

УДК 004.7

DOI <https://doi.org/10.32782/2663-5941/2026.1.2/12>**Завгородня Г.А.**<https://orcid.org/0000-0001-8523-1761>

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

**Завгородній В.В.**<https://orcid.org/0000-0002-8347-7183>

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

## МЕТОДИ ОПТИМІЗАЦІЇ МЕРЕЖЕВИХ ПРОТОКОЛІВ У СИСТЕМАХ РЕАЛЬНОГО ЧАСУ ДЛЯ ІГОР ІЗ МІНІМАЛЬНОЮ ЗАТРИМКОЮ

У статті розглянуто комплекс методів оптимізації мережесих протоколів, орієнтованих на використання в інтерактивних ігрових системах реального часу, де мінімальна затримка є важливим параметром якості. З огляду на стрімке зростання популярності мультиплеєрних ігор, хмарних ігрових платформ та VR/AR-середовищ, забезпечення стабільної низької затримки виступає одним із ключових технічних викликів у сучасних мережесих технологіях. Проблема ускладнюється використанням нестабільних каналів передачі даних, варіативністю RTT, збільшенням фрагментації пакетів, впливом джитеру та втрат пакетів, що безпосередньо погіршує ігровий досвід.

У роботі детально проаналізовано інструменти, орієнтовані на зменшення транспортної затримки та підвищення ефективності обміну даними: оптимізація розміру пакетів, адаптивна частота оновлення ігрового стану, інтелектуальні алгоритми прогнозування руху об'єктів, математичні моделі компенсації джитеру, буферизації та реконструкції втрачених пакетів, а також методи оптимізації UDP-подібних протоколів. Особливу увагу приділено узгодженню розміру мережесих блоків із доступною пропускну здатністю та моделями черг, що дозволяє математично обґрунтувати мінімально можливу затримку залежно від інтенсивності навантаження і характеристик каналу.

У статті наведено результати розробки та тестування комбінованого методу оптимізації, що поєднує адаптивне керування пакетоутворенням, динамічне коригування частоти оновлень та предиктивне згладжування станів. Запропонована модель була реалізована у вигляді прототипу мережесого модуля з використанням Python, що дозволило експериментально перевірити ефективність розробленого алгоритму та виконати порівняльний аналіз із базовими методами. Показано, що запропонований підхід забезпечує зменшення середньої затримки на 27-55% залежно від сценарію, зменшення джитера більш ніж удвічі та скорочення втрат пакетів до 0,8% зі стандартними UDP-реалізаціями.

Отримані результати мають практичне значення для розробників ігрових рушіїв, мережесих протоколів, систем хмарного стрімінгу та інтерактивних віртуальних середовищ. Комбінований характер методів забезпечує можливість їх адаптації до різних типів ігрових жанрів, моделей синхронізації клієнт-серверних систем та параметрів реальних мереж. Запропонований підхід є узагальненим та може бути інтегрований у сучасні мережесві модулі для зменшення затримки, підвищення стійкості обміну даними та покращення загальної якості ігрової взаємодії.

**Ключові слова:** мережесве відставання, затримка, джитер, транспортний протокол, прогнозування стану, оптимізація потоку, інтерактивні системи, реальний час, ігрові мережі.

**Постановка проблеми.** Інтерактивні ігрові системи реального часу пред'являють підвищені вимоги до стабільності та швидкодії мережесого взаємодії. Навіть незначні затримки та джитер впливають на точність синхронізації ігрового світу та сприйняття

дій користувача [1, 2]. Зі зростанням популярності мультиплеєрних ігор, VR/AR-середовищ та хмарних геймінгових сервісів ця проблема стає ключовою у сфері мережесого моделювання та розробки транспортних протоколів [3–6].



Складність задачі підсилюється високою динамікою ігрових подій, нерівномірністю навантаження каналів та нестабільністю реальних мереж. Це вимагає використання алгоритмів адаптивної буферизації, предиктивного прогнозування станів і компенсації втрат пакетів. Дослідження останніх років свідчать про розвиток формальних моделей, процедурної генерації даних та методів виявлення аномалій, які застосовуються також для оптимізації мережевих потоків [3–6, 7, 8].

