень і відключається від лінійки пошукових серверів, то другий буфер у цей час відключається від потоку і підключається до лінійки пошукових серверів. На (і+1)-му інтервалі переключення здійснюються навпаки – перший буфер відключається від потоку і підключається до пошукових серверів, а другий буфер відключається від пошукових серверів і підключається до потоку. Сумарний потік з виходів буферних пристроїв можливо представити як дискретний ON/OFF процес послідовної передачі «пачок» звернень, що накопичуються у чергах, на увідні порти лінійки пошукових серверів ПС.

Кожен черговий «пакунок» звернень витісняється із буферної пам'яті, де була створена черга, на протязі одного поточного інтервалу активності ON/OFF процесу до тих пір, поки останнє звернення у черзі не покине цей буфер. В цей момент буфер звільниться від черги. Далі починається черговий інтервал, на протязі якого здійснюється процес накопичення у даному буфері чергового «пакунку» звернень, який буде переданий на опрацювання засобами пошукових серверів на наступному інтервалі активності процесу.

Інтервали активності τ_a є однаковими і співпадають із інтервалом усереднення швидкості у фрагменті потоку $\tau_{ус}$. У той час як тривалість OFF-складової процесу дорівнює тривалості переключення потоку з однієї черги на іншу. Так що, впливом цієї складової на реактивність інтерактивної системи можливо знехтувати.

Щодо вибору довжини (тривалості) інтервалу активності ON/OFF процесу τ_a слід зазначити наступне. На поточному інтервалі активності передається «пакунок» звернень, що був сформований як черга у буфері на протязі попереднього інтервалу активності. Оскільки затримка у буфері згідно умові (2) повинна не перевищувати $\tau^0_{чрг}$, то логічно обрати значення тривалості інтервалу активності як $\tau^0_{чрг}$. Отже, маємо

$$\tau_a = \tau_{ус} = \tau^0_{чрг}. \quad (23)$$

Задаємося значенням величини максимально можливої інтенсивності тієї складової потоку звернень, що має просуватися через буфер, як I^0_{max}. Зрозуміло, що поточне значення інтенсивності потоку на вході у буфер $I^0_{вх}$ внаслідок можливих сплесків може суттєво перевищувати I^0_{max}.

Якщо умова (23) виконується, то максимальна довжина черги, що поміщається у першу або другу буферну пам'ять, визначиться як

$$k^0_{max} = I^0_{max} \cdot \tau^0_{чрг}, \quad (24)$$

де k^0_{max} - максимально можлива кількість звернень, що може поміститися в одну із двох черг.

Тоді максимально можливе середнє значення інтенсивності квазістаціонарного підпотока звернень на інтервалі τ_a на виході буфера визначиться як

$$I^0_{maxсер} = k^0_{max} / \tau^0_{чрг}. \quad (25)$$

Зрозуміло, що можливий діапазон змін середньої інтенсивності відрізків квазістаціонарного трафіка на виході буферу $I^0_{сер}$ буде знаходитись у наступних межах:

$$0 \le I^0_{сер} \le I^0_{maxсер}. \quad (26)$$

Якщо ж інтенсивність увідного потоку в якийсь момент підвищиться так, що буфер почне переповнюватися «зайвими» зверненнями, то ці «зайві» звернення мають напряму, обминаючи буфер, потрапляти на порти лінійки пошукових серверів через підсистему вирівнювання поточних значень коефіцієнтів завантаження пошукових серверів згідно розподілу, що задається програмою Менеджер.

Висновки

У роботі визначено основні показники реактивності інтерактивних Інтернет-систем — середній час реакції та дисперсію затримок, а також проаналізовано чинники, що їх формують. Показано, що мережеве обладнання робить обмежений внесок у загальну затримку, тоді як вирішальний вплив мають параметри прикладного рівня. Це узгоджується з сучасними дослідженнями, де основна увага приділяється оптимізації алгоритмів обробки запитів, балансуванню навантаження та структурі пошукових процесів.

Встановлено, що наявні моделі трафіка відображають його загальні властивості, однак не завжди адекватно описують непередбачувані сплески, що істотно впливають на реактивність. Для подолання цієї обмеженості запропоновано розглядати реальний трафік як фрактальний дискретний квазістаціонарний процес із передбачуваними характеристиками та додатковою складовою випадкових збурень. Такий підхід дозволяє розглядати інтерактивну систему як нелінійну динамічну, що потребує адаптивного керування параметрами в реальному часі для забезпечення стійкості.

