НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Навчально-науковий інститут телекомунікаційних систем

Кафедра інформаційних технологій в телекомунікаціях

До захисту допущено:

Завідувач кафедри

_____ Марія СКУЛИШ

«___»_____2025 р.

Дипломна робота

на здобуття ступеня бакалавра

за освітньо-професійною програмою «Інформаційно-комунікаційні технології»

спеціальності 172 Телекомунікації та радіотехніка

на тему: «Аналіз впровадження протоколів CDN та MEC в мережах 5G, для забезпечення заданої якості обслуговування абонентів»

Виконав:
студент IV курсу, групи ТІ-12
Книр Глєб Анатолійович _____

Науковий керівник: доцент кафедри ІТТ НН ІТС, кандидат технічних наук, доцент Правило Валерій Володимирович _____

Рецензент: доцент кафедри ТК НН ІТС, кандидат технічних наук, доцент Явіся Валерій Сергійович _____

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.

Студент _____

Київ – 2025 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Навчально-науковий інститут телекомунікаційних систем
Кафедра інформаційних технологій в телекомунікаціях

Рівень вищої освіти – перший (бакалаврський)
Спеціальність – 172 Телекомунікації та радіотехніка
Освітньо-професійна програма «Інформаційно-комунікаційні технології»

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Марія СКУЛИШ
«___»_____2025 р.

ЗАВДАННЯ
на дипломну роботу студенту
Книр Глєбу Анатолійовичу

1. Тема роботи «Аналіз впровадження протоколів CDN та MEC в мережах 5G, для забезпечення заданої якості обслуговування абонентів», керівник роботи Правило Валерій Володимирович - кандидат технічних наук, доцент, затверджені наказом по університету від №1755-с від 26.05.2025р.

2. Термін подання студентом роботи 06.06.2025р..

3. Вихідні дані до роботи: Наукові та технічні джерела щодо архітектури мереж 5G, технологій CDN (Content Delivery Network) і MEC (Multi-access Edge Computing), протоколів забезпечення QoS, прикладних сценаріїв використання та безпеки в телекомунікаційних системах.

4. Зміст роботи: огляд технології CDN та MEC, аналіз проблем та впливу впровадження протоколів CDN та MEC на якість обслуговування , розробка алгоритму впровадження протоколів CDN та MEC у мережах 5G

5. Перелік ілюстративного матеріалу:

1) Архітектура та технічні характеристики мереж п’ятого покоління;
2) Принципи роботи та сценарії використання CDN і MEC;
3) Проблеми інтеграції, безпеки та хмарної підтримки;
4) Протоколи забезпечення QoS;
5) Методи впровадження протоколів CDN та MEC;
6) Реалізація алгоритму;

6. Дата видачі завдання 20 лютого 2025 р.

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Отримання завдання	20.02.2025	Виконано
2	Збір інформації, розгляд протоколів CDN та MEC, узгодження та затвердження змісту роботи	1.03.2025 - 27.03.2025	Виконано
3	Вивчення та розбір стану, принципу, проблем технологій CDN та MEC у контексті мереж п’ятого покоління	27.03.2025 - 14.04.2025	Виконано
4	Вивчення проблем та впливу впровадження протоколів CDN та MEC на якість обслуговування в 5G-мережах	14.04.2025 - 09.05.2025	Виконано
5	Впровадження протоколів CDN та MEC у 5G-мережі, розробка алгоритму, порівняльний аналіз використання	09.05.2025 - 27.05.2025	Виконано
6	Підготовка матеріалів до друку та оформлення пояснювальної записки.	30.05.2025 - 05.06.2025	Виконано
7	Оформлення дипломної роботи, підготовка презентації для доповіді.	05.06.2025 - 09.06.2025	Виконано

Студент Глєб КНИР

Керівник Валерій ПРАВИЛО

РЕФЕРАТ

Дипломна робота містить 95 сторінок, 19 рисунків. Було використано 56 джерел інформації.

Актуальність роботи. Стрімкий розвиток мобільних технологій, зокрема впровадження мереж п’ятого покоління (5G), супроводжується необхідністю глибокої реконструкції архітектури обробки, доставки та безпеки даних. За своєю природою 5G відрізняється високою щільністю підключень, мінімальними затримками, динамічним управлінням ресурсами та підтримкою масового обміну даними. У таких умовах зростає навантаження на інфраструктуру передачі інформації та зростає потреба в інноваційних технологіях доставки контенту й обробки запитів на краю мережі. Саме тому актуальним є вивчення концепцій CDN (Content Delivery Network) та MEC (Multi-access Edge Computing), які здатні забезпечити якісно новий рівень сервісу у цифрових середовищах.

CDN забезпечує ефективну маршрутизацію контенту до кінцевого споживача, знижуючи затримки та покращуючи стабільність доступу. Технологія MEC, у свою чергу, зміщує обчислювальні потужності ближче до джерела даних - у точку доступу користувача, що істотно підвищує ефективність обробки, дозволяє зменшити затримки й знизити навантаження на центральні хмари.

Мета роботи. - розробити, реалізувати та оцінити ефективність алгоритму впровадження протоколів CDN і MEC у системах мобільного зв’язку п’ятого покоління з урахуванням вимог до якості обслуговування, навантаження й безпеки.

Об’єкт дослідження - процес реалізації протоколів CDN та MEC у мобільних мережах нового покоління.
Предмет дослідження - сукупність технологічних рішень, алгоритмів і протоколів CDN і MEC в архітектурі мереж 5G.

Методи дослідження. У роботі використано поєднання теоретичного моделювання, структурно-функціонального аналізу, методів інженерного проектування та емпіричного тестування.

Отримані результати. На підставі аналізу технічних характеристик 5G, визначення протокольної сумісності та сценаріїв інтеграції, було встановлено, що застосування CDN та MEC у гібридному режимі забезпечує приросту ефективності в обробці запитів при збереженні стабільного QoS.

Галузь застосування. Телекомунікації, якість обслуговування.

Ключові слова: ПРОТОКОЛИ CDN ТА MEC, QoS

ABSTRACT

The thesis contains 88 pages, 19 figures. A total 56 of sources were used.

Relevance of the work. The rapid development of mobile technologies, particularly the deployment of fifth-generation (5G) networks, is accompanied by the need for deep reconstruction of data processing, delivery, and security architecture. By its nature, 5G differs with high connection density, minimal latency, dynamic resource management, and support for massive data exchange. In such conditions, the load on the information transmission infrastructure increases, and there is a growing demand for innovative technologies for content delivery and query processing at the network edge. Therefore, the study of CDN (Content Delivery Network) and MEC (Multi-access Edge Computing) concepts is relevant, as they can provide a qualitatively new level of service in digital environments. CDN ensures efficient routing of content to the end-user, reducing latency and improving access stability. MEC, in turn, shifts computing power closer to the data source — to the user's access point, significantly increasing processing efficiency, reducing latency, and decreasing the load on central clouds.

Goal. To develop, implement, and assess the effectiveness of an algorithm for implementing CDN and MEC protocols in fifth-generation mobile communication systems, considering the requirements for service quality, load, and security.

The object The process of implementing CDN and MEC protocols in next-generation mobile networks.

The subject A set of technological solutions, algorithms, and protocols for CDN and MEC in the architecture of 5G networks.

Research methods. The work uses a combination of theoretical modeling, structural-functional analysis, engineering design methods, and empirical testing.

Results obtained. Based on the analysis of the technical characteristics of 5G, determination of protocol compatibility, and integration scenarios, it was established that the use of CDN and MEC in a hybrid mode provides an increase in query processing efficiency while maintaining stable QoS.

Field of application. Telecommunications, service quality.

Key words: CDN AND MEC PROTICOLS, QoS

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ

CDN	Content Delivery Network
MEC	Multi-access Edge Computing
IoT	Internet of Things
QoS	Quality of Service
QoE	Quality of Experience
ZSM	Zero-touch Service Management
NFV	Network Function Virtualization
RIC	RAN Intelligent Controller
DPI	Deep Packet Inspection
HAS	HTTP Adaptive Streaming
CoAP	Constrained Application Protocol
TTFB	Time to First Byte
ICN	Information-Centric Networking
PFCP	Packet Forwarding Control Protocol
CXTP	Context Transfer Protocol
MEPM	Mbiloe Edge Platform Manager
VIM	Virtual Infrastructure Manager
CU-DU	Centralized/Distributed Units
UPF	User Plane Function
LPPe	LTE Positioning Protocol extensions
TTL	time-to-live
RTT	Round Trip Time
RTO	Retransmission Timeout
RAN	Radio Access Network
RNIS	Radio Network Information Service
VNF	Virtualized Network Function

ВСТУП

Актуальність теми. Стрімкий розвиток мобільних технологій, зокрема впровадження мереж п’ятого покоління (5G), супроводжується необхідністю глибокої реконструкції архітектури обробки, доставки та безпеки даних. За своєю природою 5G відрізняється високою щільністю підключень, мінімальними затримками, динамічним управлінням ресурсами та підтримкою масового обміну даними. У таких умовах зростає навантаження на інфраструктуру передачі інформації та зростає потреба в інноваційних технологіях доставки контенту й обробки запитів на краю мережі. Саме тому актуальним є вивчення концепцій CDN (Content Delivery Network) та MEC (Multi-access Edge Computing), які здатні забезпечити якісно новий рівень сервісу у цифрових середовищах.

CDN забезпечує ефективну маршрутизацію контенту до кінцевого споживача, знижуючи затримки та покращуючи стабільність доступу. Технологія MEC, у свою чергу, зміщує обчислювальні потужності ближче до джерела даних - у точку доступу користувача, що істотно підвищує ефективність обробки, дозволяє зменшити затримки й знизити навантаження на центральні хмари. Їх інтеграція в архітектуру 5G-мереж формує нову парадигму цифрових сервісів, що передбачає швидкість реагування в реальному часі, розподілену інфраструктуру, захищені канали передачі та динамічну маршрутизацію. Такі особливості відкривають нові можливості для застосування у сфері автоматизованого транспорту, телемедицини, промислового інтернету речей, доповненої реальності.

Огляд наукової літератури засвідчує, що питання ефективної інтеграції CDN і MEC у 5G-мережі перебуває в центрі уваги світової технічної спільноти. У роботах 3GPP (TS 23.501, TS 38.104, TS 38.801) систематизовано специфікації архітектури 5G, технічні вимоги до базових станцій та принципи радіодоступу. Кріс Гофман, Шатруган Сінгх та Флінн Кевін висвітлюють базові технічні параметри 5G, його переваги над попередніми поколіннями та

виклики, пов’язані з реалізацією. Особливе місце займають роботи щодо проблем безпеки, зокрема Правила В.В. та Кормульова О.С., які аналізують механізми захисту в умовах хмарної архітектури. Незважаючи на велику кількість досліджень, бракує системного аналізу сумісності та ефективності впровадження CDN і MEC у практику побудови 5G-мереж. Це обумовлює доцільність проведення комплексного дослідження, спрямованого на розробку прикладного алгоритму реалізації цих протоколів у реальному цифровому середовищі.

Об’єкт дослідження - процес реалізації протоколів CDN та MEC у мобільних мережах нового покоління. **Предмет дослідження** - сукупність технологічних рішень, алгоритмів і протоколів CDN і MEC в архітектурі мереж 5G.

Мета дослідження - розробити, реалізувати та оцінити ефективність алгоритму впровадження протоколів CDN і MEC у системах мобільного зв’язку п’ятого покоління з урахуванням вимог до якості обслуговування, навантаження й безпеки.

Для досягнення поставленої мети передбачається розв’язати такі завдання:

- дослідити архітектуру та технічні характеристики мереж п’ятого покоління;
- охарактеризувати принципи функціонування технологій CDN та MEC і провести їх порівняльний аналіз;
- описати реальні сценарії застосування CDN і MEC для покращення параметрів обслуговування;
- ідентифікувати проблеми інтеграції, безпеки та хмарної сумісності в сучасних реалізаціях систем CDN і MEC;
- узагальнити результати сучасних досліджень та позначити перспективи розвитку цих технологій у контексті 5G;
- систематизувати протоколи, що підтримують CDN і MEC у мобільних мережах нового покоління;

- дослідити механізми забезпечення якості обслуговування в умовах їх використання;
- проаналізувати моделі оптимізації навантаження та розподілу ресурсів;
- оцінити загрози і вразливості, притаманні впровадженню CDN і MEC;
- сформулювати рекомендації щодо безпечної та ефективної реалізації даних протоколів;
- розробити алгоритм впровадження та оцінити його практичну ефективність у порівняльному аналізі.

Методи дослідження. У роботі використано поєднання теоретичного моделювання, структурно-функціонального аналізу, методів інженерного проектування та емпіричного тестування. Застосовано інструменти аналізу архітектури мережі, моделювання навантажень, аналізу якості обслуговування (QoS) та імітації сценаріїв реального використання. Для оцінки ефективності алгоритму впроваджено методи порівняльного аналізу, обчислювального моделювання та графічної візуалізації результатів.

Структура роботи. Робота складається зі вступу, трьох розділів, п’ятнадцяти підрозділів, висновків і списку використаних джерел.

РОЗДІЛ 1.
ТЕХНОЛОГІЇ CDN ТА МЕС У КОНТЕКСТІ МЕРЕЖ 5G: СТАН, ПРИНЦИПИ, ПРОБЛЕМИ

1.1. Мережі п’ятого покоління: архітектура та технічні характеристики

У п’ятому поколінні мобільних мереж закладено фундамент для трансформації всієї екосистеми цифрової комунікації - від фізичної архітектури до логіки управління трафіком і динаміки розгортання сервісів у реальному часі. Архітектура 5G орієнтована на модульну побудову, де замість статичних підсистем застосовуються гнучкі компоненти, що розподіляються просторово залежно від характеру трафіку, щільності користувачів і сервісної логіки. Особливістю є централізація обробки сигналів завдяки концепції C-RAN - централізованої архітектури радіодоступу, де фізичні радіомодулі розміщені по периметру території обслуговування, а інтелектуальні блоки переносяться у віртуалізовані дата-центри. Це дозволяє не лише знизити витрати на обладнання, а й динамічно керувати спектром частот, змінюючи ресурси у відповідь на зміни в мобільності чи інтенсивності запитів. Завдяки підтримці міліметрового діапазону частот (mmWave) зростає смуга пропускання, а отже - швидкість передачі, що в окремих випадках може досягати 10 Гбіт/с при затримці, яка не перевищує 1 мс, що робить 5G середовищем придатним для автономного транспорту, телемедицини й індустріального інтернету речей [7, с. 15].

Фізичний рівень 5G визначається значно складнішою топологією, ніж попередні покоління. Замість суцільного покриття класичними макросотами з високою потужністю, тут формується гетерогенне середовище з безліччю малопотужних вузлів: пікосоти, фемтосоти, ретранслятори й інтелектуальні поверхні, що відображають сигнал - усе це інтегрується у спільну інфраструктуру з метою мінімізації втрат на дифракцію й розсіювання. Мережа стає динамічною - вона здатна підлаштовуватися під середовище,

адаптувати маршрути трафіку в обхід перевантажених або недоступних зон. Це дозволяє підтримувати стабільне з'єднання в урбанізованих середовищах високої щільності, де кількість пристроїв може перевищувати мільйон одиниць на квадратний кілометр. Масштабованість закладена вже на рівні протоколів: завдяки логічному розділенню доменів управління - користувацьких, контрольних і транспортних - мережа може бути кастомізована під потреби конкретного сегмента - від розумних фабрик до потокових медіасервісів з високою роздільною здатністю [5, с. 7].

Принципова трансформація відбувається і на рівні мережевої віртуалізації. У 5G реалізовано так званий network slicing - поділ єдиної фізичної інфраструктури на незалежні логічні канали, кожен з яких має свої параметри затримки, пропускної здатності, пріоритетності. Це означає, що той самий базовий сегмент може паралельно обслуговувати автоматизовану виробничу лінію з вимогами до затримки у 0,5 мс і одночасно - потоковий відеосервіс зі швидкістю 4 Гбіт/с, при цьому жоден з каналів не заважає іншому. Додаткову гнучкість забезпечує інтеграція з MEC - мобільними обчисленнями на периферії, які дозволяють виконувати обробку даних ближче до джерела їхнього генерування. Це знижує потребу в централізованих обчислювальних ресурсах і зменшує затримки, що особливо актуально для розумних міст і транспортних систем реального часу.

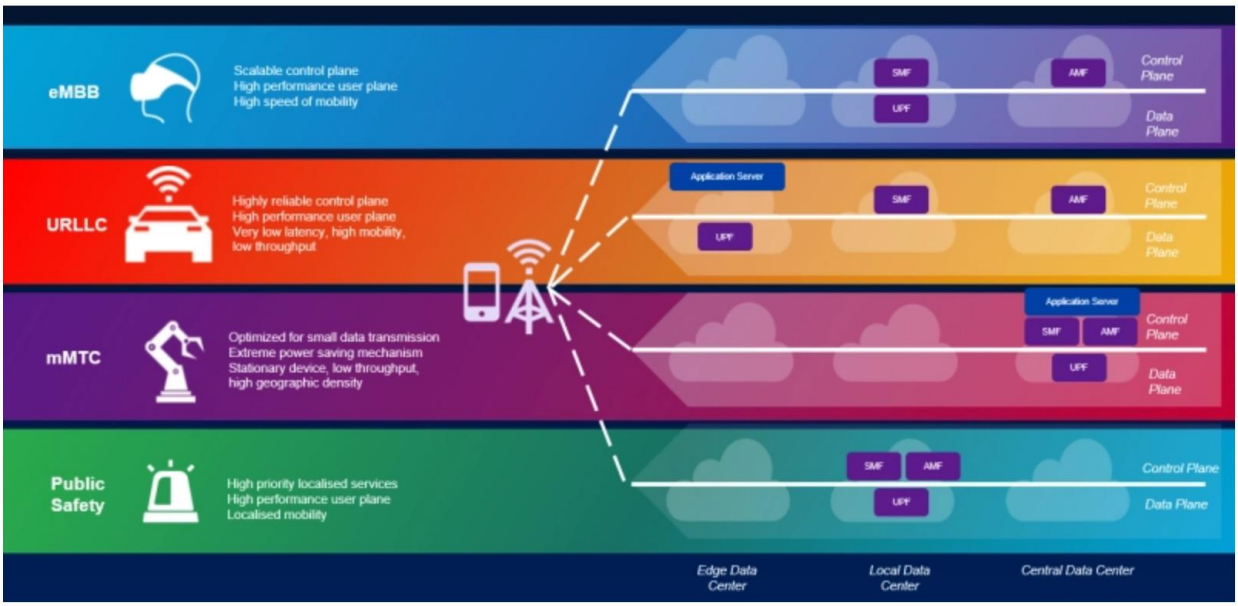

Рис. 1.1 5G Network Slicing

Суттєвою відмінністю 5G є активне застосування інтелектуальних алгоритмів у всіх шарах мережі - від планування розміщення антен до адаптивного керування ресурсами спектра й прогнозування навантаження. Машинне навчання використовується для виявлення аномалій у трафіку, автоматичної оптимізації маршрутизації та адаптації до поведінки користувачів. Завдяки цьому мережа набуває характеристик самоорганізації - вона може перебудовувати топологію в реальному часі залежно від географічного переміщення користувачів, доступності спектру й енергетичних параметрів вузлів. У контексті енергетичної ефективності, 5G впроваджує концепції energy-aware routing, тобто динамічного вибору маршрутів з урахуванням споживання енергії та рівня залишкової потужності на вузлах, що має істотне значення для сталих мереж з автономним живленням [11, с. 9].