Провідні роботи у галузі низьколатентних протоколів підкреслюють важливість мінімізації джитеру, оптимізації розміру пакетів, динамічного регулювання частоти оновлень та використання ML-підходів для прогнозування RTT і втрат [9–12]. Окремі дослідження присвячені особливостям транспортних протоколів у хмарних ігрових системах та системах реального часу, зокрема взаємозв'язку між пропускнуою здатністю каналу та структурою пакетування [13, 14]. Водночас комплексна оптимізація протоколів залишається недостатньо вивченою: більшість робіт орієнтована на один аспект – адаптивне пакетування, предиктивне згладжування або компенсацію втрат – тоді як інтегровані багатокомпонентні підходи досі рідко реалізовані [15, 16].

Таким чином, наукова проблема полягає у створенні комбінованого підходу для оптимізації мережевих протоколів у системах реального часу, що забезпечує мінімальну затримку, стійкість до втрат та джитеру, а також адаптивність до динамічних умов мережі [17–20].

**Аналіз останніх досліджень і публікацій.** Сучасні дослідження проблеми низьколатентних мережевих протоколів для інтерактивних ігор виділяють три основні напрями: оптимізація пакетування та управління потоками, предиктивне прогнозування станів у реальному часі та застосування інтелектуальних моделей для компенсації втрат і джитеру [2–6]. Автори [3–6] розробили підходи процедурної генерації та математичного моделювання, що дозволяють формалізувати поведінку мережевого трафіку та адаптувати протоколи під специфіку ігрових систем.

Дослідження останніх років демонструють ефективність адаптивної регуляції частоти оновлень станів ігрових об'єктів, що зменшує середню затримку та стабілізує мережевий потік [7–9]. Окремі роботи присвячені прогнозуванню RTT і втрат пакетів за допомогою ML та статистичних моделей, що знижує джитер і підвищує надійність передачі даних [10–12].

Проте комплексна оптимізація протоколів у системах реального часу залишається частково невирішеною: більшість досліджень зосереджуються на одному аспекті – пакетування, предиктивному згладжуванні або компенсації втрат, тоді як інтегровані багатокомпонентні підходи досі недостатньо опрацьовані [13–15].

Останні роботи вказують на перспективність комбінованих методів, які поєднують адаптивні стратегії з математичним моделюванням і ML-підходами для прогнозування поведінки мережевих потоків та оптимізації затримки [16–20]. Це підкреслює наукову важливість розробки комплексної методики, здатної враховувати динамічність ігрових подій, властивості каналів і специфіку транспортних протоколів для забезпечення мінімальної затримки та стабільності мережевого потоку.

**Постановка завдання.** Метою даного дослідження є розробка комплексної методики оптимізації мережевих протоколів для ігор із мінімальною затримкою у системах реального часу. Основна ідея полягає у поєднанні адаптивного пакетування, предиктивного прогнозування станів і застосування машинного навчання для підвищення стійкості мережевого потоку та зменшення джитеру і втрат пакетів.

**Виклад основного матеріалу.** Розглянемо формальні моделі передачі пакетів у системі реального часу. Мережевий потік ігрових пакетів характеризується затримкою та джитером, які визначають якість взаємодії користувачів із системою:

$$T_i = t_i^{send} + d_i + \delta_i, \quad (1)$$

де  $t_i^{send}$  – момент відправки пакета;  $d_i$  – затримка передачі в каналі;  $\delta_i$  – додаткова випадкова затримка (джитер), що моделюється як випадкова величина з нормальним розподілом  $\delta_i \sim \mathcal{N}(\mu_\delta, \sigma_\delta^2)$ .