На цій основі розроблено модель пакетного Інтернет-трафіка та метод його обслуговування засобами телекомунікаційних протоколів. Показано, що для практичного застосування доцільним є поєднання аналітичних моделей із нормативними критеріями реактивності, які використовуються в Україні.

Синтезовано структурно-функціональну модель інтерактивної системи як системи масового обслуговування, яка враховує поділ реального трафіка на передбачувану та непередбачувану складові. Це дозволяє здійснити параметричну оптимізацію обслуговування квазістаціонарного потоку та зменшити негативний вплив сплесків шляхом балансування навантаження між серверами.

Запропоновано метод формування вхідного трафіка, що перетворює нестаціонарний потік у послідовність квазістаціонарних відрізків з обмеженою дисперсією. Відповідна структурно-функціональна схема та буферний згладжувач забезпечують підвищення стабільності та якості роботи прикладних Інтернет-систем в умовах нерівномірного навантаження.

Література

1. Iyengar, J., & Thomson, M. (2021). *QUIC: A UDP-based multiplexed and secure transport* (RFC 9000). Internet Engineering Task Force. https://doi.org/10.17487/RFC9000 Retrieved January 16, 2026, from https://www.rfc-editor.org/rfc/rfc9000

2. Bishop, M. (2022). *HTTP/3* (RFC 9114). Internet Engineering Task Force. https://doi.org/10.17487/RFC9114 Retrieved January 16, 2026, from https://www.rfc-editor.org/rfc/rfc9114

3. Oku, K., & Pardue, L. (2022). *Extensible prioritization scheme for HTTP* (RFC 9218). Internet Engineering Task Force. https://doi.org/10.17487/RFC9218 Retrieved January 16, 2026, from https://www.rfc-editor.org/rfc/rfc9218

4. Snyder, P., et al. (2020). QUIC: A large-scale deployment and performance study. In *Proceedings of the ACM Internet Measurement Conference (IMC)*. Retrieved January 16, 2026, from https://dl.acm.org/

5. Gupta, A., & Bartos, R. (2024). *Improving HTTP/3 quality of experience with incremental EPS* (arXiv:2403.04074). arXiv. Retrieved January 16, 2026, from https://arxiv.org/abs/2403.04074

6. W3C & WHATWG. (2025). *WebTransport over HTTP/3* (Editor's Draft / Working Draft). Retrieved January 16, 2026, from https://www.w3.org/TR/webtransport/

7. Herbots, J., et al. (2023). Vegvisir: A testing framework for HTTP/3 media streaming. In *Proceedings of MMSys 2023*. Retrieved January 16, 2026, from https://jherbots.info/public_media/research/mmsys2023_vegvisir_authorversion.pdf

8. Kong, L., Tan, J., Huang, J., Chen, G., Wang, S., Jin, X., Zeng, P., Khan, M., & Das, S. K. (2022). Edge-computing-driven Internet of Things: A survey. *ACM Computing Surveys, 55*(8), Article 174. https://doi.org/10.1145/3555308

9. Chen, C.-L., Brinton, C. G., & Aggarwal, V. (2021). Latency minimization for mobile edge computing networks. *IEEE Transactions on Mobile Computing*. https://doi.org/10.1109/TMC.2021.3117511

10. Singh, R., et al. (2023). A survey of mobility-aware multi-access edge computing. *Computer Networks*. https://doi.org/10.1016/j.comnet.2022.109341

11. Loutfi, S. I., et al. (2024). An overview of mobility awareness with MEC over 6G networks. *ICT Express*.

12. Doan, T. V., Nguyen, G. T., Reisslein, M., & Fitzek, F. H. P. (2021). FAST: Flexible and low-latency state transfer in mobile edge computing. *IEEE Access, 9*, 115315–115334. https://doi.org/10.1109/ACCESS.2021.3105583

13. Chrome Developers & Web.dev. (2024, January 31). *Interaction to Next Paint becomes a Core Web Vital on March 12, 2024*. Retrieved January 16, 2026, from https://web.dev/blog/inp-cwv-march-12