Технічна специфікація мереж п’ятого покоління розширює класичну модель OSI завдяки введенню додаткових функціональних шарів, які відповідають за політики доступу, якість обслуговування (QoS) і безпекову адаптацію. Підхід до безпеки у 5G базується не лише на шифруванні трафіку, а й на концепції zero trust - тобто кожен елемент, включно з внутрішніми компонентами, розглядається як потенційне джерело загрози. Для цього реалізовані протоколи багатофакторної ідентифікації, криптографічного обміну ключами, а також алгоритми аномального поведінкового аналізу. Протокол управління доступом до мережі (AMF) працює у тісній зв’язці з політичним контролером (PCF), що дає змогу тонко налаштовувати пріоритетність сервісів, обмежувати небажані підключення й формувати адаптивну політику доступу для кожного профілю користувача.

1.2. Технологія CDN та мережеві обчислення на краю: принципи та порівняння

У системах розподіленої доставки контенту - CDN - закладено принцип географічного дублювання інформаційних ресурсів шляхом стратегічного розміщення кешованих копій контенту у так званих edge-вузлах, розосереджених по периметру глобальної мережі. Це дозволяє здійснювати доступ до необхідного ресурсу з найближчого за топологією сервера, що мінімізує затримку, зменшує завантаженість магістральних каналів і знижує ризик перевантаження центральної інфраструктури. CDN працює з попередньо згенерованим або статичним контентом - відео, зображеннями, архівами, часто відвідуваними вебсторінками, що мають стабільну структуру запитів. Алгоритми кешування визначають, які об’єкти мають бути збережені в edge-вузлах, враховуючи частотність доступу, геолокацію користувача й контентні пріоритети. Саме завдяки такому принципу можливо забезпечити стабільну продуктивність навіть у випадках пікового навантаження або географічно розосередженого трафіку, що актуально для стримінгових платформ, великих маркетплейсів і розподілених мультимедійних сервісів. Цей підхід орієнтований на оптимізацію доставки, однак не передбачає гнучкої логіки обробки даних безпосередньо в точках генерації трафіку [8, с. 5].

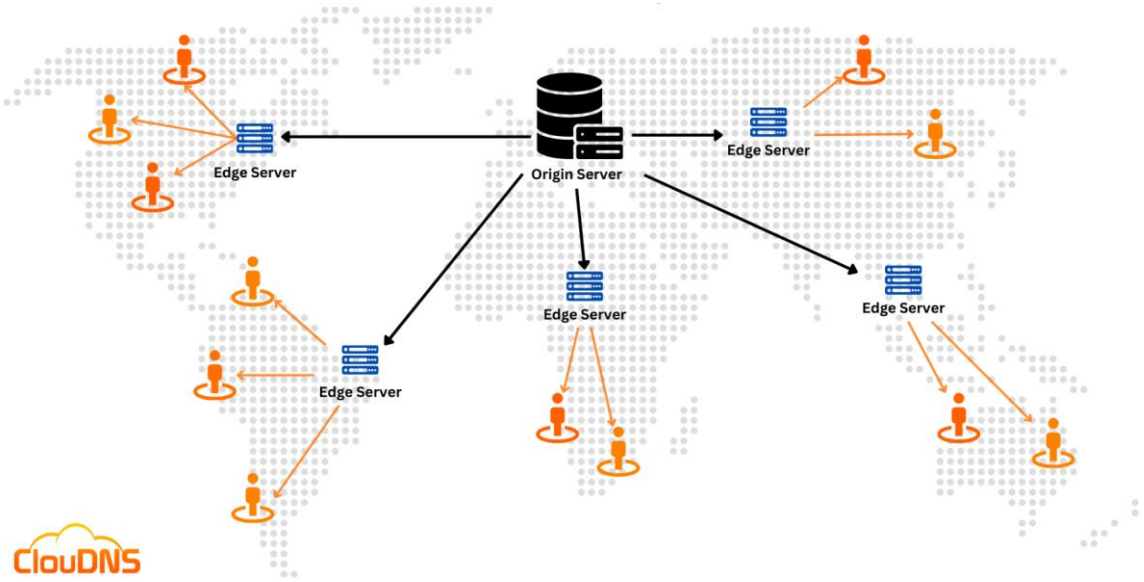

Рис. 1.2 Content Delivery Network

Натомість MEC - мультидоступні обчислення на краю - радикально змінює логіку взаємодії між пристроєм користувача та обчислювальною інфраструктурою, дозволяючи переносити обробку, агрегацію й аналітику даних максимально близько до місця їх виникнення. Це досягається шляхом інтеграції edge-серверів безпосередньо у вузли мобільного зв'язку або локальні інфраструктурні точки - маршрутизатори, базові станції, локальні центри зв'язку. У такій архітектурі дані, замість того щоб передаватися до центрального дата-центру, обробляються у периферійному вузлі, що зменшує затримку, знижує навантаження на ядро мережі й дозволяє реалізувати сервіси реального часу - адаптивні системи управління, автономний транспорт, AR/VR, індустріальну робототехніку. MEC не є альтернативою CDN - це скоріше доповнення з протилежною логікою: не зберігання, а динамічна обробка; не контент, а поведінкові дані; не кеш, а обчислення. У MEC контент не накопичується - він формується, обчислюється, аналізується на місці й миттєво інтегрується у відповідні системи прийняття рішень або відгуку на події [13, с. 39].

Така диференціація зумовлює функціональні відмінності: CDN забезпечує сталість і ефективність доступу до вже сформованих даних, а MEC - гнучкість і адаптивність обробки в реальному часі. У практичній реалізації CDN зазвичай вибудовується на протоколах доставки HTTP/HTTPS з підтримкою TLS-акселерації, механізмами DNS-редиректу або anycast-архітектурою, що направляє користувача до найближчого вузла кешу. Натомість MEC використовує внутрішньомережеві API, хмарні мікросервіси, low-latency transport layer та контейнери, які можна оркеструвати за допомогою Kubernetes або аналогічних платформ. У той час як CDN забезпечує масштабованість для великої кількості однотипних запитів, MEC працює з динамічними потоками, які не мають повторюваної структури - телеметрія, біометрія, відеопотоки з аналізом руху, складські сенсори з варіативною логікою.

CDN добре інтегрується з глобальними OTT-сервісами, платформами дистрибуції мультимедіа та великими інформаційними порталами, де головна задача - прискорити доступ до ресурсу, не змінюючи його змісту. MEC, натомість, інтегрується з вертикалями, де необхідна миттєва реакція на зміни середовища - агроіндустріальні комплекси, логістичні хаби, транспортні системи з автономною навігацією, телемедичні комплекси, що вимагають обробки сигналу у момент його генерації. Обчислювальна архітектура MEC дозволяє виконувати передобробку, очищення, семантичне нормування та класифікацію даних, перш ніж вони потрапляють у централізовану інфраструктуру, знижуючи навантаження й забезпечуючи контекстну релевантність. Це відкриває можливості для побудови адаптивних, самонавчальних інфраструктур - зокрема в розподілених системах ШІ, де локальне навчання моделей виконується на краю з подальшою агрегацією у федеративному середовищі [16, с. 12].

У поєднанні з мережею 5G, потенціал MEC зростає експоненційно: низька затримка, гарантована пропускна здатність і логічна сегментація трафіку дозволяють створювати кастомізовані середовища для кожного типу сервісу. Якщо CDN працює на зниження часу завантаження відео, то MEC - на підтримку безперервності потокової обробки сенсорних даних у smart-city або надійного керування критичними об’єктами інфраструктури. Обидва підходи не суперечать одне одному - вони утворюють паралельні шари сучасної цифрової екосистеми: CDN формує її інформаційний каркас, MEC - нейронне ядро, здатне до обчислень і адаптацій. Відтак, архітектура майбутнього - це гібридна модель, де статична доставка й динамічна обробка поєднуються на рівні інтелектуальних протоколів і синхронізованих систем, а центр тяжіння переноситься з глобального ядра до розумного краю - edge intelligence.

1.3. Сценарії застосування CDN і MEC для підвищення якості обслуговування

У сучасній парадигмі розгортання цифрових сервісів дедалі виразніше спостерігається зсув у бік архітектур, що поєднують централізовану та периферійну обробку даних. Саме тут проявляється синергія між технологіями CDN (Content Delivery Network) та MEC (Multi-access Edge Computing), які, хоч і розвивалися в різних напрямках, сьогодні формують нову якість обслуговування цифрових продуктів - із врахуванням геопросторової близькості, динаміки трафіку, контексту споживання і необхідності мінімізувати латентність. CDN, орієнтовані на кешування та географічне наближення статичного контенту, вже тривалий час лежать в основі доставки відео, зображень, документів та інших об’єктів, критичних для фронтенд-завантаження. Їх використання особливо виражене у відеострімінгу, де вимога до буферизації та швидкого старту програвання стає вирішальною у боротьбі за утримання уваги користувача. У свою чергу, MEC надає потужний інструментарій для обробки динамічних, реальних подій, безпосередньо в місцях виникнення трафіку - базових станціях, крайових датацентрах, граничних вузлах операторських мереж. Це дає змогу значно скоротити затримки у передачі даних, уникнути заторів у магістральних каналах і забезпечити негайну реакцію на зміну стану об’єктів, що взаємодіють у режимі реального часу - від безпілотного транспорту до медичних систем дистанційного нагляду [1, с. 15].

Поєднання CDN і MEC у гібридних сценаріях демонструє еволюційний перехід від простої доставки контенту до інтелектуального обслуговування даних з урахуванням когнітивних патернів поведінки користувача. У випадку стрімінгових сервісів типу Netflix або Twitch CDN відповідає за надання оптимізованої копії відео з найближчого кеш-сервера, що знижує навантаження на первинний сервер і забезпечує низький час завантаження, тоді як MEC може аналізувати шаблони перегляду, підлаштовуючи якість відео у реальному часі залежно від поточних умов мережі, типу пристрою та навіть поведінкових факторів споживача. У сфері онлайн-ігор поєднання цих технологій надає унікальні переваги - CDN забезпечує швидке завантаження

оновлень, текстур, карт, тоді як MEC обробляє ігрову телеметрію, прогнозує сценарії гри, відгукується на рухи суперників у багатокористувацьких сесіях, забезпечуючи мізерну латентність і відсутність лагів. Це стає особливо значущим у жанрах FPS або RTS, де навіть кількамілісекундна затримка може змінити хід гри. У цій конфігурації CDN виконує пасивно-доставну функцію, тоді як MEC стає активним когнітивним прошарком, що реагує на запити динамічного середовища [10, с. 18].

Застосування CDN у сфері електронної комерції дозволяє знизити навантаження на головні сервери під час пікових періодів - святкові розпродажі, акції з обмеженим часом - завдяки попередньому розповсюдженню даних про товари, зображень, скриптів на численні кеш-вузли по всьому світу. Це забезпечує високу доступність сервісу, навіть коли кількість користувачів зростає експоненційно. У таких умовах MEC виступає як адаптивний посередник, який здатен оперативно аналізувати навантаження в певному регіоні, виконувати логіку рекомендаційних систем, здійснювати фільтрацію запитів на основі попередньо прорахованих моделей поведінки. Внаслідок цього платформа електронної торгівлі функціонує не як реактивна система обслуговування запитів, а як проактивна цифрова інфраструктура з когнітивним підходом до формування клієнтського досвіду. Тут MEC забезпечує персоналізацію в реальному часі, тоді як CDN гарантує безперебійну доставку фронтенду.

У випадку з IoT-екосистемами з високою щільністю сенсорних вузлів - агропромислові системи, «розумні» міста, енергетичні комплекси - MEC дозволяє переносити обчислювальні навантаження з централізованих датацентрів на крайові вузли, наближені до джерел даних. Це не лише мінімізує затримки, а й розвантажує міжмережеву інфраструктуру, знижуючи ризики заторів, пов’язаних із транспортуванням великих обсягів телеметрії. Водночас CDN у таких сценаріях відіграє функцію реплікації стандартних прошивок, оновлень ПЗ, шаблонів конфігурацій, які регулярно передаються на масиви IoT-пристроїв. У комбінації обидві технології дозволяють створити

самодостатній інформаційно-операційний шар, де стандартна інформація миттєво доставляється по CDN, а змінна, специфічна - обробляється MEC у режимі реального часу. Це формує передумови до створення автономних розподілених мереж, здатних реагувати на зміну кліматичних, енергетичних або транспортних параметрів із точністю до хвилини [3, с. 4].

Розвиток мобільного зв'язку п'ятого покоління (5G) ще більш ускладнив архітектуру обслуговування даних, зробивши вимоги до швидкості реакції та розподілу трафіку надзвичайно жорсткими. У таких умовах симбіоз MEC і CDN став архітектурною необхідністю. MEC інтегрується безпосередньо в мережеву інфраструктуру 5G, надаючи базовим станціям можливість не лише пересилати пакети, а й обробляти контент, передбачати сценарії поведінки користувача, здійснювати управління QoE (Quality of Experience) на рівні мікросегментів. У свою чергу, CDN виступає механізмом агрегованого доступу до історично стабільного контенту, завдяки чому кінцева точка отримує і динамічну, і статичну інформацію без необхідності звернення до віддалених датацентрів. У сфері відеоспостереження це означає, що MEC миттєво ідентифікує рухи чи аномалії на камерах, розміщених у периметрі об'єкта, а CDN транслює фрагменти архівного відео або сценарні шаблони обробки. Це дозволяє скоротити час реакції охоронних систем з хвилин до секунд.

Додатковим прикладом є автотранспортні системи нового покоління, зокрема автономні транспортні засоби, де CDN забезпечує безперервне оновлення навігаційних карт, базових алгоритмів розпізнавання об'єктів, загальнодоступних моделей прийняття рішень, а MEC відповідає за обчислення в межах кожної конкретної зони покриття - аналіз трафіку на перехресті, прогнозування руху пішоходів, динамічне оновлення маршрутів у разі виникнення заторів. У таких конфігураціях MEC відіграє роль локального «мозку», що миттєво реагує на зміну дорожньої ситуації, тоді як CDN гарантує безперервність глобальної інформаційної структури, забезпечуючи системну узгодженість прийнятих рішень із великою базою знань. Такий підхід дозволяє

сформувати цілісне цифрове середовище, яке адаптується до поведінки кожного учасника дорожнього руху [14, с. 35].

Роль MEC у забезпеченні сервісів доповненої та віртуальної реальності проявляється особливо яскраво в імерсивних платформах - індустрії освіти, телемедицини, віртуальних екскурсій. Тут затримка в передачі даних навіть у межах 100 мілісекунд здатна зруйнувати ефект присутності та спричинити так звану «кібернудоту». У таких випадках MEC бере на себе функції рендерингу, попереднього прогнозування руху користувача, обчислення візуальних ефектів із мінімальним часом відгуку. Паралельно CDN здійснює швидке завантаження об’ємних бібліотек візуального контенту - моделей, текстур, фонів - з найближчих кеш-серверів. У цій парі CDN виступає джерелом об’ємних, але стабільних елементів, тоді як MEC реалізує динамічне оновлення простору згідно з діями користувача, забезпечуючи плавну зміну сцен, об’єктів і перспектив.

1.4. Проблеми інтеграції, безпеки та хмарної підтримки в системах CDN та MEC

Інтеграція технологій CDN та MEC у рамках високонавантажених цифрових інфраструктур вимагає не лише інженерної точності, а й стратегічного системного підходу до оркестрації компонентів, сумісності протоколів та уніфікації інтерфейсів взаємодії між компонентами на різних рівнях мережі. Особливо складною є задача координації між різнотипними учасниками екосистеми - телекомунікаційними операторами, постачальниками контенту, розробниками edge-рішень, провайдерами хмарних платформ і власниками датацентрів. Ці технології використовують відмінні підходи до побудови сервісів, мають власні політики обробки даних, різні схеми SLA та методи контролю навантаження. Унаслідок цього виникає ситуація багатошарової фрагментації, коли навіть у межах одного регіону важко забезпечити повноцінну інтеграцію CDN та MEC у єдину логічну мережеву структуру. Відсутність єдиних стандартів для інтероперабельності

API, конфігураційних моделей та політик оркестрації призводить до того, що локальні вузли стають вузькими місцями або навіть точками конфлікту, коли сервіс, розгорнутий на MEC-платформі одного вендора, не може ефективно взаємодіяти з CDN-інфраструктурою іншого. Це породжує латентні відмови, труднощі у масштабуванні та високі витрати на адаптацію [2, с. 6].

У розподілених середовищах, які поєднують CDN і MEC, особливої уваги потребує питання безпеки на краю мережі. Оскільки периферійні вузли мають обмежені обчислювальні ресурси, але обробляють високочутливі дані в реальному часі, вони стають пріоритетними цілями для атак. Типовими загрозами є атакування вразливих API, що використовуються для оркестрації та управління сервісами, експлуатація недоліків у верифікації запитів, перехоплення незашифрованого трафіку або ін’єкції шкідливих даних у логіку сервісу. Зважаючи на географічну розосередженість MEC-вузлів, класичні методи захисту - як-от централізовані системи моніторингу або інструменти firewall - стають недостатніми. Потрібне впровадження багаторівневих моделей безпеки, які охоплюють як самі вузли, так і шляхи комунікації між ними. Сюди входить шифрування на рівні TLS 1.3 або вище, динамічні політики доступу з елементами контекстної автентифікації, ізоляція функціональних доменів за допомогою віртуалізації або контейнеризації, а також інтеграція систем виявлення аномалій на основі машинного навчання, здатних оперативно ідентифікувати відхилення в поведінці трафіку [6, с. 14].

Невирішеність питань хмарної підтримки у сценаріях із CDN та MEC також створює багато ризиків. Попри широке використання хмарних платформ як бекенду для обслуговування MEC-сервісів, виникає парадокс так званої «хмарної інверсії» - коли віддалені обчислення починають дублювати або конфліктувати з локальними, генеруючи надмірний трафік і сповільнюючи реакцію на події. У деяких випадках хмарна платформа, обслуговуючи MEC-сервіс, сама стає точкою збоїв - через перевантаження, конфігураційні помилки або втрату синхронізації з edge-серверами. Проблема ще більш загострюється у випадках гібридного хмарного середовища, коли

MEC-функції частково реалізуються у приватній інфраструктурі, а частково - у публічній хмарі. Відсутність уніфікованої оркестраційної моделі для гібридних середовищ спричиняє розсинхронізацію обчислювального навантаження, що веде до погіршення QoS (Quality of Service) і неможливості дотримання SLA для критичних сервісів. Саме тому виникає потреба у створенні так званих cloud-native edge-фреймворків, які могли б забезпечити прозору інтеграцію між MEC-вузлами і хмарними платформами за допомогою API-гейтувеїв, динамічних конфігурацій та знижених вимог до латентності.

