

УДК 621.397:004.032.6

**SPAİK: МІКРОСЕРВІСНА ПЛАТФОРМА СТРИМІНГУ
РЕАЛЬНОГО ЧАСУ З ВЛАСНИМИ МЕДІАСЕРВІСАМИ**

Гашимов І.А.

e-mail: ilkin.hashymov@nure.ua

Науковий керівник – к.т.н, проф. каф. МІРЕС Зубков О.В.

Харківський національний університет радіоелектроніки, каф. МІРЕС
м. Харків, Україна

This paper presents the concept of SPAİK (Streaming Platform for Adaptive Interactive Communications), a low-latency streaming platform evolved from a course project that originally relied on third-party media components. The diploma-stage goal is to replace external streaming infrastructure with in-house microservices responsible for signaling, real-time media routing, recording, and adaptive delivery. We propose a hybrid media pipeline: WebRTC for interactive «glass-to-glass» communication and low-latency HTTP-based streaming (LL-HLS / MPEG-DASH) for scalable audience distribution. A key research focus is minimizing end-to-end latency under volatile networks while preserving perceptual quality. We analyze ABR approaches (buffer-based BOLA vs. learning-based policies) and outline an ML-assisted bandwidth prediction module to reduce rebuffering and quality oscillations. The expected outcome is an experimentally validated architecture and a reproducible evaluation framework for latency and QoE trade-offs.

Онлайн-стрімінг та інтерактивні відеокommunікації дедалі частіше потребують не лише високої якості відео, а й мінімальної затримки «glass-to-glass», адже саме затримка визначає можливість природної взаємодії (відеочат, спільні події, інтерактивні донати/голосування). У курсовій реалізації платформи для стрімінгу було закладено сучасну серверну архітектуру та інструменти розгортання, але медіаплощина (media plane) спиралася на зовнішні рішення; у дипломному проєкті планується еволюція до мікросервісної архітектури та заміна сторонніх компонентів власними сервісами.

Назва «Спайк» пропонується як абревіатура SPAİK: S – Streaming (стрімінгова), P – Platform (платформа), A – Adaptive (адаптивних), I – Interactive (інтерактивних), K – Communications (комунікацій).

Розглядаються дві взаємопов'язані проблеми.

Перша проблема – мінімізація end-to-end latency за умов збереження якості. Для систем реального часу домінують складові затримки захоплення/кодування, транспорту, джитеру, декодування та відображення.

Друга проблема – адаптивне відеодоставлення в мережах із нестабільною пропускнуою здатністю, де основна мета – мінімізувати буферизацію, не руйнуючи якість і не провокуючи часті стрибки бітрейту. Стандартні підходи до HTTP-адаптивного стрімінгу полягають у сегментуванні та виборі представлення (representation) на клієнті; це лежить в основі MPEG-

DASH і суміжних профілів.

У MPEG-DASH/HLS логіка ABR (adaptive bitrate) зазвичай живе на клієнті. Для систем реального часу домінують складові затримки захоплення/кодування, транспорту, джитеру, декодування та відображення. У вигляді розрахункової моделі (для набору в Microsoft Equation) пропонується декомпозиція:

$$QoE = \sum_{t=1}^T q(r_t) - \lambda \sum_{t=1}^T Rebuf_t - \mu \sum_{t=2}^T |q(r_t) - q(r_{t-1})|$$

де:

- QoE – якість досвіду користувача,
- $q(r_t)$ – якість відео на момент часу t ,
- $Rebuf_t$ – кількість буферизацій на момент часу t ,
- λ та μ – вагові коефіцієнти для штрафів за буферизацію та зміни якості відео.

Як представник buffer-based підходів розглядається BOLA (Buffer Occupancy based Lyapunov Algorithm) – алгоритм, виведений через апарат ляпуновської оптимізації та прив'язаний до рівня буфера, що дозволяє пряму «штрафувати» ризик ребуферингу через стан буфера. Саме цей клас корисний для нестабільних мереж, де миттєві вимірювання throughput можуть бути шумними.