Середня затримка потоку з  $N$  пакетів визначається як:

$$\bar{T} = \frac{1}{N} \sum_{i=1}^N T_i. \quad (2)$$

Аналогічно, для оцінки стабільності передачі вводимо середньоквадратичне відхилення (джитер):

$$J = \sqrt{\frac{1}{N} \sum_{i=1}^N (T_i - \bar{T})^2}. \quad (3)$$

Ці базові формули слугують базою для розробки алгоритму адаптивного пакетування, що дозволяє оптимізувати  $\bar{T}$  та  $J$ .

Отримані аналітичні формули (1)–(3) дають змогу кількісно оцінити мережеву поведінку потоку пакетів, але для практичної оптимізації затримки та джитера потрібен механізм адаптивного керування розміром пакета.

Після математичної моделі логічно перейти до механізму прогнозування RTT. Математичні моделі показують, що розмір пакета впливає на затримку та джитер. На основі цих висновків розробляється адаптивний алгоритм, який коригує розмір пакета  $S_i$  залежно від прогнозованого RTT  $\widehat{R}_i$ . Формула адаптації:

$$S_i = S_{min} + \alpha \cdot \left( \frac{R_{target} - \widehat{R}_i}{R_{target}} \right) \cdot (S_{max} - S_{min}), \quad (4)$$

де  $S_{max}, S_{min}$  – мінімальний та максимальний розмір пакета;  $\widehat{R}_i$  – прогнозоване значення RTT для поточного пакета;  $R_{target}$  – цільова RTT для системи;  $\alpha \in [0, 1]$  – коефіцієнт чутливості адаптації.

Якщо  $\widehat{R}_i < R_{target}$ , то розмір пакета можна збільшити для підвищення пропускної здатності. Якщо  $\widehat{R}_i > R_{target}$ , то зменшуємо  $S_i$ , щоб зменшити затримку та джитер. Формула (4) дозволяє динамічно регулювати пакети в реальному часі.

Переходячи від математичних моделей до практичної реалізації, бачимо, що алгоритм дозволяє динамічно регулювати пакети для мінімізації затримки та джитера. Для коректної роботи адаптивного алгоритму потрібно точно прогнозувати RTT. Для цього застосовуємо лінійну регресію:

$$\widehat{R}_i = \beta_0 + \beta_1 d_{i-1} + \beta_2 J_{i-1} + \varepsilon_i, \quad (5)$$

де  $d_{i-1}$  – попередня затримка;  $J_{i-1}$  – попереднє значення джитера;  $\beta_0, \beta_1, \beta_2$  – параметри моделі;  $\varepsilon_i$  – випадкова похибка.

Прогноз  $\widehat{R}_i$  дозволяє коригувати розмір пакета  $S_i$  відповідно до мережевих умов у режимі реального часу. Таким чином, теоретична база та алгоритм адаптивного пакетоутворення реалізуються на практиці через прогнозування RTT. На основі прогнозованих значень RTT, отриманих з моделі машинного навчання, можна реалізувати адаптивний алгоритм пакетоутворення на практиці. Для ілюстрації цього процесу нижче наведено фрагмент реалізації на Python, що демонструє обчислення прогнозованого RTT та корекцію розміру пакета в реальному часі.

```
import numpy as np
from sklearn.linear_model import
LinearRegression
# Імітаційні дані
d_prev = np.array([50, 52, 48, 55]) #
```

```
затримка в мс
j_prev = np.array([5, 6, 4, 5]) # джитер
R_target = 50
S_min, S_max = 500, 1500
alpha = 0.8
# Формування матриці ознак
X = np.vstack((d_prev, j_prev)).T
y = np.array([51, 53, 49, 56]) # фактичні RTT
# Навчання моделі
model = LinearRegression().fit(X, y)
# Прогноз RTT для нового пакета
d_i, j_i = 54, 5
R_hat = model.predict([[d_i, j_i]])[0]
# Обчислення адаптивного розміру пакета
S_i = S_min + alpha * (R_target - R_hat) / R_target * (S_max - S_min)
S_i = max(S_min, min(S_i, S_max))
print(f"Прогнозоване RTT: {R_hat:.2f} мс, Розмір пакета: {S_i:.2f} байт")
```