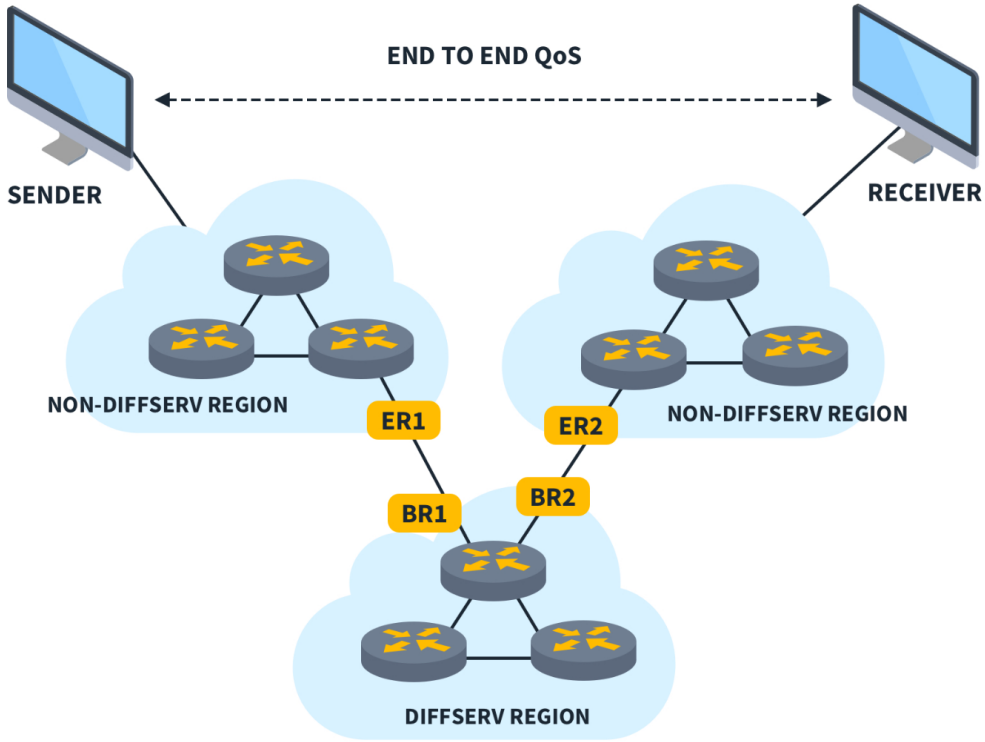

Рис. 1.3 QoS (Quality of Service)

Надзвичайно актуальним стає питання кіберсуверенітету в умовах CDN–MEC-синергії. У багатьох країнах використання хмарних сервісів, особливо зарубіжного походження, накладає законодавчі обмеження на передачу, зберігання та обробку персональних даних. Це стосується не лише загальних регламентів типу GDPR, а й локальних нормативів - як-от закони про критичну інфраструктуру, які регулюють розміщення даних про енергетику, транспорт чи оборону. CDN-сервери, які автоматично

розповсюджують кешований контент у різні юрисдикції, можуть порушити принципи територіальної обробки, тоді як MEC-вузли, що взаємодіють з такими серверами, потрапляють у ситуацію «юридичної розмитості». Вирішення цієї проблеми потребує впровадження геофенсингу - технології, яка обмежує географію обробки певних класів даних, автоматично адаптуючи маршрутизацію контенту відповідно до законодавчих норм. У поєднанні з механізмами знеособлення (анонімізації) даних це дозволяє уникнути порушення регуляторних актів та підвищити довіру до MEC-CDN-інфраструктури з боку державних та корпоративних замовників [9, с. 3].

Ще одним бар'єром до ефективного впровадження є недостатня адаптивність існуючих протоколів маршрутизації до потреб edge-архітектур. Традиційні мережеві протоколи (OSPF, BGP, EIGRP) були спроєктовані для централізованих топологій, де маршрути статичні або слабо варіативні. Проте у MEC-середовищах, де вузли можуть динамічно змінюватися, де обчислювальні потужності мігрують залежно від навантаження, така жорсткість веде до затримок у перепрокладанні маршрутів, неефективного балансування навантаження та низької відмовостійкості. CDN частково компенсує це за рахунок DNS-based rerouting, але ця модель погано працює у випадках, коли рішення мають прийматися на рівні кількох мілісекунд. Потрібне створення нового класу протоколів - edge-aware routing - які враховували б не лише топологію, а й параметри обчислювального середовища: CPU-навантаження, об'єм доступної пам'яті, тепловий профіль, енергетичну доступність тощо.

Узгодження QoE між CDN і MEC - ще одна точка напруження. У класичних моделях CDN орієнтується на вимірювання середньої затримки, швидкості завантаження та відсотку помилок при доставці контенту, тоді як MEC-метрики зосереджуються на затримці обчислення, часі відповіді сервісу та стабільності підключення. Відсутність єдиного стандарту для агрегування цих метрик призводить до складності у верифікації кінцевої якості послуги - одна система може вважати сервіс прийнятним, тоді як інша - непридатним до

експлуатації. Це створює конфлікт в моделі SLA, де необхідно впроваджувати узгоджені множинні метрики, які інтегрують як параметри доставки контенту, так і параметри його обробки. Такі підходи вже починають реалізовуватись у фреймворках типу ETSI ZSM (Zero-touch Service Management), але вони ще не мають належного поширення та потребують глибокої модифікації під специфіку MEC-середовища.

1.5. Сучасні дослідження та перспективи розвитку технологій у 5G-мережах

Розвиток 5G-технологій сьогодні концентрується не лише на зростанні пропускної здатності, зниженні затримок або розширенні спектра частот, а й на формуванні принципово нової інфраструктурної парадигми, де межа між ядром мережі, периферією й хмарою втрачає свою жорсткість. Найновіші дослідження у сфері архітектури 5G акцентують на гібридизації - інтеграції обчислювальних можливостей edge, розподіленого зберігання даних, інтелектуальних транспортних протоколів і автономної оркестрації сервісів. Поєднання CDN та MEC у межах однієї логіки стало відправною точкою для створення мереж з автоматизованим перерозподілом ресурсів, де завдання не мають фіксованого місця виконання, а мігрують відповідно до змін у топології, поведінці користувача чи навантаженні на конкретні вузли. Такий підхід вимагає не лише масштабованості, а й когнітивної адаптивності - системи мають розпізнавати сигнатури трафіку, прогнозувати події, приймати рішення про релокацію процесів без втручання адміністратора. Це стало можливим завдяки широкому впровадженню методів машинного навчання, включно з reinforcement learning, що навчає мережу самостійно коригувати правила маршрутизації, балансування навантаження й шифрування залежно від контексту користувача та технічних характеристик середовища [15, с. 4].

Паралельно відбувається еволюція гетерогенних мереж - таких, що поєднують різнотипні радіотехнології, спектри, інфраструктурні компоненти й протоколи. У межах 5G-мереж це означає підтримку як низькочастотних

діапазонів для широкого покриття, так і міліметрових хвиль для високошвидкісних зон із високою щільністю пристроїв. Гетерогенність проявляється також у комплементарності обчислювальних моделей - edge, fog, cloud - які тепер розглядаються не як конкуренти, а як елементи єдиної обчислювальної структури. У цьому середовищі MEC та CDN не просто співіснують - вони функціонально зливаються: MEC забезпечує негайну обробку запитів і аналітику в реальному часі, тоді як CDN дбає про узгодженість, стабільність і повторне використання популярного контенту. У перспективі саме гетерогенна мережа стає базовою платформою для таких застосувань, як масове підключення IoT-пристроїв (до 1 млн/км², згідно з оцінками CLX Forum), мобільна хірургія, управління автономним транспортом та військові комунікації з елементами кіберманеврування.

В горизонті 2030-х років простежується виразна тенденція до формування самонавчальних edge-мереж, які здатні до автономного прийняття рішень, самовідновлення після збоїв і самоконфігурації топології. У таких мережах MEC-платформи перестають бути пасивними вузлами обробки й трансформуються на когнітивні модулі, що володіють функцією розпізнавання патернів, автоматичної ідентифікації критичних ситуацій, реконфігурації політик доступу й адаптації QoS на рівні окремого сегменту. Алгоритми типу federated learning дають змогу проводити навчання без необхідності централізованого збору даних, що критично важливо в умовах законодавчих обмежень на передачу персональних даних. Це уможливлює формування edge-моделей штучного інтелекту, що відображають локальну специфіку поведінки користувачів або об’єктів - у розумних містах, енергетичних системах, системах захисту або агротехнічних рішеннях. У цій архітектурі роль CDN зміщується у бік дистрибуції pre-trained моделей, оновлень, шаблонів рішень, що функціонують як еталони, до яких MEC здійснює локальну адаптацію.

Поглиблюючи цю логіку, сучасні дослідження зосереджені на проєктуванні автономних сервісів із повною edge-компетенцією - таких, що

функціонують без зворотного зв'язку з центральною хмарою. Це дозволяє уникати затримок, викликаних передачею великих обсягів даних, а також забезпечити локальний контроль, що критично для оборонних та критичних інфраструктур. У військових системах та безпілотних зонах автономність обчислень стає фактором виживання - підрозділи повинні зберігати операційну спроможність навіть при втраті зв'язку з центром. На цьому фоні дослідження концентруються на способах формування edge-кластерів, які здатні колективно аналізувати ситуацію, ухвалювати рішення та передавати завдання між собою за динамічно адаптивною логікою. Така автономна MEC-мережа потребує протоколів, що включають елементи колективного навчання, взаємної верифікації та резильєнтного обміну. Тут виникає простір для застосування технологій блокчейну, які дозволяють формалізувати процедури дистрибуції довіри, автентифікації взаємодій і фіксації змін без централізованого реєстру [17, с. 10].

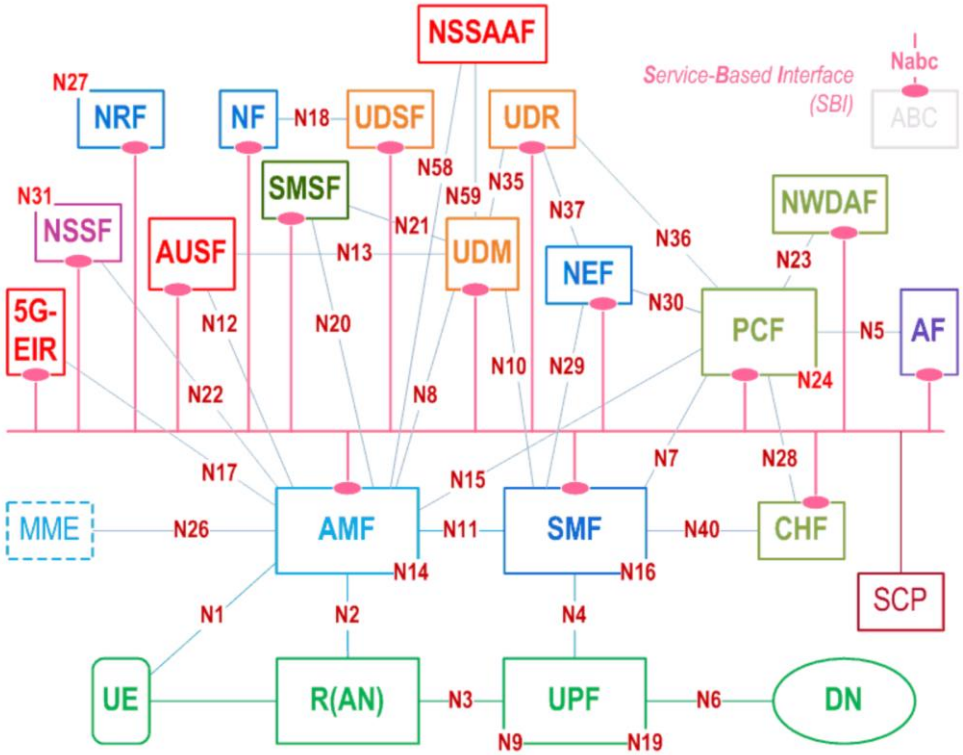

Рис. 1.4 TS 23.50

В контексті еволюції 5G у напрямі 5G-Advanced і майбутнього 6G, зростає зацікавлення мультидоменною оркестрацією - здатністю мережевих

компонентів не лише координувати між собою потоки даних, а й управляти різнотипними сервісами через абстракцію логіки їхньої роботи. Це означає перехід від ієрархічної структури до адаптивної, де ресурси не жорстко розподілені, а динамічно орендуються. MEC у цій моделі функціонує як мікрохмара з обмеженим часом життя, що може створюватися, масштабуватись і ліквідовуватись залежно від миттєвих потреб - події, навантаження або сценарного контексту. Це призводить до того, що інфраструктура втрачає сталість і набуває характеристик рідини - вона плине за потребою, за логікою, за попитом. CDN у таких умовах перетворюється на систему передбачуваного кешування, де завдяки інтелектуальному аналізу попередніх запитів вдається розташувати контент напередодні його використання, фактично реалізуючи концепцію zero-wait delivery [4, с. 18].

Сучасні технічні специфікації 3GPP, такі як TS 23.501 та TR 38.801, формалізують базові принципи архітектури 5G, однак дослідницька думка вже виходить за межі закладених у них моделей. Пропонуються нові топології доступу з нелінійною маршрутизацією, де дані не рухаються від точки А до точки Б, а розділяються на інформаційні контури, які перебувають у стані когнітивної взаємодії. Такі системи самі вибирають маршрут, спосіб доставки, формат шифрування, інтенсивність перевірки автентичності - залежно від ідентифікатора джерела, типу трафіку, рівня довіри, історичних патернів. Це не просто самоорганізація, а когнітивна самоадаптація - ознака переходу до нейромережевого типу комунікації, де мережа починає відображати структуру людської когніції. В цьому контексті CDN та MEC стають не інструментами доставки або обробки, а елементами штучної нейронної мережі, в якій кожен вузол - це не просто точка проходження, а вузол прийняття рішень.

Висновки

У межах розділу було здійснено комплексний аналіз технічної та концептуальної еволюції мереж п’ятого покоління у взаємозв’язку з технологіями CDN та MEC, що визначають характер сучасної цифрової

інфраструктури. Розглянуто архітектурну модель 5G як багаторівневу систему із гнучкою топологією, здатною забезпечити наднизькі затримки, масштабовану обробку трафіку й підтримку масового підключення IoT-пристроїв. Виявлено суттєві розбіжності у функціональних характеристиках MEC і CDN - перша забезпечує миттєву обробку даних на периферії, друга - доставку статичного або повторюваного контенту з географічно розподілених вузлів. Уточнено прикладні сценарії, у яких ці технології використовуються спільно або автономно, що дозволяє підвищити якість користувацького досвіду в медіа, транспорті, ігровій індустрії та міському управлінні. Проаналізовано спектр викликів - від несумісності стандартів до проблем безпеки та уразливості API, що потребує інтеграції багаторівневих захисних механізмів. Завершальним елементом став огляд тенденцій, згідно з якими MEC і CDN поступово об’єднуються в адаптивні гібридні системи, що керуються штучним інтелектом, формуючи передумови до автономних, когнітивно здатних мереж, здатних до самонавчання, самоконфігурації та реактивної адаптації.

РОЗДІЛ 2.
АНАЛІЗ ПРОБЛЕМ ТА ВПЛИВУ ВПРОВАДЖЕННЯ ПРОТОКОЛІВ CDN ТА MEC НА ЯКІСТЬ ОБСЛУГОВУВАННЯ В 5G-МЕРЕЖАХ

2.1. Вступ

В умовах експоненційного зростання споживання мобільного контенту, підвищення ефективності обслуговування у 5G-мережах перестає бути факультативним вектором технологічного розвитку. Із запровадженням масового підключення розумних пристроїв, відеострімінгу у форматах 4K і 8K, систем дистанційного управління, доповненої та віртуальної реальності, зростає тиск на транспортну інфраструктуру оператора. Спектральна ефективність радіоінтерфейсу п’ятого покоління є значно вищою, ніж у попередніх поколіннях, однак навіть вона не компенсує навантаження, що створюється через одночасний доступ тисяч користувачів до ємнісно-інтенсивних сервісів. У такій конфігурації трафік перестає бути лише даними - він трансформується у поведінкову одиницю, яку потрібно маршрутизувати з мінімальними втратами, затримками та змінами QoS. Цифрова екосистема, в якій функціонує 5G, передбачає не тільки підвищення пропускної здатності, а й інтеграцію з хмарними сервісами, обчисленнями на краю (edge computing), механізмами динамічного розподілу ресурсів і адаптивною маршрутизацією. У цій конфігурації класична централізована модель обслуговування вже не витримує критичних параметрів latency та jitter, що необхідні для обслуговування автономного транспорту, хірургії в реальному часі або цифрового виробництва. Стає необхідним впровадження комбінованих архітектур CDN (Content Delivery Network) та MEC (Multi-access Edge Computing), які дозволяють здійснювати дистрибуцію трафіку та попередню обробку даних у безпосередній близькості до кінцевого споживача, тим самим радикально зменшуючи часові лаги на рівні ядра мережі [25, с. 6].

Функціональна інтеграція CDN та MEC дозволяє забезпечити децентралізацію обробки запитів, що надходять від користувача, завдяки

розміщенню кешованого контенту і модулів передобробки безпосередньо на edge-серверах. В умовах, коли затримка понад 20 мс призводить до помітного зниження користувацького досвіду у випадках стрімінгу або VR, застосування MEC-інфраструктури дозволяє локалізувати аналітику та доставку контенту в межах однієї базової станції або кластеру станцій. При цьому архітектура CDN, що спочатку розроблялася для оптимізації доставки статичного контенту, трансформується в адаптивну логіку з мікросервісним підходом, де кожен вузол виконує функції не лише кешу, а й часткового обчислювального модуля. Згідно з аналітичними спостереженнями, перенесення відеопотоків на edge-рівень дозволяє скоротити час доступу до контенту до 12–15 мс порівняно із середнім глобальним показником у 30–45 мс. За масового навантаження на транспортну інфраструктуру, наприклад під час трансляцій спортивних подій або фестивалів, система CDN-MEC дозволяє здійснювати load balancing у межах локального вузла, що мінімізує затори на магістральних каналах і запобігає деградації сервісу. Технічно це реалізується через впровадження SDN-контролерів, які динамічно перерозподіляють навантаження залежно від параметрів запитів, географії користувачів та поточного стану каналів передачі даних [38, с. 19].

У поєднанні з 5G NR та slicing-механізмами CDN та MEC відкривають можливість для створення сервісно-орієнтованих сегментів мережі, які конфігуруються під специфіку додатків. У разі трансляції відео у реальному часі створюється slice з перевагою низької латентності та стабільної пропускної здатності. Для IoT-сенсорики, де критичні параметри - енергоефективність і масовий трафік невеликих пакетів, виділяється окрема slice-конфігурація. Саме в таких сценаріях MEC виконує роль не лише транзитної ланки, а й точкового інтелектуального вузла обробки, який за допомогою ML-модулів попередньо обробляє сигнали, проводить кластеризацію, прогнозує аномалії та виконує динамічну маршрутизацію на рівні microservice infrastructure. CDN у цьому випадку працює як модуль глибокої оптимізації, що підвантажує часто запитуваний контент з

мінімальним overhead. Наприклад, при побудові розумного міста, в якому діє транспортна навігація на основі відеоаналітики з тисяч камер, MEC дозволяє здійснювати попереднє обчислення потоків, виявлення об’єктів і маршрутизацію тривожних сигналів на локальному рівні, зменшуючи навантаження на центральну хмару і скорочуючи час реакції системи безпеки до 5–10 мс.