14. Chrome Developers & Web.dev. (n.d.). *Interaction to Next Paint (INP)*. Retrieved January 16, 2026, from https://web.dev/articles/inp

15. Google Search Central. (2023, May 10). *Introducing INP to Core Web Vitals*. Retrieved January 16, 2026, from https://developers.google.com/search/blog/2023/05/introducing-inp

16. Chrome Developers & Web.dev. (2024, March 12). *INP is now a Core Web Vital (FID deprecated)*. Retrieved January 16, 2026, from https://web.dev/blog/inp-cwv-launch

17. Web.dev. (2023, November 14). *Optimize resource loading with the Fetch Priority API (fetchpriority)*. Retrieved January 16, 2026, from https://web.dev/articles/fetch-priority

18. WHATWG. (2025, August 8). *HTML Standard: Fetch priority*. Retrieved January 16, 2026, from https://html.spec.whatwg.org/

19. Osmani, A. (2022, August 14). *Use fetchpriority=high to load your LCP hero image sooner*. Retrieved January 16, 2026, from https://addyosmani.com/blog/fetch-priority/

20. Web.dev. (2020, August 5). *content-visibility: A CSS property to boost rendering performance*. Retrieved January 16, 2026, from https://web.dev/articles/content-visibility

21. MDN Web Docs. (2025, August 8). *content-visibility — CSS*. Retrieved January 16, 2026, from https://developer.mozilla.org/en-US/docs/Web/CSS/content-visibility

22. Gao, T., Ge, Y., Wu, G., & Ni, J. (2010). A reactivity-based framework of automated performance testing for web applications. In *Proceedings of the International Conference on the Digital Content, Multimedia Technology and Its Applications (IDC)* (pp. 127–132). IEEE. https://doi.org/10.1109/DCABES.2010.127

23. Dias, J. P., Faria, J. P., & Ferreira, H. S. (2018). A reactive and model-based approach for developing Internet-of-Things systems. In *Proceedings of the 11th International Conference on the Quality of Information and Communications Technology (QUATIC)* (pp. 276–281). IEEE. https://doi.org/10.1109/QUATIC.2018.00049

24. Удосконалений метод цільового оновлення інтерфейсу користувача для підвищення ефективності веб-застосунків на основі реактивних потоків та Virtual DOM. (2025, July 19). *ResearchGate*. Retrieved January 16, 2026, from https://www.researchgate.net/publication/393789345

25. Chęć, D., & Nowak, Z. (2019). The performance analysis of web applications based on Virtual DOM and reactive user interfaces. In *Advances in Intelligent Systems and Computing* (pp. 119–134). Springer. https://doi.org/10.1007/978-3-319-99617-2_8