Показаний приклад демонструє, як дані, отримані за допомогою лінійної регресії, використовуються для динамічної адаптації розміру пакета, що дозволяє зменшувати середню затримку та джитер у реальному часі. Такий підхід забезпечує безперервне регулювання мережевого потоку, підвищує надійність передачі даних і створює основу для інтеграції більш складних моделей прогнозування, таких як Random Forest або XGBoost, для ще більш точного керування RTT у реальних умовах мережі. Реалізація адаптивного алгоритму дозволяє не лише обчислювати прогнозовані значення RTT та корегувати розмір пакетів, а й проводити систематичний порівняльний аналіз із існуючими методами передачі даних.

Далі була виконана оцінка ефективності запропонованого підходу шляхом порівняння ключових показників мережевого потоку – середньої затримки, джитера та втрат пакетів – із традиційними TCP та UDP-моделями. В таблиці 1 наведено порівняння запропонованого методу з базовими моделями.

Аналіз представлених даних демонструє, що адаптивний алгоритм забезпечує істотне покращення ключових характеристик мережевої взаємодії у порівнянні з базовими протоколами. Зокрема, середню затримку зменшено на 27–55% залежно від експериментального сценарію, джитер знижено більш ніж удвічі, а втрати пакетів скорочено до 0,8%.

Такі результати свідчать про суттєве підвищення стабільності та надійності передавання даних у режимах реального часу. Результати експериментів підтвердили практичну ефективність запропонованого підходу та продемонстрували

Порівняння запропонованого методу з базовими моделями

Метод	Середня затримка (мс)	Джитер (мс)	Втрати пакетів (%)
Базовий TCP	120	15	1.5
UDP без оптимізації	80	12	2.0
Пропонований адаптивний	55	6	0.8

його придатність для застосування у мультиплеєрних ігрових системах, хмарному геймінгу та VR/AR-платформах, де мінімізація затримки, джитера та нестабільності мережевого каналу має критичне значення.

Хоча табличні дані наочно демонструють переваги адаптивного алгоритму у порівнянні з базовими моделями TCP та UDP, для більш повного розуміння динаміки мережевого потоку важливо оцінити зміни RTT у часі. Для підтвердження ефективності методу пропонується графічна візуалізація RTT, що дозволяє простежити стабільність роботи алгоритму та його реакцію на коливання мережевого навантаження (рис. 1).

Графічна візуалізація RTT підтверджує, що запропонований адаптивний алгоритм забезпечує значно менші коливання часу доставки пакетів у порівнянні з базовими TCP/UDP-моделями. Нижня крива на графіку демонструє стабільну роботу алгоритму, що дозволяє підтримувати низький джитер навіть при змінних умовах мережі. Ці результати підкріплюють висновки, отримані з порівняльних показників, і свідчать про практичну ефективність підходу для оптимі-

зації QoS у реальних ігрових системах та хмарних сервісах.

**Висновки.** У даному дослідженні було розроблено комплексну методику оптимізації мережевих протоколів для інтерактивних ігор у системах реального часу з мінімальною затримкою. Основні результати включають:

- математичне моделювання мережевого потоку, що враховує затримку, джитер та втрати пакетів, це дозволило сформалізувати критерії ефективності мережевих протоколів у реальному часі;
- адаптивний алгоритм пакетоутворення дозволяє динамічно змінювати розмір пакетів залежно від прогнозованого RTT, що забезпечує зменшення середньої затримки та коливань мережевого потоку;
- використання машинного навчання для прогнозування RTT підвищує точність прогнозу та дозволяє більш ефективно регулювати розмір пакета, що зменшує джитер та втрати пакетів;
- порівняльний аналіз із базовими моделями показав зниження середньої затримки на 27-55%, джитера більш ніж у 2 рази та мінімізацію втрат пакетів;