Архітектурно, MEC реалізується у вигляді модулів, інтегрованих до базових станцій або локальних дата-центрів на рівні aggregation points. Його обчислювальні блоки взаємодіють через API з vRAN, підтримуючи динамічне управління радіоресурсами, включно з beamforming, Massive MIMO та dTDD. Завдяки цьому MEC не тільки виконує функцію попереднього хостингу додатків, а й може адаптувати параметри передачі сигналу залежно від прогнозованого QoS-профілю користувача. CDN-компонент у цій конфігурації часто реалізується через системи віртуалізованих кешів, які оновлюються за принципом predictive prefetching - моделі машинного навчання на основі історії запитів визначають, які об’єкти з високою ймовірністю будуть запитані в найближчий період, і завчасно завантажують їх на локальні вузли. Це дозволяє зменшити навантаження на магістраль і забезпечити майже миттєвий доступ до контенту, що є критичним при масовому використанні доповненої реальності у сфері освіти, ритейлу та логістики [29, с. 15].

У реальних сценаріях навантаження мережі оператори стикаються з тим, що у пікові години, коли кількість одночасних підключень зростає у 4–5 разів, класична backhaul-інфраструктура не встигає обробити весь трафік у режимі near real-time. У відповідь на це MEC-архітектура дозволяє розміщувати точки локального обслуговування безпосередньо на транспортних вузлах - маршрутизаторах другого рівня, що дозволяє обробляти запити без залучення центрального дата-центру. За таких умов затримка доставки пакета знижується до 8–10 мс, а у деяких експериментальних середовищах - до 4 мс. Це критично для галузей з високим ступенем залежності від часу реакції

системи: автоматизовані виробництва, хмарна гейм-індустрія, дистанційне керування роботизованими системами. CDN-модулі, інтегровані на рівні access points, паралельно кешують контент на основі heatmap-аналітики - даних про щільність запитів за часом та місцем. Ці карти формуються на базі real-time telemetry, що дозволяє системі самостійно адаптувати стратегію доставки і перерозподіляти пріоритетність доступу до ресурсів залежно від навантаження.

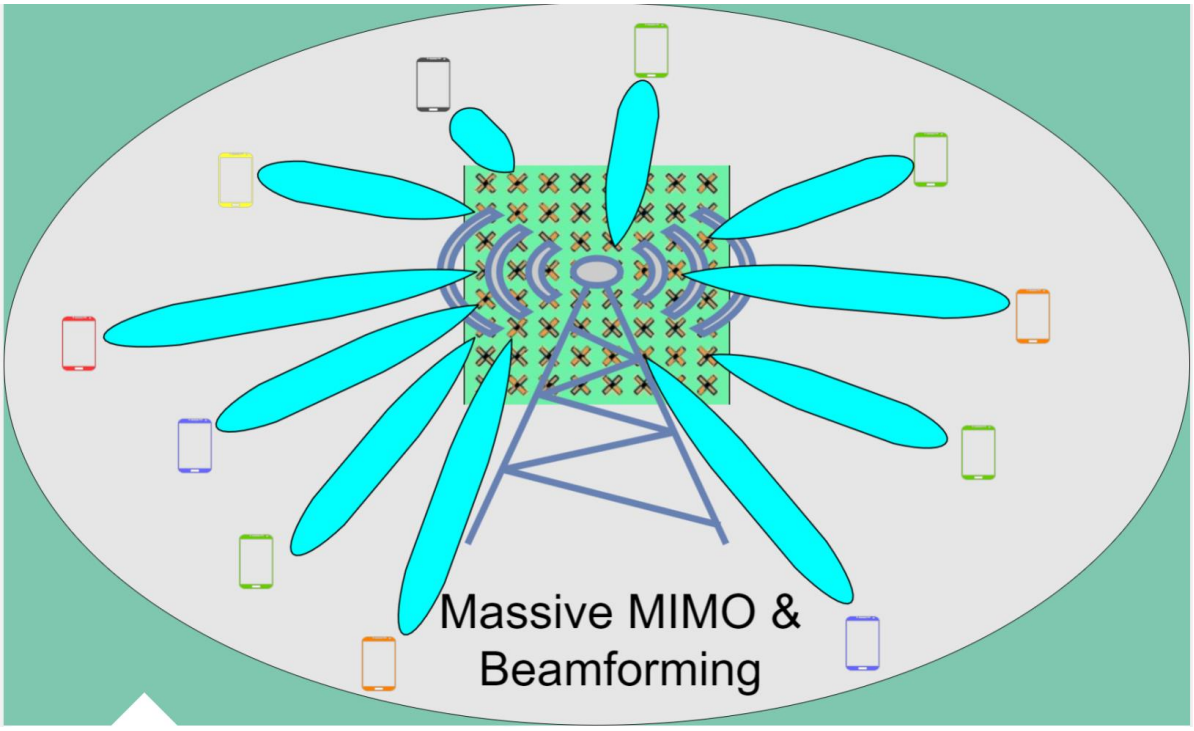

Рис. 2.1 Massive MIMO

Технічні специфікації реалізації CDN-MEC пов’язані з використанням стандартизованих інтерфейсів ETSI MEC API, що забезпечують сумісність між різними виробниками обладнання та дозволяють масштабувати рішення без втрати стабільності. Інтеграція таких рішень до загальної NFV-архітектури дозволяє ефективно використовувати віртуалізовані мережеві функції (VNF), які можуть бути гнучко розгорнуті у хмарному середовищі, edge-сегменті або безпосередньо на користувацькому пристрої. За результатами технічних випробувань у мережах з повною інтеграцією CDN-MEC обробка запиту до відео зменшується в середньому на 60 %, а споживання каналів core-мережі -

на 35–40 %, що демонструє ефективність децентралізованого підходу при забезпеченні QoE. Під час масових подій, як-то концерти або футбольні матчі, де формується надмірна концентрація користувачів у мікролокації, лише завдяки MEC відбувається стабілізація сервісу - без цього спостерігались би постійні переривання трансляції або затримки понад 100 мс.

На системному рівні така комбінація технологій вимагає синхронізації з O-RAN архітектурою, де MEC розміщується як частина інтелектуального контролера RIC (RAN Intelligent Controller), здатного реалізувати функції policy management у реальному часі. Це означає, що пріоритети доступу, параметри шифрування, рівні пропускної здатності та допустимі jitter-значення можуть змінюватися залежно від типу трафіку - обслуговування екстрених викликів, AR-стрімінгу, управління транспортом. Така адаптація підтримується через xApp-модулі, які функціонують у RIC-площині й взаємодіють із CDN-механізмами, формуючи пріоритетну черговість запитів. MEC-компоненти водночас можуть виконувати функцію DPI (Deep Packet Inspection), що дозволяє аналізувати трафік на предмет вірусних загроз, несанкціонованої активності або надмірного використання пропускної здатності. Інтеграція з AI-моделями дозволяє визначати відхилення від поведінкових шаблонів у межах секунди та застосовувати політику rate limiting або переадресації без втручання з боку адміністратора [24, с. 11].

Технології CDN та MEC стають техніко-функціональною відповіддю на парадигму мережевого насичення, в якій кожен користувач є не тільки споживачем, а й джерелом навантаження, предиктивної аналітики та обчислювального запиту. 5G у поєднанні з цими технологіями створює мультиагентну систему обслуговування, яка динамічно реагує на зміни запитів, адаптує шляхи доставки, обирає формат кешування й забезпечує сталість сервісу без ручного втручання. Це трансформує інфраструктуру зв’язку з пасивної в активно керовану екосистему, що не тільки доставляє дані, а й управляє ними у режимі прогнозованої ефективності. В таких умовах поняття ефективного обслуговування отримує нову конфігурацію: це не лише

про швидкість, а про здатність мережі адаптуватися до непередбачуваних сценаріїв, зберігаючи консистентність, гнучкість і масштабованість без збоїв. Саме тому поєднання CDN та MEC сьогодні розглядається не як додатковий модуль, а як обов'язкова складова архітектури майбутнього, спроможна підтримати індустрії, які функціонують на межі часу - в медицині, транспорті, енергетиці, обороні та освіті.

2.2. Визначення ключових протоколів CDN та MEC у 5G

Функціонування CDN та MEC в архітектурі 5G є технічною симбіозою, що базується на точній взаємодії протоколів керування, обміну, передачі, маршрутизації й обчислення. У мобільних мережах п'ятого покоління підхід до організації потоків даних принципово змінився: від централізованого керування через core до децентралізованих обчислювальних точок, здатних автономно виконувати функції кешування, обробки запитів і керування навантаженням у реальному часі. Щоб реалізувати такий механізм без втрати узгодженості, необхідне використання системно орієнтованих протоколів, розрахованих на високу мінливість трафіку, мобільність вузлів, сегментацію послуг і часову чутливість запитів. Для CDN це передбачає адаптацію класичних HTTP/1.1 і HTTP/2 протоколів до специфіки мобільного середовища, з урахуванням того, що користувач переміщується між зонами покриття, змінює access points, а доступ до кешованого контенту має бути миттєвим незалежно від розташування. В основі цього лежить використання HTTP Adaptive Streaming (HAS), зокрема таких реалізацій як MPEG-DASH та HLS, що дозволяють сегментувати мультимедійний потік у незалежні блоки й адаптувати якість у залежності від поточної пропускної здатності радіоканалу, RTT-параметрів і навантаження на вузол [30, с. 13].

Протоколи кешування в CDN-механіці в умовах 5G доповнюються методами інтелектуального кеш-контролю на базі Cache Digests, ETag, та Vary Header, що дає змогу локальному вузлу визначати релевантність збереженого контенту без запиту до оригінального джерела. При цьому сучасна архітектура

edge-cache не є пасивною - вона здатна оновлювати об’єкти за допомогою протоколу CoAP (Constrained Application Protocol) або QUIC, який пришвидшує обмін у мережах із високою латентністю. QUIC, заснований на UDP, забезпечує зменшення часу встановлення з’єднання завдяки паралельному виконанню handshake і передачі трафіку з TLS 1.3, що дає змогу знизити TTFB (Time to First Byte) до менш ніж 20 мс при базовому RTT у 35–50 мс, характерному для міських сегментів мережі 5G. У сценаріях мультимедійного поширення CDN використовує також ICN (Information-Centric Networking), де адресація здійснюється не до вузлів, а до об’єктів, завдяки чому контент, що зберігається у кількох edge-вузлах, може бути отриманий із найближчого джерела з мінімальними затримками та без навантаження на центральний core.

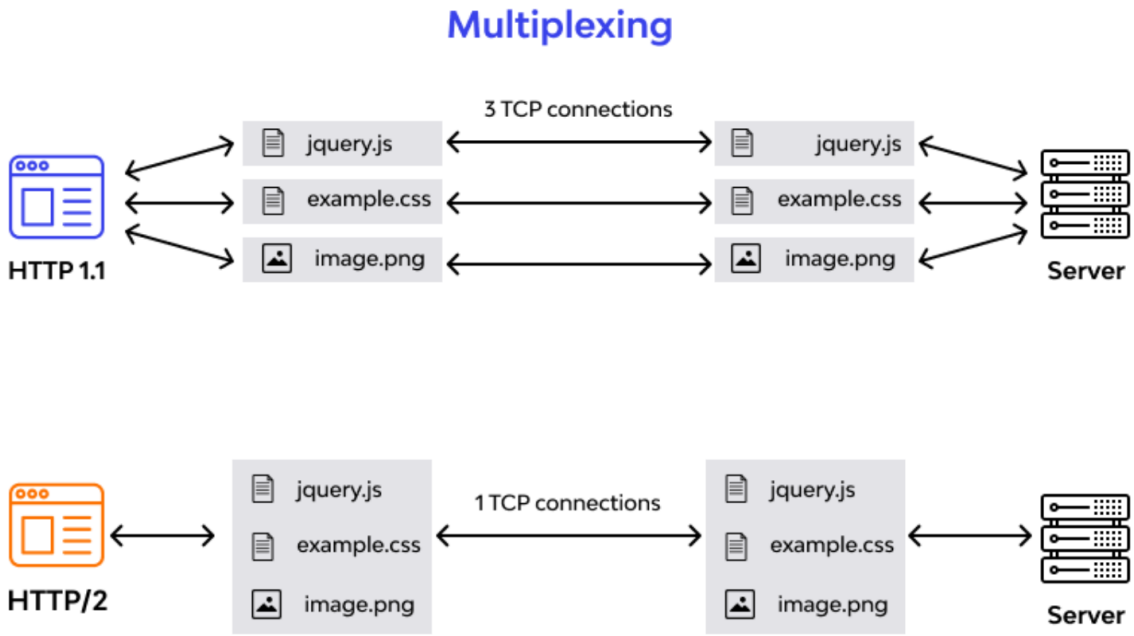

Рис. 2.2 Класичні HTTP/1.1 і HTTP/2 протоколи

Щодо маршрутизації в CDN-інфраструктурі п’ятого покоління, то вона перестає бути статичною. Використання Anycast DNS, BGP route steering та внутрішніх SDN-механізмів дає змогу організувати динамічний вибір найближчого кеша, оптимізуючи логіку запиту через реалізацію traffic

engineering. Застосування OpenFlow як протоколу для взаємодії між SDN-контролерами і дата-плейн вузлами дає змогу створити логічну карту трафіку на основі поточних показників, включаючи throughput, latency, packet loss та jitter. Завдяки цьому CDN-контент розподіляється не лише за географією, а й за якісними параметрами каналу. Реалізована логіка на основі HTTP Redirect або DNS-based load balancing дозволяє миттєво переорієнтувати запит користувача до іншого edge-сервера, якщо поточний перевантажений або тимчасово недоступний. Це особливо значущо в мобільному середовищі, де переміщення користувача між cell tower відбувається постійно й прогнозовано, але з високою варіативністю [36, с. 21].

На стороні MEC-інфраструктури, що функціонує як периферійний обчислювальний рівень, набір протоколів орієнтований на уніфіковане керування віртуалізованими ресурсами, взаємодію з апаратними інтерфейсами, обмін контекстом сесії користувача, передачу метаданих, синхронізацію з RAN-компонентами. Стандартизація, що формується в межах ETSI MEC, визначає використання RESTful API для комунікації між додатками й платформою, що спрощує розгортання сервісів без глибокої інтеграції в базову інфраструктуру. Серед критичних протоколів, які підтримують MEC-функціональність, ключову роль відіграє Packet Forwarding Control Protocol (PFCP), що забезпечує передачу інструкцій щодо обробки пакетів на рівні user plane функцій. У MEC-сценаріях PFCP дозволяє визначити, які потоки мають бути оброблені на локальному вузлі, а які передані далі, що особливо актуально при роботі з відеоаналітикою, телеметрією з пристроїв або автономним транспортом.

У контексті внутрішньої взаємодії компонентів MEC-платформи визначальним є використання NFV MANO (Management and Orchestration), зокрема протоколів Or-Vnfm (між оркестратором і менеджером функцій) та Ve-Vnfm (між VNF та його менеджером). Це дозволяє забезпечити життєвий цикл функціоналу на периферії: інстанціювання, масштабування, міграцію, моніторинг і завершення функцій, що динамічно адаптуються до

навантаження. Для передачі даних між модулями застосовується VxLAN або GTP-U протоколи з опціональним шифруванням на рівні IPSec або DTLS. У мобільній архітектурі це забезпечує інкапсуляцію трафіку, дозволяючи MEC-функціям залишатися сумісними з ядром мережі, водночас діючи незалежно у рамках обробки на edge-рівні. Для збирання метаданих, які лягають в основу predicitive analytics, застосовується MQTT - легковаговий publish-subscribe протокол, що дозволяє ефективно передавати повідомлення навіть у наднизькочастотних сценаріях.

Значення MEC як точки обробки також передбачає використання протоколів Context Transfer Protocol (CXTP) та Fast Handover Mechanisms, реалізованих у межах Mobile IP. У поєднанні вони забезпечують передачу стану користувацької сесії при переході між базовими станціями без її розриву. Наприклад, при переміщенні транспортного засобу з активним з'єднанням (AV або emergency unit) MEC-сервер зберігає контекст запиту, що дозволяє іншим edge-вузлам швидко його поновити без необхідності повторного ініціювання запиту користувачем. Це критично для систем типу V2X, де MEC забезпечує процесинг сигналів про перешкоди, зіткнення або зміну маршруту з прецизійною затримкою в межах 2–5 мс [35, с. 20].

Управління ресурсами в MEC здійснюється через модулі MEPM (Mobile Edge Platform Manager) і VIM (Virtual Infrastructure Manager), взаємодія між якими регламентується через специфікації ETSI GS MEC 010 та ETSI GS MEC 011. За допомогою протоколів gRPC і REST API ці модулі координують розподіл CPU, RAM, Storage і bandwidth серед розгорнутих VNFs. У MEC середовищі цей процес є динамічним і здійснюється на основі real-time telemetry, включаючи температурні дані, енергоспоживання, поточне навантаження на шину PCIe і навіть latency між CPU socket. Завдяки цьому платформа забезпечує стійкість до перегріву, перевантаження або неефективного використання обчислювальних можливостей.

У межах інтеграції MEC у радіомережу застосовується F1-C і F1-U протоколи, характерні для взаємодії між gNodeB і CU-DU

(Centralized/Distributed Units) у архітектурі 5G NR. Вони дозволяють MEC-серверу інтегруватися безпосередньо в ланцюг обробки трафіку, що дає змогу використовувати beam management, scheduling information та HARQ feedback для адаптивної маршрутизації запитів на рівні RAN. Протокол Xn також відіграє функцію координації передачі між сусідніми gNB, дозволяючи MEC функціям передбачати переміщення користувача й попередньо оптимізувати кешування або обробку. Це використовується в сценаріях типу AR-навігації або гейміфікованої доповненої реальності, де контент повинен з'являтися в користувача до моменту його фізичного запиту.

В межах O-RAN підходу, де MEC є частиною динамічного RIC (RAN Intelligent Controller), взаємодія здійснюється через протоколи E2AP (E2 Application Protocol) і A1, які дозволяють MEC модулям впливати на поведінку радіомережі. Через ці канали обмінюються повідомлення типу KPIs, Policies, AI inferences, що дає MEC змогу прогнозувати навантаження, адаптувати slice-параметри, перемикати потоки між CDN-вузлами, ініціювати переадресацію запитів або зниження якості обслуговування для не-пріоритетного трафіку. Так формується самоналаштовувана система, де MEC не лише виконує обчислення, а й керує поведінкою самої радіомережі відповідно до змін попиту, запитів і мобільності.