Як представник learning-based підходів розглядається Pensieve – система, що генерує політику ABR за допомогою підкріплювального навчання (reinforcement learning), оптимізуючи QoE (Quality of Experience) на основі спостережень плеєра. Важливо, що такі підходи націлені на автоматичне «вивчення» правил адаптації без жорстко зашитих евристик.

Для дипломної реалізації SPAIK пропонується модифікація ABR у вигляді двокомпонентної схеми:

- Базовий регулятор якості (buffer-based ядро за мотивами BOLA), який гарантує безпечний рівень буфера й обмежує агресивні рішення.

- Модуль прогнозування пропускну здатності (ML-передбачення $\hat{B}_{t+1..t+k}$, що не «замінює» регулятор, а коригує його параметри (наприклад, допустимий стрибок якості або цільовий рівень буфера) – ідеологічно близько до того, як Pensieve експлуатує стан середовища для вибору дій.

Ключова інженерна відмінність від «чистого» RL у тому, що в SPAIK ML не знімає з системи детерміновані гарантії (буфер як страховка), а працює як антишумовий шар для нестабільного каналу.

У дипломній версії SPAIK пропонується реалізувати чітке розділення control plane і media plane (без претензії на «єдино правильну» схему; нижче – проєктна пропозиція):

- API/Control plane (NestJS + GraphQL): акаунти, сесії, каталоги, права доступу, конфігурація трансляцій.

- Signaling service: встановлення WebRTC-сесій (SDP/ICE обмін) як окремий сервіс, ізоляція від бізнес-логіки.

- SFU/Media router (Golang): власний сервіс маршрутизації медіапотоків для багатокористувацьких кімнат (селективне пересилання), що замінює функціональність зовнішньої WebRTC-інфраструктури в частині real-time.

- Ingest/Transcode/Packager: pipeline, який приймає потоки з клієнтів (веб/мобільний/настільний застосунок замість залежності від стороннього енкодера), формує LL-HLS і MPEG-DASH представлення для глядачів. LL-HLS механізми часткових сегментів та preload hints є нормативною основою для низьколатентної HTTP-доставки.

- ABR-analytics service: збір телеметрії плеєра (час буфера, ребуферинг, патерни перемикань якості) та тренування/валідація моделі прогнозування пропускнуої здатності на цих даних (в рамках дозволеного стека – як окремий сервіс з інференсом, інтегрований у pipeline прийняття рішень).

Висновки. У межах тези обґрунтовано напрям дипломної еволюції SPAIK: поєднання WebRTC для інтерактивності з LL-HLS/MPEG-DASH для масштабованої доставки та дослідження ABR на перетині buffer-based (BOLA) і learning-based (Pensieve) підходів. Архітектурно пропонується мікросервісне винесення медіакомпонентів у власний бекенд, що зменшує залежність від зовнішньої медіаінфраструктури та створює керований експериментальний стенд для вимірювань L_{e2e} і QoE.

Список використаних джерел:

1. Roach A. Overview: Real-Time Protocols for Browser-Based Applications : RFC 8825. IETF, 2021. URL: <https://datatracker.ietf.org/doc/html/rfc8825> (дата звернення: 11.03.2026).
2. Moving Picture Experts Group. MPEG-DASH: Media Presentation Description and Segment Formats : вебсторінка. URL: <https://www.mpeg.org/standards/MPEG-DASH/1/> (дата звернення: 10.03.2026).
3. Spiteri K., Urgaonkar R., Sitaraman R. BOLA: Near-Optimal Bitrate Adaptation for Online Videos : arXiv:1601.06748. 2016. URL: <https://arxiv.org/pdf/1601.06748> (дата звернення: 10.03.2026).
4. Mao H., Netravali R., Alizadeh M. Neural Adaptive Video Streaming with Pensieve. Proc. ACM SIGCOMM 2017. URL: <https://people.csail.mit.edu/hongzi/content/publications/Pensieve-Sigcomm17.pdf> (дата звернення: 12.03.2026).