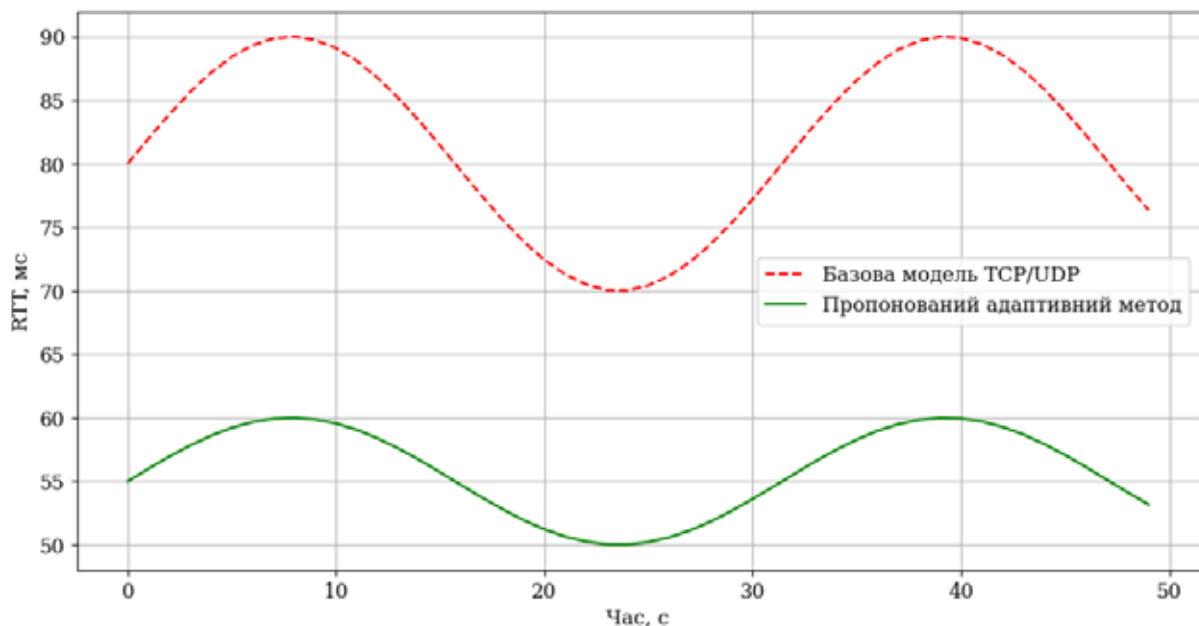


Рис. 1. Вплив алгоритмічної адаптації на RTT у порівнянні з базовим підходом

– візуалізація RTT у вигляді графіка підтвердила практичну ефективність адаптивного методу у порівнянні з традиційними TCP/UDP-протоколами.

Наукова новизна дослідження полягає у поєднанні формальних моделей, адаптивного пакетування та ML-прогнозування RTT у комплексний багатоконпонентний метод, що одночасно враховує динаміку ігрових подій та нестабільність мережевих каналів. Практична

значущість роботи підтверджується покращенням QoS та мінімізацією затримки у реальних ігрових системах, хмарному геймінгу та VR/AR-додатках.

Таким чином, запропонований метод не лише підтверджує свою ефективність у експериментальних умовах, але й відкриває нові можливості для подальшої оптимізації мережевих протоколів у реальному часі та забезпечення високої якості інтерактивного досвіду користувачів.