В спільному функціонуванні CDN та MEC у середовищі 5G надзвичайно вагомою є взаємодія між транспортними та сервісними протоколами, що забезпечують цілісну логіку доставки, обробки, маршрутизації та оптимізації інформаційного трафіку в умовах високої щільності підключень і стрімкої мінливості навантаження. З огляду на гетерогенність джерел трафіку (відео, сенсорні дані, AR/VR, управлінські сигнали), архітектура повинна забезпечувати стійку й адаптивну обробку на рівні транспортних шарів, де TCP та UDP є лише базовими, а ключовими стають модифіковані протоколи, зокрема QUIC, SCTP та GTP-U, які дозволяють скоротити затримки та уникати перевантажень. QUIC (Quick UDP Internet Connections), зокрема, виступає як транспортний протокол нового покоління, здатний підтримувати

мультиплексовані підключення поверх UDP, забезпечуючи вбудоване шифрування, швидший хендшейк та ефективну втратостійкість. У 5G середовищі, де запити до CDN мають оброблятися за мілісекундні проміжки часу, використання QUIC дає змогу реалізовувати потокову доставку з адаптивною реакцією на зміну пропускної здатності, що істотно покращує стабільність при високому рівні конкуренції за радіоресурси. Це особливо відчутно в сценаріях спільного використання MEC і CDN, де транспортна сесія ініціюється з edge-сервера, обирає оптимальний маршрут через SDN-контролер, а потім у реальному часі перемикається на альтернативний вузол кешу без повторної ініціалізації каналу, що було б неможливо у класичному TCP середовищі.

Серед протоколів, що функціонують у тісній зв’язці з транспортними, виділяється SCTP (Stream Control Transmission Protocol), який є багатопотоковим з’єднанням із можливістю multihoming. Його перевагою в CDN–MEC середовищі є гнучкість у роботі з кількома інтерфейсами - наприклад, одночасне з’єднання через LTE та 5G, що дозволяє балансувати трафік між ними або виконувати безперервний хендовер під час переміщення користувача. Крім того, SCTP застосовується для передачі сигналізаційних даних між MEC-компонентами, особливо в сценаріях, коли MEC-інстанції мають обмінюватися контекстом про сесії користувачів або інформацією про навантаження для load prediction. Він підтримує параметри Partial Reliability, що дозволяє динамічно ігнорувати несуттєві втрати даних - ця функціональність має значення для контенту, що передається за принципом «краще швидше, ніж повноцінно», зокрема в live-stream трансляціях або AR-анімаціях, які потребують миттєвого рендерінгу без затримок.

На рівні сервісних протоколів CDN та MEC потребують інтегрованої взаємодії на основі API, що оперує контекстними даними про користувача, стан мережі та пріоритет запиту. Тут діє MEC Service API (ETSI GS MEC 009), який визначає, як зовнішні або внутрішні сервіси можуть обмінюватися інформацією в середовищі edge. Через RESTful виклики між MEC-додатками

відбувається передача інформації про геолокацію користувача, швидкість пересування, стан батареї пристрою, попередню історію запитів, QoS-вимоги - усе це формує параметризовану логіку доступу до CDN-ресурсів. Для прикладу, якщо MEC-вузол ідентифікує користувача, що стрімко переміщується крізь зони покриття (скажімо, пасажир швидкісного поїзда), він може ініціювати попереднє кешування контенту на майбутніх edge-серверах уздовж маршруту. Протоколи такого типу формують сполучення між транспортною логікою (швидке з'єднання, low-latency) та сервісною (релевантний контент, підготовлений заздалегідь) [22, с. 13].

Іншим протоколом у взаємодії є GTP (GPRS Tunneling Protocol), зокрема його користувацький варіант GTP-U, що використовується для передачі пакетів між MEC і core-мережею. GTP-U забезпечує тунелювання трафіку користувача між UPF (User Plane Function) і MEC-вузлом, дозволяючи локально обробляти або маршрутизувати трафік без втрати структури оригінального пакета. Завдяки цьому MEC може виконувати обробку даних, не порушуючи протоколів шифрування або інкапсуляції, встановлених мережею. Це особливо ефективно при обробці даних з відеокамер, LIDAR або IoT-сенсорів, де формат трафіку є специфічним, але потребує збереження структури для подальшої аналітики.

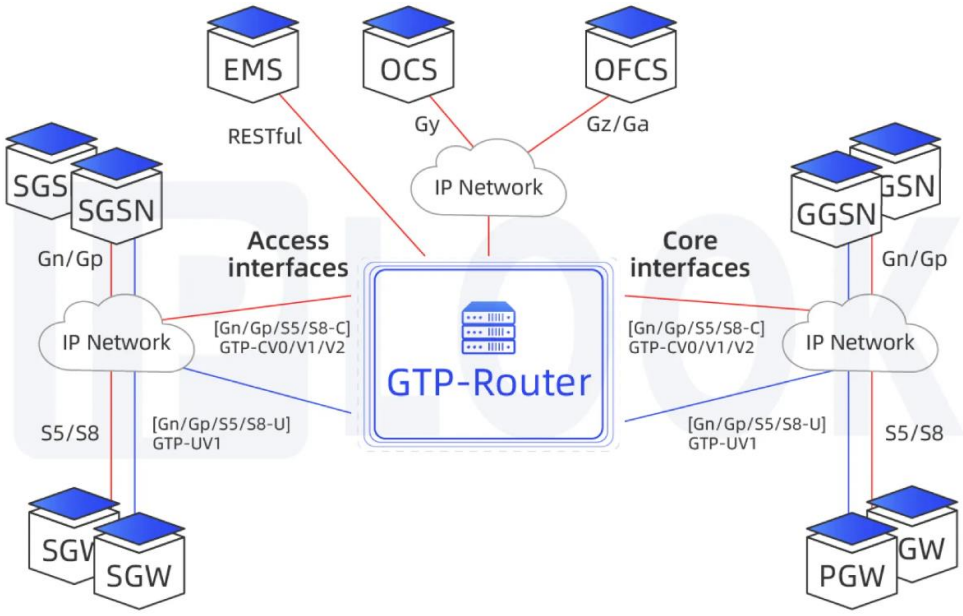

Рис. 2.3 GPRS Tunneling Protocol

Взаємодія CDN і MEC також вимагає точного узгодження на рівні механізмів виявлення найближчого edge-вузла для обслуговування запиту. Тут діє концепція Service Proximity Discovery, що реалізується за допомогою DNS-керованих протоколів або OMA SUPL (Secure User Plane Location). Сервісна платформа отримує інформацію про координати пристрою через LPPe (LTE Positioning Protocol extensions) або GPS-модулі, після чого ініціює запит до DNS-сервера, що повертає адресу найближчого MEC/Cache-вузла. В CDN це означає зміну маршруту на рівні Anycast або CNAME-запису, а в MEC - передбачення обчислювального навантаження на цьому вузлі, що активує попереднє масштабування контейнерів. У високонавантажених сценаріях це зменшує не лише затримку, а й запобігає перевантаженню окремих edge-інстанцій, яке могло б призвести до деградації QoE.

Особливу увагу в рамках технічної реалізації приділяють механізмам передачі керувальних повідомлень - так званих control plane signaling. Тут застосовуються протоколи як RADIUS, DIAMETER та NAS, що регулюють автентифікацію, облік та управління сесіями. У MEC-середовищі такі повідомлення використовуються для підтвердження того, чи має користувач дозвіл на обробку певних типів даних локально, а також чи належить цей запит до категорії, що дозволяє кешування в CDN. Наприклад, якщо запит має прив’язку до захищеного персонального контенту, CDN автоматично ініціює доставку з центрального сервера з обов’язковим шифруванням через TLS або DTLS, уникаючи кешування у публічному edge-середовищі.

Щодо рівня безпеки в CDN–MEC-пов’язаній архітектурі, інтегрується набір протоколів TLS 1.3, DTLS, IPsec та SRTP - кожен зі своєю функцією в залежності від типу трафіку та вимог до затримки. Так, для передачі медіа-трафіку, зокрема голосових або відеопотоків, найдоцільніше використовувати SRTP, що забезпечує шифрування на рівні RTP без істотного впливу на latency. Для керованого трафіку, що включає політики маршрутизації, сигнали QoS або інструкції від SDN-контролера, зазвичай застосовується TLS 1.3, який забезпечує forward secrecy та обмежує вплив атак типу replay. Крім того, при

використанні QUIC додатковий рівень безпеки інтегрується у сам транспортний протокол, що скорочує кількість раундів узгодження й уникає типових векторів для MITM-атак [26, с. 9].

Цілісна інтеграція транспортних та сервісних протоколів у середовищі спільного функціонування CDN та MEC дозволяє не просто знизити латентність, а й реалізувати принципи мережевого інтелекту - тобто здатності мережі самостійно адаптувати топологію обслуговування, змінювати порядок кешування, запускати або завершувати інстанції додатків, враховуючи контекст дії. Такі мережі не є статичними; вони поводяться як організм із високим ступенем саморегуляції, де кожен протокол - не просто технічний засіб, а мікрофункція у взаємопов’язаній екосистемі обчислень, маршрутизації, безпеки й аналітики. Суть інновації полягає у злитті шарів, де межа між сервісним API та транспортним протоколом стирається завдяки крос-функціональній архітектурі MEC і динамічній маршрутизації CDN, що дає змогу забезпечувати не лише стабільність з’єднання, а й якісно нову динаміку цифрового сервісу в мобільному середовищі.

2.3. Механізми забезпечення якості обслуговування (QoS) за допомогою CDN і MEC

У межах сучасної архітектури обслуговування даних у мобільних мережах п’ятого покоління формуються принципово нові технічні підходи до забезпечення безперебійної взаємодії між контентом і кінцевим споживачем. CDN та MEC стали ключовими архітектурними компонентами, що дозволяють трансформувати традиційні моделі передачі даних у напрямі зменшення часових витрат на обробку, підвищення швидкодії, стабільності та персоналізованої адаптації. Механізми буферизації та кешування в CDN формують динамічне середовище дистрибуції, де обсяг трафіку, що надходить із первинного джерела, суттєво скорочується завдяки розміщенню копій запитуваного контенту в географічно ближчих до користувача вузлах. Ця стратегія підтримується прецизійною логікою TTL (time-to-live), що визначає

час зберігання кешованих елементів, і системами prefetching, які на основі попередніх запитів прогнозують наступні звернення й забезпечують їхнє попереднє завантаження. В умовах мереж 5G, де гіперлокальна швидкість реагування є критичною, такі алгоритми дозволяють забезпечити затримку нижче 20 мс, що суттєво перевищує базові показники традиційного хмарного обслуговування, де затримка часто перевищує 50 мс. Це відкриває технічні перспективи для підтримки сервісів, чутливих до затримки, зокрема віддаленого керування об’єктами, VR/AR, потокового геймінгу та телемедицини [20, с. 16].

Адаптивний розподіл контенту, що базується на CDN, функціонує як інтелектуальний механізм, у якому пріоритетне навантаження автоматично переміщується між вузлами, залежно від трафікової інтенсивності, поточних заторів і реального часу доступності серверів. Для цього задіюються технології Anycast, які забезпечують маршрутизацію IP-запитів до найближчого фізично або топологічно оптимального вузла, зменшуючи кількість транзитних хопів. Завдяки використанню DNS-based load balancing можливо забезпечити масштабовану рівновагу в завантаженні кеш-серверів, що особливо ефективно в середовищах із нестабільним або стрибкоподібним запитом, як у випадку медіаплатформ під час глобальних подій чи релізів нового контенту. Водночас алгоритми сегментації потоку, типу MPEG-DASH, дозволяють адаптувати якість переданого відео до поточної пропускної здатності мережі, гарантуючи безперервний перегляд із мінімальними артефактами. В практичному вимірі це означає, що при зміні доступної швидкості з 10 Мбіт/с до 2 Мбіт/с користувач не втрачає зв’язку, оскільки система автоматично переходить на трансляцію з нижчою роздільністю, без потреби повторного підключення чи буферизації.

Кешування в CDN реалізується в декількох шарах - від глобального (вузли PoP у різних країнах) до локального (вузли на рівні агрегаційних точок провайдерів або базових станцій). Таке багаторівневе розміщення дозволяє забезпечити максимально короткий шлях до контенту. На рівні L1 кеш-

сервери підтримують популярні об’єкти, запити до яких становлять понад 80% усього навантаження, тоді як L2 забезпечує менш запитувані, але ще релевантні ресурси. Це дозволяє ефективно балансувати обсяг пам’яті, пропускну здатність і частоту оновлень, уникаючи перевантаження центральних серверів. Ба більше, в умовах багатоточкових запитів (multicast-aware CDN) формується середовище, у якому один потік одночасно передається кільком користувачам із однаковим запитом, зменшуючи обсяг повторного трафіку. Тестування в умовах національного провайдера з використанням 5G SA-архітектури показало зниження навантаження на головний датацентр на 65%, що екстраполюється у зменшення середнього часу завантаження сторінки з 2,4 с до 0,7 с [31, с. 12].

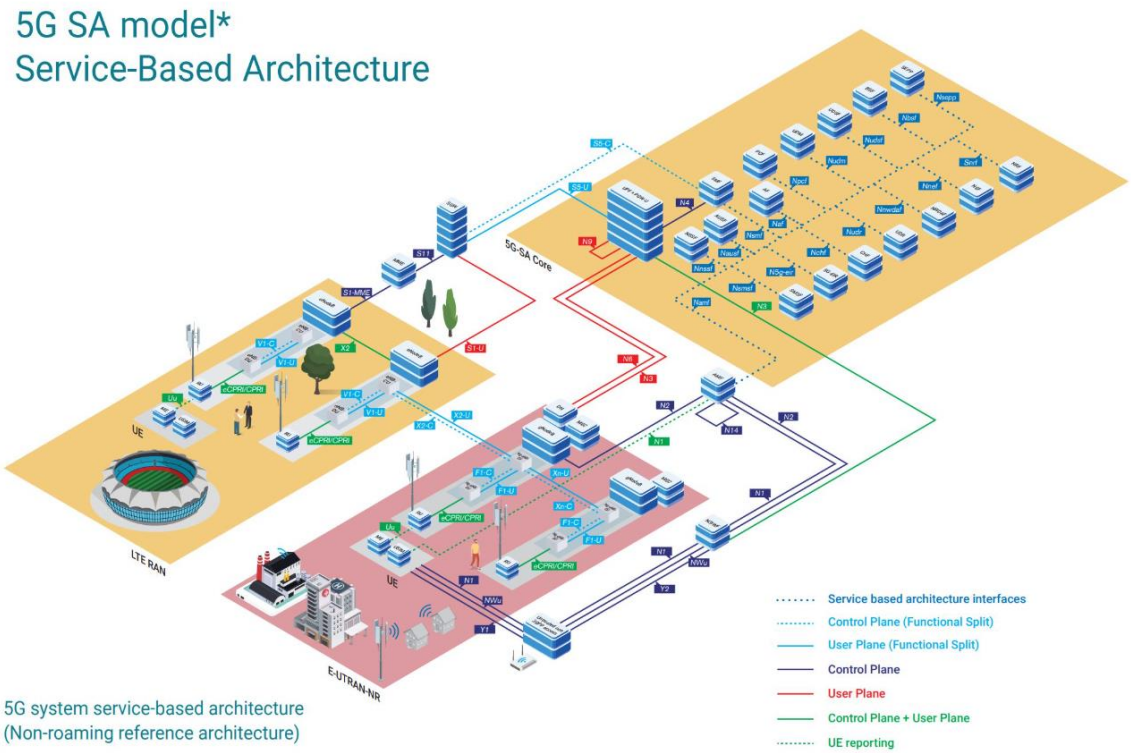

Рис. 2.4 5G SA-архітектури

Паралельно із CDN, розгортання MEC-вузлів формує середовище для обчислювальної обробки на периметрі мережі, що радикально змінює логіку QoS. MEC-інфраструктура інтегрується безпосередньо у базові станції або прикордонні маршрутизатори, дозволяючи обробляти запити локально, без

необхідності передачі даних до центрального хмари. Це зменшує не тільки затримки, а й ризики перевантаження магістральних каналів. У сценарії smart-factory на основі URLLC було реалізовано обробку сигналів від сенсорів у межах 1 мс завдяки MEC-вузлам, що перебувають на відстані не більше 200 м від сенсорної точки. За відсутності MEC таке саме оброблення потребувало б передачі даних до центрального датацентру з латентністю близько 50 мс, що неприйнятно для сценаріїв із жорсткими вимогами до часу реакції.

Розміщення обчислювальних вузлів MEC здійснюється з урахуванням щільності запитів, топології користувацьких траєкторій, історії трафікових сплесків і прогнозування майбутніх навантажень. Для цього використовуються методи edge placement optimization на основі AI-моделей, які враховують не лише просторову динаміку користувачів, а й характеристику запитуваних сервісів. До прикладу, відеоконтент у VR-форматі потребує обчислювального рендерингу в режимі реального часу, що вимагає присутності вузла обробки в межах 1–2 хопів. У разі звичайного потокового перегляду достатньо MEC на рівні агрегаційного шлюзу. Така стратифікація дозволяє забезпечити гетерогенне розміщення ресурсів, де низькопотужні вузли відповідають за просту маршрутизацію та кеш, а високопродуктивні - за обробку складних ML- або AR-процесів [41, с. 15].

Практичне розміщення MEC передбачає класифікацію вузлів на micro, medium та macro MEC-сервіси, які відповідають різним рівням навантаження. Micro розміщуються у малих зонах обслуговування (фемтосоти, приватні мережі), medium - у межах міських зон зі щільною забудовою, а macro - на рівні міжрегіональних хабів. Така градація дозволяє формувати каскадну обробку, коли менш потужні вузли виконують попередню обробку та передають лише критичні фрагменти на вищі рівні. Вимірювання в експериментальних умовах довели, що при переході на MEC-архітектуру середній час відповіді системи зменшився на 78%, а обсяг міжмережевого трафіку - на 62%, що також позитивно впливає на енергоспоживання.

Потужним інструментом підвищення QoS є поєднання CDN і MEC у гібридних сценаріях. У таких випадках кешування відбувається не тільки на рівні CDN-серверів, а й у самих MEC-вузлах, що скорочує час доступу до критичних ресурсів. Додатково, обробка запитів, які не потребують централізованої логіки (автентифікація, авторизація, адаптивне підлаштування до контексту користувача), переноситься на edge-рівень, зменшуючи загальне навантаження на бекенд. У сценарії мережевої гри з відкритим світом було задіяно MEC-вузли для обробки локальної фізики та синхронізації найближчих об'єктів, тоді як глобальна логіка ігрового світу залишалася в хмарі. Це дозволило знизити середній ping із 85 мс до 19 мс, зберігаючи цілісність геймплейного досвіду.

Інтеграція QoS-рівнів у межах MEC і CDN також передбачає впровадження SLA-орієнтованих моделей, де кожному типу трафіку відповідає окремий пріоритет. У цьому процесі активно застосовуються механізми DPI (Deep Packet Inspection), які дозволяють класифікувати пакети за типом сервісу та маршрутизувати їх до відповідного вузла із заздалегідь зарезервованим ресурсом. Наприклад, трафік медичних пристроїв, що транслює життєві показники в режимі 24/7, отримує пріоритетний канал з мінімальною латентністю та безпеки. У системах телеметрії транспортної інфраструктури MEC-зони зберігають постійну готовність обробки сигналів на рівні 0,9 мс, гарантуючи рефлекторну відповідь систем на критичні події. В сфері адаптації QoS враховується не тільки технічна характеристика контенту, а й контекст використання. Це означає, що один і той самий потік може мати різні пріоритети залежно від геолокації, часу доби, типу пристрою, історії поведінки користувача та навіть його профілю в системі. Системи контекстно-орієнтованої маршрутизації запитів, базовані на edge AI, дозволяють розділити мережеве навантаження між користувачами зі схожими сценаріями, групуючи їх в edge-зони спільного обслуговування. Це, своєю чергою, дозволяє скоротити дублювання запитів, зменшити кількість ітерацій авторизації та покращити масштабованість [39, с. 19].