References

1. Iyengar, J., & Thomson, M. (2021). *QUIC: A UDP-based multiplexed and secure transport* (RFC 9000). Internet Engineering Task Force. https://doi.org/10.17487/RFC9000 Retrieved January 16, 2026, from https://www.rfc-editor.org/rfc/rfc9000
2. Bishop, M. (2022). *HTTP/3* (RFC 9114). Internet Engineering Task Force. https://doi.org/10.17487/RFC9114 Retrieved January 16, 2026, from https://www.rfc-editor.org/rfc/rfc9114
3. Oku, K., & Pardue, L. (2022). *Extensible prioritization scheme for HTTP* (RFC 9218). Internet Engineering Task Force. https://doi.org/10.17487/RFC9218 Retrieved January 16, 2026, from https://www.rfc-editor.org/rfc/rfc9218
4. Snyder, P., et al. (2020). QUIC: A large-scale deployment and performance study. In *Proceedings of the ACM Internet Measurement Conference (IMC)*. Retrieved January 16, 2026, from https://dl.acm.org/
5. Gupta, A., & Bartos, R. (2024). *Improving HTTP/3 quality of experience with incremental EPS* (arXiv:2403.04074). arXiv. Retrieved January 16, 2026, from https://arxiv.org/abs/2403.04074
6. W3C & WHATWG. (2025). *WebTransport over HTTP/3* (Editor's Draft / Working Draft). Retrieved January 16, 2026, from https://www.w3.org/TR/webtransport/
7. Herbots, J., et al. (2023). Vegvisir: A testing framework for HTTP/3 media streaming. In *Proceedings of MMSys 2023*. Retrieved January 16, 2026, from https://jherbots.info/public_media/research/mmsys2023_vegvisir_authorversion.pdf
8. Kong, L., Tan, J., Huang, J., Chen, G., Wang, S., Jin, X., Zeng, P., Khan, M., & Das, S. K. (2022). Edge-computing-driven Internet of Things: A survey. *ACM Computing Surveys, 55*(8), Article 174. https://doi.org/10.1145/3555308
9. Chen, C.-L., Brinton, C. G., & Aggarwal, V. (2021). Latency minimization for mobile edge computing networks. *IEEE Transactions on Mobile Computing*. https://doi.org/10.1109/TMC.2021.3117511
10. Singh, R., et al. (2023). A survey of mobility-aware multi-access edge computing. *Computer Networks*. https://doi.org/10.1016/j.comnet.2022.109341
11. Loutfi, S. I., et al. (2024). An overview of mobility awareness with MEC over 6G networks. *ICT Express*.
12. Doan, T. V., Nguyen, G. T., Reisslein, M., & Fitzek, F. H. P. (2021). FAST: Flexible and low-latency state transfer in mobile edge computing. *IEEE Access, 9*, 115315–115334. https://doi.org/10.1109/ACCESS.2021.3105583
13. Chrome Developers & Web.dev. (2024, January 31). *Interaction to Next Paint becomes a Core Web Vital on March 12, 2024*. Retrieved January 16, 2026, from https://web.dev/blog/inp-cwv-march-12
14. Chrome Developers & Web.dev. (n.d.). *Interaction to Next Paint (INP)*. Retrieved January 16, 2026, from https://web.dev/articles/inp
15. Google Search Central. (2023, May 10). *Introducing INP to Core Web Vitals*. Retrieved January 16, 2026, from https://developers.google.com/search/blog/2023/05/introducing-inp
16. Chrome Developers & Web.dev. (2024, March 12). *INP is now a Core Web Vital (FID deprecated)*. Retrieved January 16, 2026, from https://web.dev/blog/inp-cwv-launch
17. Web.dev. (2023, November 14). *Optimize resource loading with the Fetch Priority API (fetchpriority)*. Retrieved January 16, 2026, from https://web.dev/articles/fetch-priority
18. WHATWG. (2025, August 8). *HTML Standard: Fetch priority*. Retrieved January 16, 2026, from https://html.spec.whatwg.org/
19. Osmani, A. (2022, August 14). *Use fetchpriority=high to load your LCP hero image sooner*. Retrieved January 16, 2026, from https://addyosmani.com/blog/fetch-priority/
20. Web.dev. (2020, August 5). *content-visibility: A CSS property to boost rendering performance*. Retrieved January 16, 2026, from https://web.dev/articles/content-visibility
21. MDN Web Docs. (2025, August 8). *content-visibility — CSS*. Retrieved January 16, 2026, from https://developer.mozilla.org/en-US/docs/Web/CSS/content-visibility
22. Gao, T., Ge, Y., Wu, G., & Ni, J. (2010). A reactivity-based framework of automated performance testing for web applications. In *Proceedings of the International Conference on the Digital Content, Multimedia Technology and Its Applications (IDC)* (pp. 127–132). IEEE. https://doi.org/10.1109/DCABES.2010.127
23. Dias, J. P., Faria, J. P., & Ferreira, H. S. (2018). A reactive and model-based approach for developing Internet-of-Things systems. In *Proceedings of the 11th International Conference on the Quality of Information and Communications Technology (QUATIC)* (pp. 276–281). IEEE. https://doi.org/10.1109/QUATIC.2018.00049
24. Удосконалений метод цільового оновлення інтерфейсу користувача для підвищення ефективності веб-застосунків на основі реактивних потоків та Virtual DOM. (2025, July 19). *ResearchGate*. Retrieved January 16, 2026, from https://www.researchgate.net/publication/393789345
25. Chęć, D., & Nowak, Z. (2019). The performance analysis of web applications based on Virtual DOM and reactive user interfaces. In *Advances in Intelligent Systems and Computing* (pp. 119–134). Springer. https://doi.org/10.1007/978-3-319-99617-2_8