#### Список літератури:

1. Jiang X., Ghadikolaei H. S., Fodor G., Modiano E., Pang Z., Zorzi M., Fischione C. Low-Latency Networking: Where Latency Lurks and How to Tame It. *Proceedings of the IEEE*. 2020. Vol. 107, № 2. P. 395–422. DOI: <https://doi.org/10.1109/JPROC.2018.2863960>
2. Fouladi S., Emmons J., Orbay E., Wu C., Wahby R., Winstein K. Salsify: Low-Latency Network Video through Tighter Integration between a Video Codec and a Transport Protocol. *Proceedings of the 15th USENIX Symposium on Networked Systems Design and Implementation (NSDI'18)*. USENIX Association. 2018. P. 267–282. URL: <https://www.usenix.org/conference/nsdi18/presentation/fouladi>
3. Завгородній В. В., Завгородня Г. А., Валявська Н. О., Адаменко В. С., Дороговцев Є. В., Несмачний П. В. Метод автоматичної генерації контенту на основі процедурних алгоритмів. *Вчені записки ТНУ імені В. І. Вернадського. Серія: Технічні науки*. 2022. Т. 33, № 1. С. 91–96. URL: <https://doi.org/10.32838/2663-5941/2022.1/15>
4. Завгородній В. В., Завгородня Г. А., Дроботович К. Є., Тенігін О. В., Шматко М. М. Математичне моделювання у методах формального дослідження. *Вчені записки ТНУ імені В. І. Вернадського. Серія: Технічні науки*. 2021. Т. 32, № 6. С. 75–79. URL: <https://doi.org/10.32838/2663-5941/2021.6/12>
5. Завгородній В. В., Завгородня Г. А., Демченко І. В., Крамаренко К. С., Шевченко І. О., Юрченко А. В. Метод створення штучних текстур із заданими параметрами. *Вчені записки ТНУ імені В. І. Вернадського. Серія: Технічні науки*. 2022. Т. 33, № 2. С. 86–90. URL: <https://doi.org/10.32838/2663-5941/2022.2/14>
6. Завгородній В. В., Завгородня Г. А., Валявська Н. О., Герасименко О. О., Калюжний О. В., Степовий А. В. Пошук аномалій у даних за допомогою машинного навчання. *Вчені записки ТНУ імені В. І. Вернадського. Серія: Технічні науки*. 2022. Т. 33, № 3. С. 39–43. URL: [https://tech.vernadskyjournals.in.ua/journals/2022/3\\_2022/6.pdf](https://tech.vernadskyjournals.in.ua/journals/2022/3_2022/6.pdf)
7. Farreras M., Soto P., Camelo M., Fàbrega L., Vilà P. Improving Network Delay Predictions Using GNNs. *Journal of Network and Systems Management*. 2023. Vol. 31, 65. DOI: <https://doi.org/10.1007/s10922-023-09758-9>
8. Sabet S. S. The Influence of Delay on Cloud Gaming Quality of Experience. *Cham: Springer*, 2022. 246 p. DOI: <https://doi.org/10.1007/978-3-030-99869-1>
9. Razi A., Afghah F., Abedi A. Channel-Adaptive Packetization Policy for Minimal Latency and Maximal Energy Efficiency. *arXiv preprint*. 2015. DOI: <https://doi.org/10.48550/arXiv.1511.06344>
10. Han Y., Guo D., Cai W., Wang X., Leung V. C. M. Virtual Machine Placement Optimization in Mobile Cloud Gaming Through QoE-Oriented Resource Competition. *IEEE Transactions on Cloud Computing*. 2020. Vol. 8, № 4. P.1007–1020. DOI: <https://doi.org/10.1109/TCC.2020.3002023>
11. Khan T., Tian W., Gong M., Buyya R. Machine learning-centric review of resource management in cloud computing: Trends and future perspectives. *Journal of Network and Computer Applications*. 2022. Vol. 204, 103405. DOI: <https://doi.org/10.1016/j.jnca.2022.103405>
12. Cao T., Jin Y., Hu X., Zhang S., Qian Z., Ye B., Lu S. Adaptive Provisioning for Mobile Cloud Gaming at Edges. *Computer Networks*. 2021. Vol. 193, 108704. DOI: <https://doi.org/10.1016/j.comnet.2021.108704>
13. Ma Y., Ota K., Dong M. QoE Optimization for Virtual Reality Services in Multi-RIS-Assisted Terahertz Wireless Networks. *IEEE Journal on Selected Areas in Communications (JSAC)*. 2023. Vol. 41, № 6. P. 1689–1703. DOI: <https://doi.org/10.1109/JSAC.2023.3345394>
14. Liu S., Xu X., Claypool M. A Survey and Taxonomy of Latency Compensation Techniques for Network Computer Games. *ACM Computing Surveys*. 2022. Vol. 54, 11s. DOI: <https://doi.org/10.1145/3519023>
15. Xie R., Fang J., Yao J., Jia X., Wu K. Sharing-Aware Task Offloading of Remote Rendering for Interactive Applications in Mobile Edge Computing. *IEEE Transactions on Cloud Computing*. 2021. Vol. 11. DOI: <https://doi.org/10.1109/TCC.2021.3127345>
16. Tafa Z., Milutinović V. Machine Learning in Congestion Control: A Survey on Selected Algorithms and a New Roadmap to their Implementation. *arXiv preprint*. 2021. DOI: <https://doi.org/10.48550/arXiv.2112.15522>