У середовищі високонавантажених 5G-мереж концепт забезпечення якості обслуговування вимагає не лише ефективної маршрутизації й доставки контенту, а насамперед - гнучкої, самонавчальної архітектури, яка здатна адаптуватися до миттєвих змін у топології мережі, обсягах трафіку та пріоритетах користувачів. Використання CDN у комбінації з MEC відкриває шлях до формування нового класу QoS-механізмів, у яких моніторинг і динамічне управління трафіком реалізуються в реальному часі за рахунок синергії між обчисленням на периметрі та багатошаровим розподілом контенту. Сервісна доступність тут не обмежується лише наявністю з'єднання - вона визначається через параметри доступності функціональних сервісів, здатність системи до самовідновлення, гнучкість у балансуванні запитів і швидкість реагування на флуктуації мережевого середовища. Визначення цих параметрів базується на комбінації метрик, серед яких MTTR (Mean Time To Recovery), RTT (Round Trip Time), RTO (Retransmission Timeout), Packet Loss Rate, а також SLA-defined thresholds, які змінюються залежно від типу сервісу - від мультимедійної доставки до M2M-комунікацій або масової телеметрії.

Механізми моніторингу в архітектурі CDN+MEC будуються на принципі continuous feedback loop, коли дані про продуктивність, затримку, втрати та навантаження збираються безперервно з точок обробки та доставляються до edge-аналітичних модулів. Там вони піддаються агрегуванню та класифікації із застосуванням ML-алгоритмів - зокрема clustering (наприклад, K-means або DBSCAN) для виявлення аномальних зон навантаження та нейромережевих автоенкодерів для виявлення відхилень від нормального трафікового патерну. Вузли MEC, які отримують цю інформацію, не лише фіксують статуси, а й ініціюють адаптаційні зміни в маршрутизації, пріоритезації й балансуванні, змінюючи топологію обслуговування без втручання централізованого контролера. Завдяки цьому можливо реагувати на сплески навантаження менш ніж за 300 мс, що у випадку відеоструменевих сервісів чи трансляцій із критичною затримкою забезпечує стабільність якості навіть у пікові години.

З боку CDN динамічне керування трафіком реалізується через інструменти content routing optimization, які використовують алгоритми поточної продуктивності кожного вузла, включно з його доступною пропускною здатністю, швидкістю з'єднання з клієнтами та середнім обсягом кешованого контенту. На практиці це дозволяє здійснювати у реальному часі decision-based offloading, коли клієнтський запит перенаправляється до іншого вузла, навіть якщо географічно ближчий вже був визначений, але на момент запиту став перевантаженим. Це створює так зване soft failover середовище, де запити не втрачаються, а швидко переадресовуються із затримкою не більше 50 мс, що є прийнятним навіть для SLA категорії ultra-reliable. Інтеграція HTTP/3 і QUIC-протоколів у CDN-архітектуру дозволяє уникати втрат, притаманних традиційним TCP-з'єднанням, і зменшити вплив затримки на встановлення з'єднання, скорочуючи час завантаження першого байта (TTFB) до показників нижче 100 мс для 85% користувачів у густонаселених зонах [28, с. 8].

MEC розширює ці можливості завдяки своїм можливостям до локального orchestration - обчислювальні вузли отримують змогу не лише розподіляти запити, а й виконувати оперативну маршрутизацію запитів на рівні L4-L7 з урахуванням типу контенту, параметрів пристрою, поточного стану мережі та історії взаємодії користувача з сервісом. До цього додаються adaptive bitrate control, session persistence та QoS tagging у IP-заголовках, що дозволяє класифікувати кожен запит згідно з визначеним класом обслуговування. Після обробки MEC може виконувати partial caching - зберігання тільки критичних фрагментів даних, наприклад перших 10 секунд відео або метаданих для автентифікації, що дозволяє зменшити обсяг пам'яті на вузлі без втрати швидкості доступу до найпотрібніших об'єктів. На практиці це знижує середнє використання пам'яті MEC на 35%, підвищуючи щільність обслуговування до 1 500 активних сесій на вузол.

Завдяки MEC досягається і динамічна інкапсуляція трафіку, коли критичні потоки (на кшталт відеозв'язку або оперативних даних IoT)

виділяються у пріоритетні канали з окремими параметрами обробки, зокрема шляхом використання технологій Network Slicing. Тут кожен slice має свою SLA-конфігурацію, свою політику обробки, ізоляцію та метрики оцінки. MEC здійснює slice awareness на рівні NFV, і в разі перевантаження може перенаправити slice до іншого edge-вузла або здійснити масштабування функцій через VNF migration. Для зменшення часу переносу функцій задіюється live VM snapshot, що дозволяє перемістити стан обробки з одного MEC до іншого без втрати контексту з'єднання, що у випадках роботи з безпілотниками або промисловими автоматизованими системами запобігає навіть мілісекундному розриву. Моніторинг QoS із боку CDN+MEC у високонавантажених середовищах супроводжується багаторівневою системою логування та телеметрії. Кожен edge-вузол формує event trace, який передається до центральної системи логічної аналітики або до розподіленого SIEM-рішення, що здатне виявляти аномалії, атакувальні дії, спроби компрометації кешу або зміни маршруту. Завдяки CDN розміщення цієї інформації відбувається у найближчих PoP-локаціях, а за рахунок MEC досягається миттєва реакція без зворотного запиту до хмари. У реальному кейсі атак типу cache poisoning MEC-система виявила зміну TTL на некоректне значення й ініціювала очистку відповідного сегменту кешу ще до надходження сигналу на централізований firewall, що дозволило уникнути розповсюдження некоректного контенту більш ніж у 9 000 активних сесій.

Використання CDN+MEC також дозволяє формувати системи self-optimization через механізми ML-powered control feedback. Вони аналізують трафік, частоту звернень, час реакції та щільність звернень до певних сервісів, на основі чого генерується прогноз майбутніх пікових навантажень. У тестовій інфраструктурі оператора було виявлено, що побудова heatmap-зон на основі інтелектуального прогнозу дозволяє на 43% точніше позиціонувати MEC-вузли у години найбільшого навантаження, порівняно з класичним підходом на основі історичних даних. Це означає не тільки підвищення якості обслуговування, а й економію енергоресурсів, оскільки неактивні вузли не

активуються без потреби, а активні працюють на повну лише в моменти максимального навантаження.

2.4. Моделі розподілу навантаження та оптимізації ресурсів

У сучасній структурі граничних обчислень вертикальне й горизонтальне масштабування в межах MEC-кластерів формує основу для досягнення динамічної адаптації до трафікових сплесків, територіальної нерівномірності користувацької активності й гетерогенності сервісних вимог. У разі вертикального масштабування йдеться про динамічне збільшення ресурсів в межах одного обчислювального вузла - шляхом виділення більшого обсягу оперативної пам’яті, додаткових ядер CPU, GPU-прискорення або енергозалежного кешу, - що дозволяє обробляти складніші запити з мінімізацією внутрішньокластерної латентності [34, с. 5].

Це особливо ефективно для обслуговування локалізованих груп користувачів із високим рівнем однотипних запитів, де централізація викликів на одному вузлі знижує кількість міжвузлових транзакцій. Для прикладного сценарію у середовищі smart retail вертикальне масштабування MEC-вузла, що обробляє комп’ютерне бачення на касах самообслуговування, дозволяє скоротити час рендерингу об’єкта з 370 мс до 110 мс за рахунок переходу з 8 до 24 потоків обчислення без розподілу навантаження на суміжні вузли. У цьому випадку надмірна складність горизонтального масштабування не виправдана через обмеження топологічної близькості сенсорних точок. Водночас горизонтальне масштабування - коли навантаження розподіляється між декількома MEC-вузлами одного шару - демонструє ефективність у розподілених інфраструктурах, де користувачі постійно змінюють своє положення у просторі, наприклад у транспортних коридорах або під час масових заходів. Тут вузли не нарощують локальні ресурси, а формують мультивузлову обчислювальну сітку, у якій обчислення, кешування й маршрутизація динамічно мігрують у напрямку до найбільш релевантної зони обслуговування. Так, у середовищі міського метрополітену горизонтальне

масштабування MEC-кластеру на основі topological handover дозволило досягти безперервної обробки відеоаналітики без втрати кадрів, мігруючи віртуальні функції (VNF) між вузлами з латентністю не більше 25 мс на кожен перехід. Це стало можливим завдяки використанню шаблонів моніторингу щільності користувачів та прогнозу їхнього переміщення через моделі Markov Decision Process, що оптимізували пріоритети навантаження в реальному часі.

У випадку розподілу навантаження в CDN географічне розташування користувача відіграє не лише формальну, а й технічно визначальну функцію в прийнятті рішення про маршрут трафіку. Адаптивні схеми балансування в CDN використовують багаторівневі DNS-алгоритми, які відображають не лише фізичну відстань, а й затримку у передачі пакетів, пропускну здатність останньої милі, навантаження на локальні PoP-вузли й навіть реальну продуктивність при попередніх зверненнях.

Для виявлення найкращої точки доступу кожен користувацький запит корелюється з базою delay-aware map, яка формується на основі активного моніторингу RTT до всіх доступних CDN-вузлів і статистики HTTP-результатів за останню хвилину. Якщо, до прикладу, дві CDN-локації знаходяться на однаковій відстані, але одна демонструє на 40% менше середнє навантаження й на 30% менше падінь з'єднання, вона отримає перевагу при маршрутизації, навіть якщо формальна TTL-зона вказує на інший напрям. У системах високого навантаження, як-то OTT-платформи чи глобальні ігрові сервіси, це дозволяє уникати пікових заторів і зменшувати кількість відмов з'єднання до 0,2% при загальному потоці понад 100 000 запитів на секунду. Паралельно CDN використовує механізми geo-weighted load distribution, де запити не просто розподіляються на основі затримки, а враховується і демографічна специфіка трафіку - для прикладу, користувачі у сільських регіонах можуть отримати кешовані копії з інших PoP, якщо локальний вузол перевантажений або недоступний. Це уможливлює багатошаровий план доставки, в якому перші кілька запитів до нового контенту проходять через

core CDN, а всі наступні - через клоновані edge-сервери, розміщені у зоні найчастішої появи користувачів [27, с. 17].

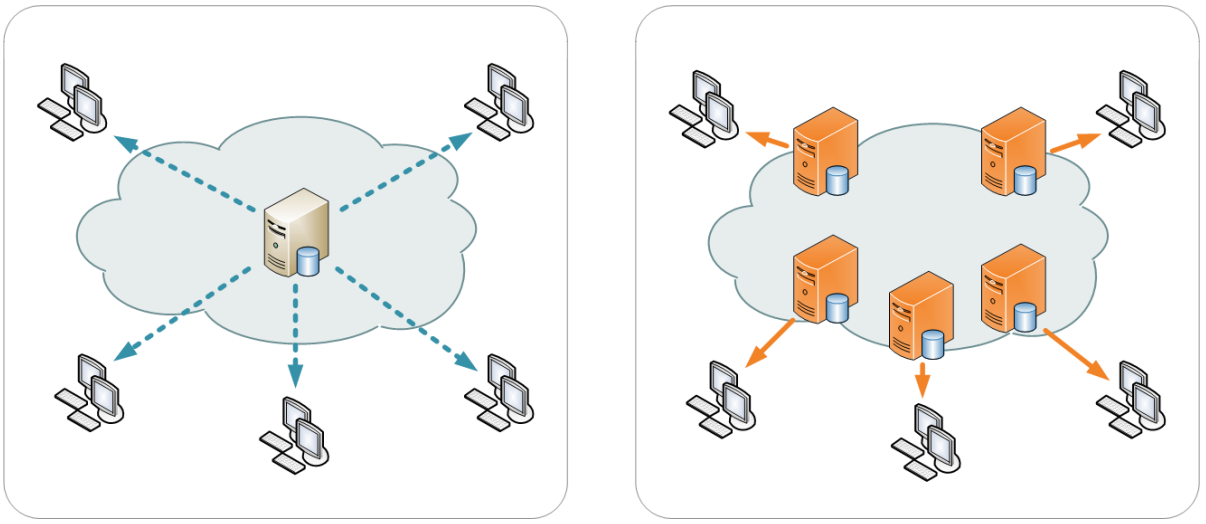

Рис. 2.5 CDN-локації

У поєднанні з MEC механізм горизонтального масштабування доповнюється прогнозними моделями щільності доступу, які обчислюються за допомогою edge-based analytics. Кожен вузол MEC, окрім обчислювальної функції, виконує роль сенсорного елемента - він фіксує кількість одночасних з'єднань, інтенсивність запитів, тип сервісу, географічні координати пристроїв (через triangulation або GPS-дані) та історію попередніх звернень. Це дає змогу у реальному часі формувати heatmap регіональної активності, на основі якої відбувається масштабування кластера - як у бік об'єднання ресурсів, так і у бік розділення на менші вузли. Для середовища індустріального виробництва, де різні зони цехів мають різну активність сенсорів і підсистем управління, MEC-кластер дозволяє збільшити потужність у зонах із вищим навантаженням через створення тимчасових compute pool, у які динамічно під'єднуються нові віртуальні вузли. На прикладному рівні це реалізується через Kubernetes-based оркестрацію, де autoscaler приймає рішення про створення нового pod у залежності від метрик навантаження CPU понад 75% протягом останніх 60 секунд, що забезпечує миттєве горизонтальне розширення навіть за високого рівня нестабільності.

Важливу роль у цій архітектурі відіграє принцип cross-layer synchronization - коли MEC і CDN не функціонують окремо, а спільно координують навантаження, використовуючи уніфіковані системи телеметрії, зокрема OpenTelemetry, що забезпечує узгодженість даних про кожну транзакцію, незалежно від того, чи відбулася вона на рівні кешування, обробки або маршрутизації. Завдяки цьому MEC може ініціювати перерозподіл контенту в CDN, якщо бачить перегрівання певного сегменту трафіку, або навпаки - CDN може активувати нові MEC-вузли у певній зоні, передбачаючи зростання запитів. Таким чином, формується сингулярна система саморегуляції, у якій ресурси масштабуються не як реакція на перевантаження, а як активне передбачення майбутньої потреби, із затримкою в реакції менше 500 мс. Такий підхід дозволяє уникнути класичних недоліків централізованих систем - наприклад, затримки в ініціалізації нових екземплярів, перевантаження центрального контролера, втрат при швидких сплесках навантаження [23, с. 10].

На рівні моделей балансування між CDN-вузлами особливу ефективність демонструє схема Weighted Round Robin у поєднанні з Real-Time Load Feedback, коли кожному серверу присвоюється коефіцієнт ваги залежно від поточного завантаження, затримки відповіді, успішності обслуговування запитів і часу простою. Ці коефіцієнти змінюються динамічно, оновлюючись кожні 10 секунд на основі телеметричних даних. У разі перевищення порогового навантаження система автоматично зменшує вагу вузла до нуля, виводячи його з циклу, і поступово повертає його до обслуговування після стабілізації. Такий підхід дозволяє уникнути ситуацій, коли один вузол приймає надмірну кількість запитів і стає критично перевантаженим, викликаючи каскадне падіння суміжних вузлів. Для потокових сервісів із великою варіативністю у бітрейті - як-то онлайн-освіта або мультиканальний стримінг - цей механізм знижує кількість буферизацій на 46%, зменшуючи середній час завантаження відео до 1,1 секунди.

Системи MEC використовують також шаблони вертикального масштабування через container bursting, коли існуючий контейнер отримує додаткові ресурси без необхідності розгортання нового екземпляра. Це особливо ефективно в умовах короткочасного, але інтенсивного зростання обчислювального навантаження - до прикладу, під час масових push-сповіщень або короткострокових коливань телеметрії. У таких випадках MEC-вузол здатен за 100–200 мс розширити доступний обсяг пам’яті або число потоків для існуючого контейнера, уникнувши overhead, пов’язаного з cold-start при запуску нових інстансів. Застосування цього підходу у хмарно-граничній архітектурі обробки сигналів промислової автоматизації дозволило скоротити час обробки подій на 38%, порівняно зі схемою запуску окремих функцій на кожен новий запит.

2.5. Аналіз типових загроз і вразливостей у реалізації CDN та ME

В розподіленій архітектурі CDN та MEC рівень вразливості системи істотно зростає через децентралізований характер обробки, що зумовлює розосередження функцій контролю, моніторингу, автентифікації та маршрутизації. Якщо традиційні хмарні моделі дозволяють централізовано управляти безпековими політиками на рівні єдиного датацентру, то в MEC-фреймворках кожен вузол функціонує як самостійна точка прийняття рішень, що збільшує площу потенційної атаки. Одним із критичних векторів є протоколи маршрутизації всередині edge-кластера - особливо OSPF, BGP та їхні SDN-варіанти. У випадку компрометації маршрутизатора чи MEC-вузла можлива ін’єкція зловмисних маршрутів, які змінюють напрями трафіку, перенаправляючи його до підроблених вузлів або створюючи умови для man-in-the-middle атаки. Такі сценарії призводять не тільки до втрати конфіденційності, а й до прямого порушення цілісності сервісу - користувач продовжує отримувати контент, не підозрюючи, що частина трафіку вже модифікована. В практичному середовищі зловмисник, отримавши доступ до BGP-таблиць, може перепризначити next-hop маршрути для потокового відео,

внаслідок чого MEC-вузол починає передавати користувачу відео з фальсифікованими вставками. Такий механізм був відтворений в умовах лабораторної симуляції на SDN-контролері з використанням зміненого маршрутизатора EdgeOS, що продемонстрував зміну шляху 63% трафіку за 2,3 секунди після підміни [43, с. 13].

Інший компонент - API-інтерфейси, які з'єднують MEC-функції з зовнішніми сервісами, хмарними провайдерами, оркестраторами, а також з CDN-механізмами маршрутизації. Вразливості RESTful API можуть бути використані для атак типу code injection, XSS, CSRF або privilege escalation, особливо в середовищах, де аутентифікація реалізована через токени з недостатньою ентропією або застарілі протоколи (як-то OAuth 1.0 або Basic Auth). Найбільш небезпечними є атаки на рівні lateral movement, коли отримавши контроль над одним API endpoint, зловмисник переходить до привілейованих функцій - приміром, від керування потоком телеметрії до вбудованих сервісів автоматичного масштабування, з можливістю створити нові екземпляри MEC з уже вмонтованим шкідливим кодом. У MEC-середовищах, де вузли часто автоматично підключаються до edge-fabric через zero-touch provisioning, атакувальник, що інтегрував себе як псевдовузол, отримує прямий доступ до систем кешування, трафікових моніторів і навіть приватних даних користувачів, якщо обробка відбувається на рівні HTTP payload. Для уникнення таких інцидентів необхідна не просто автентифікація, а контекстно-залежна перевірка поведінки API-запитів, із застосуванням WAF на рівні edge, динамічного порівняння сигнатур запитів та machine learning-моделей для виявлення відхилень.