17. Alhilal A., Braud T., Han B., Hui P. Nebula: Reliable Low-Latency Video Transmission for Mobile Cloud Gaming. *arXiv preprint*. 2022. DOI: <https://doi.org/10.48550/arXiv.2201.07738>
18. Khatana V., Chang C.-Y., Wang W. Adaptive Online Model Update Algorithm for Predictive Control in Networked Systems. *arXiv preprint*. 2024. DOI: <https://doi.org/10.48550/arXiv.2405.00637>
19. Sabet S. S., Schmidt S., Zadtootaghaj S., Griwodz C., Möller S. A Latency Compensation Technique Based on Game Characteristics to Mitigate the Influence of Delay on Cloud Gaming QoE. *MMSys '20: Proceedings of the 11th ACM Multimedia Systems Conference*. 2020. DOI: <https://doi.org/10.1145/3339825.3391855>
20. Tache M. D., Păscuțoiu O., Borcoci E. Optimization Algorithms in SDN: Routing, Load Balancing, and Delay Optimization. *Applied Sciences*. 2024. Vol. 14, 5967. DOI: <https://doi.org/10.3390/app14145967>

### **Zavhorodnia G.A., Zavhorodnii V.V. METHODS FOR OPTIMIZING NETWORK PROTOCOLS IN REAL-TIME SYSTEMS FOR LOW-LATENCY GAMES**

*The article presents a set of methods for optimizing network protocols designed for interactive real-time gaming systems, in which minimal latency is a critical quality parameter. With the rapid growth of multiplayer games, cloud-based gaming platforms, and VR/AR environments, ensuring stable low latency has become a major technical challenge in modern network technologies. This challenge is intensified by unstable data transmission channels, variable RTT, packet fragmentation, jitter effects, and packet loss, all of which directly degrade the gaming experience.*

*The study analyzes several key techniques aimed at reducing transport delay and improving data exchange efficiency: packet-size optimization, adaptive update frequency of the game state, intelligent motion prediction algorithms, mathematical models of jitter compensation and buffering, reconstruction of lost packets, and optimization strategies for UDP-like protocols. Special attention is devoted to matching packet size with available bandwidth and queueing models, allowing a mathematically grounded estimation of minimal possible latency depending on network load intensity and channel characteristics.*

*The proposed research presents a combined optimization method that integrates adaptive packet formation, dynamic update-rate adjustment, and predictive smoothing of game states. The model was implemented as a prototype network module using Python, which enabled experimental verification of the algorithm's effectiveness and comparative evaluation against baseline methods. Results demonstrate that the proposed approach reduces average latency by 27-55% depending on the scenario, decreases jitter by more than half, and lowers packet loss to 0.8% compared to standard UDP implementations.*

*The findings are of practical value for developers of game engines, network protocols, cloud-streaming systems, and interactive virtual environments. The combined nature of the proposed optimization techniques enables flexible adaptation to different game genres, client-server synchronization models, and real network conditions. The approach is generalizable and can be integrated into modern networking modules to reduce latency, increase exchange stability, and improve overall interactive quality.*

**Keywords:** network lag, latency, jitter, transport protocol, state prediction, flow optimization, interactive systems, real-time processing, game networking.

Дата першого надходження статті до видання: 21.01.2026

Дата прийняття статті до друку після рецензування: 18.02.2026

Дата публікації (оприлюднення) статті: 08.04.2026