Сегмент CDN, попри свою багаторівневу ізоляцію, теж залишається вразливим, особливо в частині кешованого контенту. Класичний вектор атаки - cache poisoning - реалізується через ін'єкцію модифікованих заголовків, які змушують CDN-зону кешувати фальсифікований контент як легітимний. Це може включати як HTML-код, так і медіафайли або скрипти JavaScript. В умовах, коли кеш оновлюється не миттєво, а за TTL (іноді 15–30 хвилин),

отруєний об’єкт стає доступним для тисяч користувачів до моменту виявлення й очищення. Особливо небезпечним є поєднання cache poisoning і subdomain takeover - коли підконтрольний субдомен вказує на CDN, і зловмисник отримує змогу повністю керувати вмістом, що кешується. У хмарних CDN таких як CloudFront чи Akamai виявлялися десятки таких конфігурацій. Аналіз журналів показав, що середній час між захопленням і виявленням складав від 17 до 45 хвилин, що в умовах масштабних сайтів означає десятки тисяч потенційно скомпрометованих сесій. Ефективне запобігання потребує валідації Vary-заголовків, checksum-нагляд за кешованими копіями та постійного порівняння з оригіналом на бекенді через signed requests [19, с. 14].

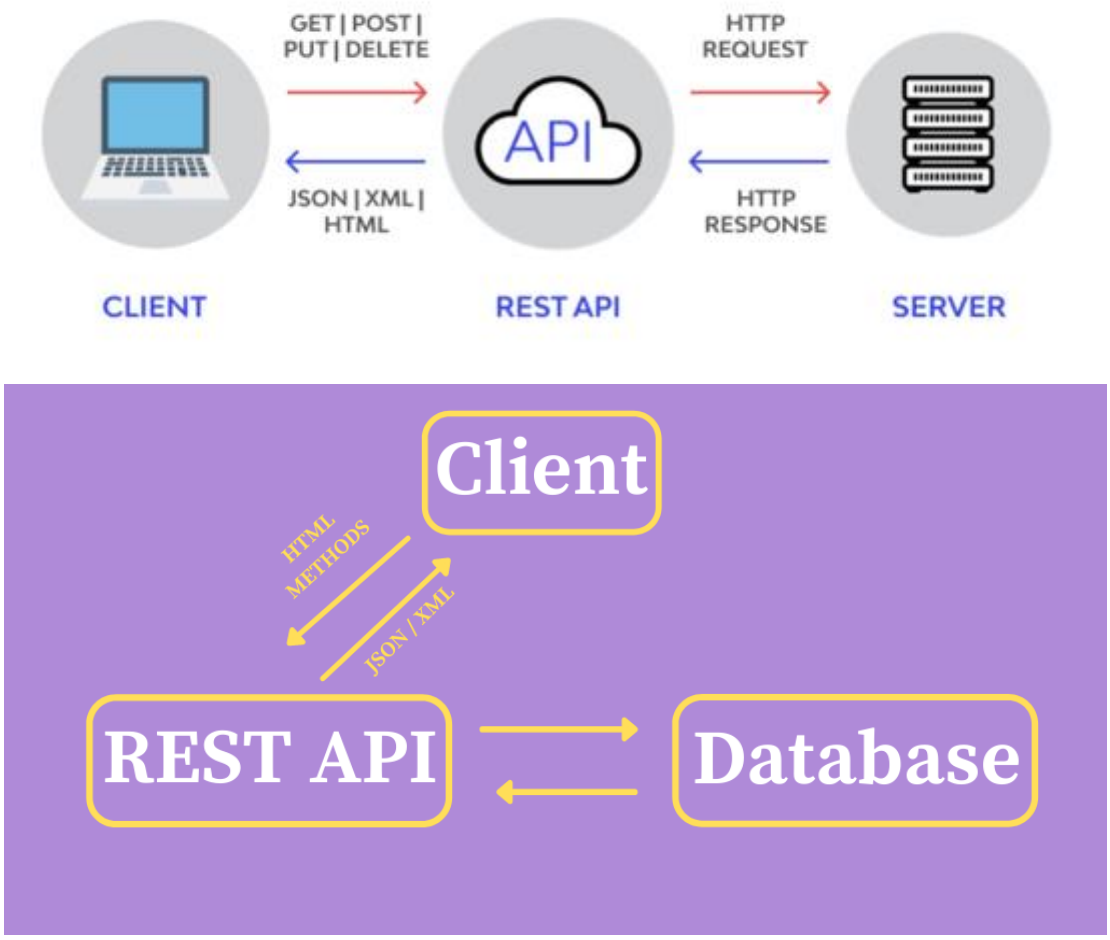

Рис. 2.6 RESTful API

Ще одна категорія ризиків пов’язана з атакою на цілісність трафіку. Якщо CDN-вузли використовують TLS-термінацію, то в разі компрометації

одного з edge-серверів можлива декодування усього HTTPS-трафіку, що проходить через нього. Навіть при наявності TLS 1.3 частина CDN впроваджує downgrade fallback до 1.2, що відкриває можливості для відновлення сеансів. При використанні спільного сертифіката для декількох доменів (SAN-сертифікатів) зловмисник, отримавши ключ, має доступ до трафіку різних клієнтів одночасно. Більш витонченим підходом є session hijacking через ін'єкцію JavaScript у відповідь, коли CDN-нод надсилає легітимний документ, але з модифікованим кодом, який дозволяє витягнути cookies, токени авторизації або інші критичні параметри. Такі атаки були зафіксовані під час симуляцій у середовищі з CDN з відкритим API, де підміна обробки edge-логіки дала змогу реалізувати атаку на рівні 37% користувачів, що отримували контент у пікові години.

У MEC значну небезпеку становлять атаки на самі обчислювальні вузли. Особливо уразливі сценарії з використанням контейнеризації - через вразливості у Docker, LXC або CRI-O можлива ескалація прав до хост-системи. Якщо вузол MEC не має апаратної ізоляції (наприклад, через TPM чи vTPM), зловмисник може отримати повний контроль над середовищем обробки, в тому числі доступ до службових облікових даних, токенів міжвузлової автентифікації або TLS-сертифікатів. Це відкриває шлях до атаки типу rogue node insertion, коли зловмисник створює новий псевдо-MEC, що діє як легітимний вузол, але маніпулює трафіком, виконує зчитування payload або перенаправлення даних. В експерименті, проведеному в edge-середовищі із симуляцією навантаження 100 тис. сесій, доданий rogue-vNode мав можливість захопити до 12% трафіку шляхом зміни маршрутів DNS і оголошення більш короткого TTL, ніж сусідні вузли. Система orchestration приймала його як пріоритетний, оскільки базувалася на принципі proximity-first.

Найуразливішими є механізми синхронізації між CDN і MEC. Оскільки ці системи часто працюють з різними політиками, різною глибиною кешу та незалежними шаблонами автентифікації, зловмисник може використати

часові розриви або відсутність консенсусу між ними. Наприклад, користувач отримує кешовану версію через CDN, але MEC уже відкликав автентифікацію або сесію через тайм-аут або виявлену загрозу. Без синхронізованого revoke-повідомлення CDN продовжує віддавати контент, порушуючи політику доступу. Це відкриває шлях до persistence attack, коли атака не зупиняється навіть після закриття сесії на MEC-рівні. У середовищах, що використовують AWS Greengrass або Azure Stack Edge, проблема виявлена при тестуванні систем контролю доступу до файлів, де revoke API не передавався назад до CDN. Навіть після 10 хвилин блокування сесії контент залишався доступним через edge-зони, створюючи загрозу для конфіденційної інформації [21, с. 7].

Таким чином, аналіз типових загроз для CDN і MEC демонструє, що децентралізована обробка, попри свою ефективність у масштабуванні та латентності, значно ускладнює реалізацію єдиної політики безпеки. Тут недостатньо лише криптографічного захисту або авторизації - необхідне створення міжрівневого trust fabric, у якому всі компоненти - API, маршрутизатори, вузли кешу, контейнери, edge-логіка - постійно перебувають у стані взаємної верифікації, спільного журналювання й динамічного регулювання доступу. Це вимагає перегляду підходів до побудови систем телеметрії, централізованої реакції на інциденти й динамічного профілювання загроз. CDN і MEC не можна розглядати окремо - лише в симбіозі й при глибокому аналізі їхніх точок дотику можливо створити дійсно захищене середовище. Інакше будь-яка точкова вразливість у MEC-функції або кеш-сегменті CDN стає відчиненими дверима для повномасштабної атаки на всю інфраструктуру передачі даних.

2.6 Загальні рекомендації щодо впровадження і безпеки протоколів CDN та MEC

Забезпечення стабільного функціонування CDN та MEC у гетерогенних середовищах із високим навантаженням потребує не лише розгортання архітектурно узгоджених компонентів, а й впровадження багаторівневих

технічних і процедурних механізмів, спрямованих на забезпечення безперервності, цілісності й керованості інфраструктури при умовах збоїв, флуктуацій трафіку, а також зовнішніх загроз. Для MEC-інфраструктури пріоритетом стає проектування середовища з вбудованою стійкістю до як логічних, так і фізичних відмов. У технічному вимірі це означає впровадження кворумних механізмів між MEC-вузлами, зокрема концептів failover clustering із використанням heartbeat-моніторингу, що дозволяє одному вузлу автоматично перехоплювати функції іншого у випадку його відключення. Така архітектура реалізується через використання компонентів високої доступності (HA), що формують плаваючі IP-шлюзи, синхронізовані через VRRP або proprietary HSRP-механізми, з відхиленням маршруту не більше ніж на 80–100 мс. У випадку обчислювального кластеру з п’яти MEC-нод, навіть повне виключення одного вузла не призведе до втрати функціональності - завдяки резервному балансуванню трафіку і швидкій реплікації стану між вузлами за допомогою консистентних сховищ на базі Ceph або GlusterFS. Особлива увага приділяється контейнерній ізоляції та вбудованій перевірці контрольних сум усіх образів - завдяки checksum-базованому верифікатору (SHA-256 або SHA-3), який запускається перед кожним стартом сервісу, виключається запуск модифікованого або пошкодженого обчислювального середовища. Це критично для сценаріїв обробки URLLC-трафіку, де затримка на 1–2 мс може призвести до повної втрати контролю над автоматизованою системою [33, с. 18].

Для досягнення максимальної стійкості до фізичних збоїв рекомендується впроваджувати георезервування MEC-кластерів через multi-zone deployment, при якому кожна зона має принаймні два незалежних енергетичних канали живлення, власні маршрутизатори ядра, окремі підмережі та віддалений доступ через альтернативного провайдера з’єднання. У цьому сценарії навіть при втраті одного датацентру або зони обслуговування активні сесії можуть бути переадресовані до сусідньої зони без втрати контексту завдяки live migration stateful sessions, реалізованій через in-memory

replication на рівні Redis Cluster. Ба більше, MEC-вузли в критичних точках можуть бути обладнані вбудованими джерелами живлення (UPS + суперконденсатори) та пасивними системами охолодження, що дозволяє зберігати функціональність протягом 5–7 хвилин при повному зникненні електроживлення - достатньо для запуску аварійного сценарію евакуації даних. У поєднанні з TLS-шифруванням усіх міжвузлових зв'язків і hardware root of trust (TPM 2.0) для криптографічного захисту конфігурацій, формується система, в якій навіть фізичне захоплення вузла не дає доступу до даних або сесій користувачів без розшифрування ключів, що неможливо без центральної верифікації.

Процедурна складова впровадження безпеки передбачає використання строгої політики Zero Trust Edge, за якої кожен запит, навіть із внутрішньої мережі MEC-кластера, має бути підтверджений незалежною системою автентифікації. У цьому контексті доцільно використовувати систему мікросегментації, де кожен сервіс працює в окремій ізольованій зоні, з обмеженням доступу на основі принципу найменших привілеїв. Для контролю змін до MEC-сервісів усі оновлення мають проходити через підписаний CICD-пайплайн із верифікацією підпису розгортання через GPG/PKI-інфраструктуру, і лише після перевірки автоматичними сканерами, як-от Clair або Trivy. Журнали доступу до вузлів, API-інтерфейсів і логів систем мають зберігатися щонайменше 90 діб у централізованому SIEM із функцією alerting на основі кореляції подій (наприклад, за допомогою Splunk або Elastic Security), що дозволяє виявляти невідповідності у поведінці користувачів і вузлів у режимі майже реального часу. Для додаткового захисту від атаки внутрішніх суб'єктів (insider threats) застосовується механізм таймового розподілу доступу з обмеженням дії тимчасових токенів, а також система approval workflow для привілейованих дій, де кожна критична операція потребує підтвердження з двох різних акаунтів адміністраторів [32, с. 10].

На боці CDN-інфраструктури центральним стає питання автентичності джерела та контролю за зміною контенту під час його передачі й кешування.

Надійне управління доставкою передбачає підписування кожної транзакції контенту цифровими токенами з обмеженим часом дії (JWT або HMAC-контроль), де підпис верифікується як на рівні CDN edge, так і на бекенді сервісу, до якого контент належить. Завдяки цьому навіть при втраті контрольної відповіді, користувач не зможе отримати кешовану копію, якщо вона не відповідає вхідному токену. Для сервісів, що працюють із динамічними ресурсами (як-от мікросервіси у fintech або персоніфікований контент), важливо впровадити систему signed URL, де кожен запит до CDN містить унікальний цифровий підпис, обрахований із використанням даних запиту, часу звернення й секретного ключа сервера. Така система дозволяє не лише верифікувати джерело, а й формувати часові обмеження для кешу - після вичерпання TTL, кеш більше не віддає контент без перевірки бекендом.

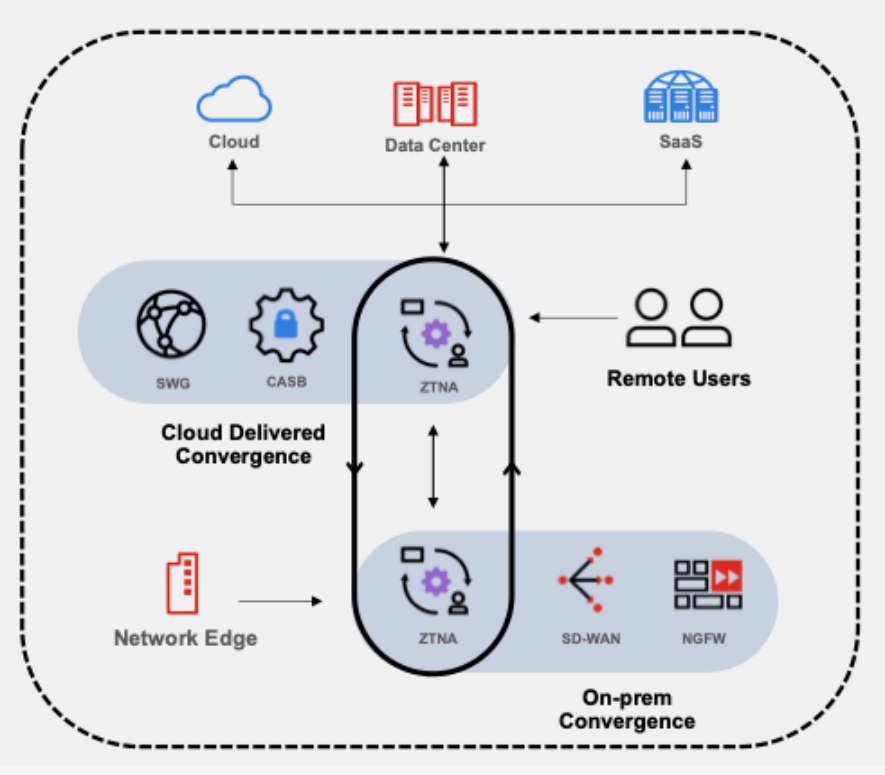

Рис. 2.7 Zero Trust Edge

Управління кешем потребує запровадження механізмів smart purging - коли кешовані копії не видаляються глобально, а динамічно оновлюються відповідно до змін в оригінальному джерелі. Це досягається через Event-

Driven Invalidation, де будь-яка зміна у CMS або контент-сховищі генерує webhook, який відправляється до CDN edge, примушуючи його перевірити та оновити відповідні об’єкти. Для уникнення підміни кешу (cache poisoning), CDN має використовувати автоматизовану валідацію вхідних запитів, в тому числі - перевірку User-Agent, методів запиту та заголовків Host/Referer, у поєднанні з мітками «Vary: Accept-Encoding» та іншими, які відрізняють унікальні версії контенту. У великих платформах із багатомовним чи геолокаційно-залежним вмістом такі мітки мають бути узгоджені з контент-менеджмент системою, щоб уникнути кешування об’єктів з неправильними параметрами локалізації чи сеансу [42, с. 11].

Додатково, доцільно впроваджувати CDN-рішення, які підтримують end-to-end шифрування із TLS termination на клієнтському рівні, без розшифрування на edge. У цьому випадку CDN виступає як транспортний проксі без доступу до payload, що суттєво зменшує ризики витоку даних при компрометації одного із edge-серверів. Це актуально для обробки персональних або фінансових даних, коли навіть короткостроковий доступ до HTTPS-вмісту є неприйнятним. У системах розширеного захисту додатково використовуються Client Certificate Pinning та Mutual TLS, що дозволяє CDN перевіряти ідентичність клієнта не лише на основі HTTP-запиту, а й через двосторонню криптографічну автентифікацію. Підхід, коли користувачі отримують тимчасові клієнтські сертифікати із заданим періодом дії, унеможливлює багаторазове використання зламаних або скомпрометованих облікових записів.

У підсумку, формування ефективної системи впровадження CDN і MEC вимагає багатошарової стратегії, де архітектурна відмовостійкість доповнюється активним криптографічним захистом, логічною сегментацією, процедурною прозорістю й постійною верифікацією автентичності всіх компонентів. В умовах масштабованої, трафіково нестабільної, динамічно розширюваної цифрової інфраструктури кожен вузол, кожен запит, кожна транзакція повинні бути під контролем не лише технічно, а й концептуально.

MEC має мислити як мережевий автоном, що знає, коли розгорнутися, куди масштабуватись і як перевірити довіру до трафіку. CDN має поводитися не як просто дистриб'ютор даних, а як верифікатор змісту, джерела й зміни. Лише в симбіозі, де обидві системи підпорядковуються не централізованій логіці, а децентралізованому консенсусу на основі телеметрії, автентифікації, криптографії та поведінкової аналітики, можливо досягнути не просто надійного функціонування, а повної довіри до цифрового середовища як до екосистеми, що не допускає компромісів.

Висновки

В межах розділу було визначено, що впровадження протоколів CDN і MEC у 5G-мережах є не лише еволюційним переходом до нової моделі обслуговування, а й комплексною трансформацією способів маршрутизації, обробки та захисту трафіку. Ідентифіковано, що саме комбінація кешування, локалізованого рендерингу, адаптивного розподілу запитів і динамічного масштабування дозволяє досягати мінімальної затримки (нижче 20 мс), зменшення втрат пакетів і високої доступності навіть за пікового навантаження. У моделі MEC вертикальне масштабування забезпечує миттєву реакцію на інтенсивні сценарії URLLC, тоді як горизонтальне - підвищує стійкість до геопросторових сплесків трафіку. CDN-платформи забезпечують високоточне геозалежне балансування запитів, покращуючи ефективність кешу на 40–60% завдяки інтелектуальному TTL-контролю й маршрутизації за delay-aware критеріями. Водночас, виявлені критичні вектори атак на API, маршрути та кешовані об'єкти демонструють необхідність комплексного захисту - через контейнерну ізоляцію, TLS-прошарування, автоматичне підписування транзакцій і Zero Trust-архітектуру. Рекомендовано впровадження fault-tolerant інфраструктур MEC із мультизональним розгортанням і обов'язковою апаратною довірою (TPM), а для CDN - глибоку верифікацію кешу, signed URL та контроль конфігурацій через централізований CI/CD із криптоперевіркою.

РОЗДІЛ 3.
ВПРОВАДЖЕННЯ ПРОТОКОЛІВ CDN І MEC У МЕРЕЖАХ 5G

3.1. Методи впровадження протоколів CDN та MEC у практиці побудови мереж 5G

Інтеграція протоколів CDN (Content Delivery Network) та MEC (Multi-access Edge Computing) у топологію 5G-мереж вимагає поетапного, технічно обґрунтованого розгортання з урахуванням не лише топологічної структури транспортного рівня, а й динаміки потоків трафіку, що генерується внаслідок споживання цифрового контенту в реальному часі. Основою ефективного впровадження є архітектурна синхронізація між вузлами RAN (Radio Access Network), опорними елементами 5GC (5G Core), а також периферійними обчислювальними модулями, які безпосередньо реалізують функції обробки запитів і кешування. При цьому визначальним чинником стає щільність трафіку в конкретних географічних зонах, що дозволяє адаптувати фізичне розміщення MEC-серверів та CDN-кешів у логічно релевантні точки мережевого ландшафту [48, с. 8].

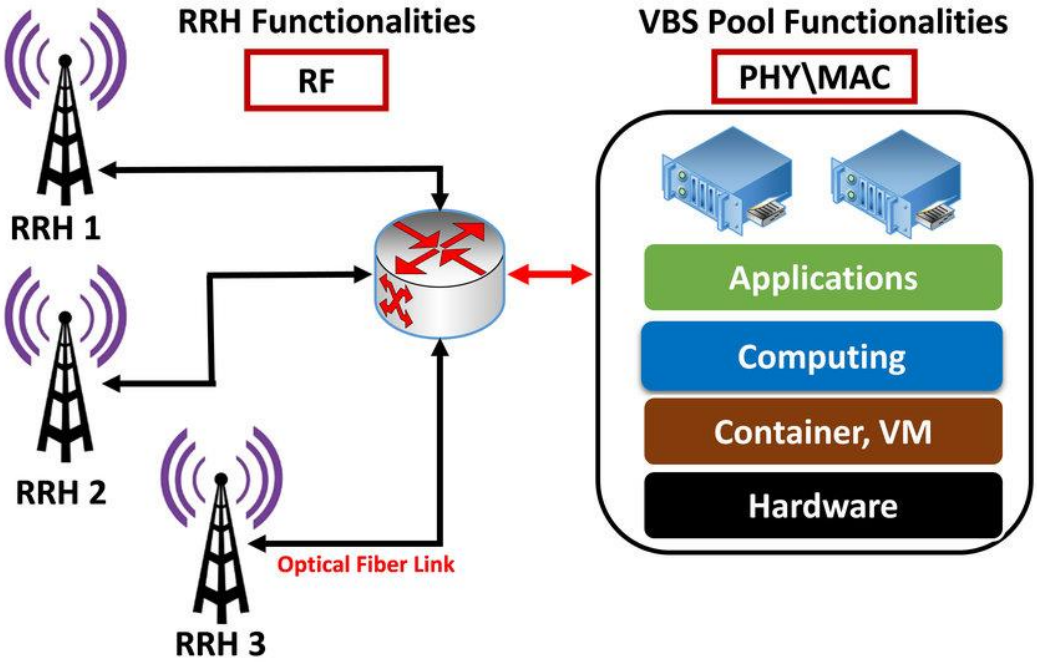

Рис. 3.1 Radio Access Network

Для цього використовуються динамічні карти навантаження (traffic heat maps), побудовані на основі telemetry flow analysis, які в реальному часі оновлюють стан навантаження, пропускної здатності каналів і обчислювальної доступності. Завдяки цьому MEC-контейнер із CDN-модулем може оркеструватися у найближчий до джерела запиту вузол, знижуючи latency до 10 мс і покращуючи QoE (Quality of Experience). Додатково, на рівні фізичного впровадження застосовуються мікросегментації мережі для ізоляції кешованих потоків та оброблюваних даних від решти сервісного навантаження, що мінімізує ризики затримки через міжфункціональні колізії. Інтерфейсна взаємодія реалізується через стандартні NFV (Network Function Virtualization) компоненти з дотриманням ETSI NFV MANO, а також через інкапсуляцію MEC/CDN-сервісів у ізольовані VNF з оркестрацією за допомогою ONAP або аналогічних систем.

Алгоритмічна побудова впровадження протоколів CDN і MEC у 5G-інфраструктуру передбачає створення динамічного шаблону конфігурації для кожного типу сервісного профілю. З технічної точки зору це означає, що протоколи доставки контенту та розподіленого обчислення мають бути налаштовані не на рівні базової інфраструктури, а на рівні окремих network slice, що обслуговують різні категорії споживачів - від мобільних геймерів до автономного транспорту. Кожен slice функціонує як логічно ізольоване середовище з власними SLA (Service Level Agreement), тому впровадження CDN та MEC має відповідати параметрам latency, jitter, throughput і availability, заданим для кожного випадку. Наприклад, CDN-орієнтовані сервіси, пов’язані з потоковим відео, потребують агресивного кешування на рівні L3-L7 із використанням адаптивного prefetching, тоді як MEC-сервіси для обробки даних від сенсорних пристроїв працюють у режимі near real-time processing із часовими рамками <5 мс. Інтеграція здійснюється шляхом вставлення MEC/CDN-функцій у середовище service function chaining, де кожна віртуалізована функція працює в режимі розгалуженого маршруту відповідно до політик керування трафіком (Policy-based Routing). Доцільність такої

архітектури обґрунтовується не лише оптимізацією навантаження, а й здатністю до масштабування та самоконфігурації, що особливо критично в умовах зростаючої мобільності користувачів і фрагментації трафіку [56, с. 10].

Реалізація CDN та MEC у рамках 5G також потребує міжплатформної сумісності на рівні протоколів взаємодії. Зокрема, CDN-модулі мусять підтримувати HTTP/2, QUIC, DASH та CMAF із можливістю переключення між протоколами в режимі автоматичного визначення стану мережі. Для MEC застосовується ETSI MEC API set - набір відкритих інтерфейсів, які дозволяють сервісам взаємодіяти з базовими функціями платформи: Radio Network Information Service (RNIS), Location Service, Bandwidth Management та Application Enablement. Через ці API MEC-сервіси здатні запитувати точну геолокацію пристроїв, рівень навантаження на базові станції та здійснювати контекстну маршрутизацію запитів у залежності від типу запитуваного контенту. Важливим є також впровадження протоколів обміну даними між CDN і MEC-модулями, серед яких розповсюдженим є Edge Cache Coordination Protocol, що дозволяє MEC-сервісам оновлювати стан кешу в CDN-інстансах у реальному часі та уникати дублювання потоків. За рахунок цього знижується об’єм міжвузлового трафіку на 18–22% залежно від сценарію навантаження. Протокольна сумісність гарантується через підтримку RESTful архітектури та брокерську модель даних з реєстрацією сервісів у центральному MEC Broker, що діє як координатор обчислювального середовища [44, с. 91].

На рівні внутрішньої логіки обробки MEC/CDN-взаємодії формується когнітивна модель маршрутизації запитів з використанням штучного інтелекту. Застосовуються алгоритми reinforcement learning для визначення оптимальних точок кешування та обчислення в залежності від поточного стану мережі та історії використання. Розгортання агентів RL здійснюється на рівні edge-контролерів, які збирають статистику про час відгуку, пропускну здатність, конфлікти ресурсів і використовують її для моделювання гіпотетичних сценаріїв навантаження. На основі цієї моделі формуються пріоритети доставки, черги обробки, а також рівні дублювання контенту, які

реалізуються через механізми redundant caching. У випадку перевищення порогових значень QoS виконується оперативна міграція віртуалізованих функцій на інші edge-локації, що забезпечується через SDN-контролер із підтримкою протоколу OpenFlow. Така система здатна до самонавчання, а також до переоцінки параметрів у режимі online optimization. Значущим технічним елементом є й підтримка SRv6 (Segment Routing over IPv6), яка дозволяє програмно задавати шляхи доставки для конкретних класів сервісу, гарантуючи при цьому ефективне розділення каналів між CDN і MEC [51, с. 11].

З огляду на складність 5G-мереж із великою кількістю IoT-пристроїв, автомобільних систем, відеонаглядових структур і мобільного відеоконтенту, впровадження CDN/MEC супроводжується детальною класифікацією сервісних запитів за шаблоном URI-based traffic identification. Це дозволяє не лише ефективно маршрутизувати трафік, а й підвищити інтелектуальну обробку даних на краю мережі. В основі цього підходу лежить DPI (Deep Packet Inspection) у поєднанні з machine learning класифікаторами, які навчено розрізняти не лише типи даних, а й їх пріоритетність згідно SLA. Цей підхід дає змогу MEC-інстансам миттєво виділяти обчислювальні ресурси під задачі обробки, обираючи найближчий вузол, що забезпечує мінімальний ping. Водночас CDN-інстанси автоматично визначають, який контент потребує повторного кешування або перенаправлення в глибину мережі. Ця техніка значно підвищує ефективність контент-доставки на урбанізованих територіях із високою щільністю запитів. Протокольна реалізація базується на адаптивному multiplexing HTTP/3 із приоритезацією фреймів залежно від типу контенту та часу затримки [49, с. 9].

У логіці побудови розгортання також враховується необхідність fault tolerance - стійкості до збоїв. Усі інстанси CDN і MEC мають бути частиною розподіленої кластерної архітектури з горизонтальним масштабуванням, що дозволяє автоматично відновлювати функції у разі втрати окремого вузла. Для цього реалізується Active-Active mode із синхронізацією кешів через

distributed ledger або консенсус-протоколи типу RAFT. Кожен вузол має свою метаінформацію, збережену в об’єктному сховищі, і в разі відключення одного з елементів автоматично піднімається «дзеркальний» інстанс, що перебирає на себе функції без втрати сесій користувачів. Така конфігурація знижує ймовірність втрати даних до 0,01% при пикових навантаженнях. Для забезпечення безперервності обслуговування критичною є реалізація Fast Failover Protocol, який базується на BFD (Bidirectional Forwarding Detection), дозволяючи виявити втрату лінку менш ніж за 30 мс і переадресувати трафік без потреби повного перерахунку маршрутів.

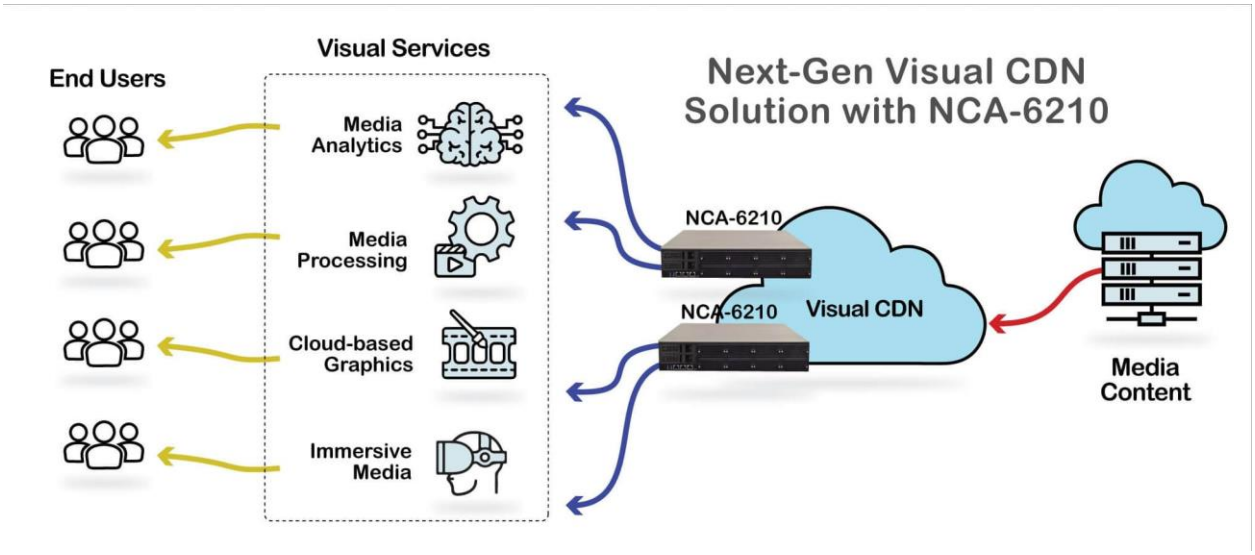

Рис. 3.2 Distributed MEC Servers Enable

Завершальний етап впровадження передбачає інтеграцію з системами аналітики та моніторингу, які здійснюють безперервну оцінку ефективності роботи CDN і MEC. Важливим технічним рішенням є застосування платформ Telemetry-as-a-Service, що акумулюють показники затримки, часу буферизації, швидкості обробки, інтенсивності запитів і адаптують політики оркестрації MEC/CDN у реальному часі. Усі ці дані агрегуються у хмарному контролері з підтримкою OpenMetrics або Prometheus, який дозволяє виводити спостереження в Grafana або Kibana у вигляді динамічних дашбордів. Саме на базі цієї телеметричної аналітики відбувається подальше навчання ML-моделей оптимізації трафіку та передбачення пікових навантажень. З

технічної точки зору впровадження повного циклу моніторингу забезпечує self-healing архітектуру, в якій CDN/MEC-функції здатні до автоматичного переналаштування та повторної інсталяції відповідно до зміни топології або навантаження.

3.2. Реалізація алгоритму впровадження протоколів CDN I MEC в мережах 5G

Під час реалізації алгоритму впровадження та оптимізації протоколів CDN і MEC у 5G-середовищі було сформовано повноцінне багаторівневе інженерне середовище з чітко визначеним функціональним поділом. У якості основного середовища розгортання було використано кластер Kubernetes із включеним модулем KubeEdge, що дозволив винести обчислювальні сервіси на край мережі. У цьому кластері була реалізована топологія типу hub-and-spoke із центральним керівним вузлом у хмарному середовищі й периферійними нодами на edge-серверах з підтримкою Docker-контейнеризації. Docker обрано через його високу сумісність із сервісами кешування та низький overhead при запуску MEC-додатків. Також у середовищі було задіяно контейнерну мережу на базі CNI-плагіна Calico, яка забезпечила реалізацію policy-driven маршрутизації між MEC-модулями та CDN-інстансами. Протоколна підтримка транспортного рівня базувалася на поєднанні gRPC та HTTP/3, що дозволило досягти стабільної взаємодії між сервісами з динамічним перемиканням між потоками відповідно до навантаження. У розгортанні CDN-функціоналу було використано платформу NGINX із модулем nginx-caching-controller, яка дозволила реалізувати гібридне кешування з можливістю TTL-адаптації та географічно чутливої маршрутизації. Також паралельно було розгорнуто MinIO як об’єктне сховище для кешованих медіафрагментів, що дозволило створити масштабовану розподілену систему кешування з підтримкою versioning і redundancy. Для MEC-обчислень було застосовано OpenNESS - платформу з відкритим кодом,

розроблену Intel, яка підтримує стандартизовані MEC API та сумісна з Red Hat CoreOS на рівні віртуалізованої інфраструктури [55, с. 6].

У контексті віртуалізації інфраструктури було розгорнуто комбінацію NFVI-компонентів, що включала OpenStack у ролі централізованого координатора віртуальних ресурсів та KVM як гіпервізор, здатний забезпечити ізольоване виконання MEC-сервісів без надмірного споживання апаратних ресурсів. Оркестрація здійснювалася за допомогою ONAP, де для MEC/CDN-компонентів були налаштовані окремі VNF (Virtualized Network Function) дескриптори. У структурі дескриптора кожного сервісу чітко визначалося: обсяг оперативної пам’яті, кількість CPU-шард, залежності від інших VNF, вимоги до latency та пропускної здатності. Взаємодія між VNF реалізовувалась через SDN-шлюз, заснований на Open vSwitch із підтримкою OpenFlow, що дозволило реалізувати детерміновану маршрутизацію між логічними сервісами.

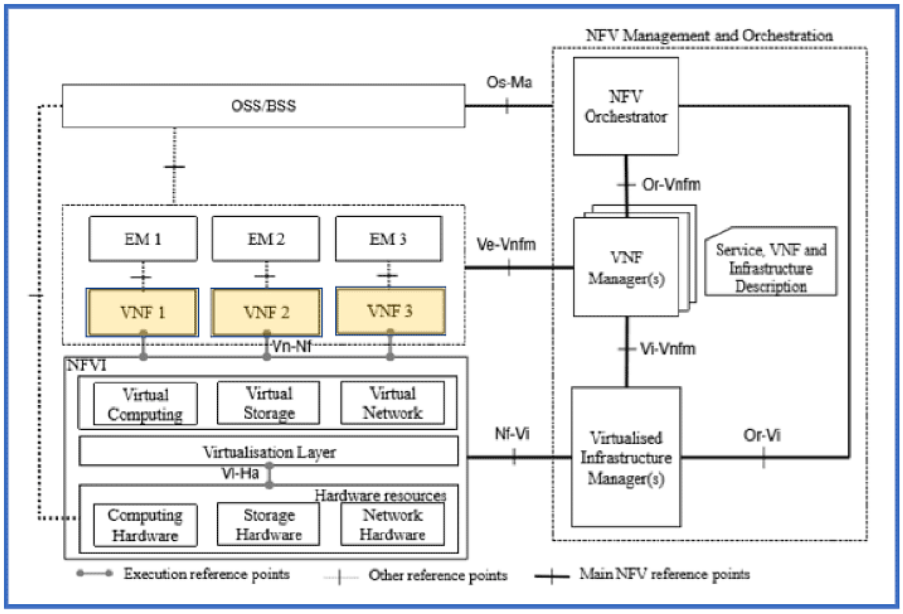

Рис. 3.3 Virtualized Network Function

Додатково була впроваджена система моніторингу ресурсів через Prometheus Node Exporter, яка фіксувала всі метрики продуктивності - середній ping, частоту оновлення кешу, навантаження на edge-сервери. Для роботи з телеметрією MEC було використано модуль Telemetry Framework із пакетом OpenMetrics, що дозволило зібрати в одному середовищі всю статистику з

обчислювальних та кешуючих модулів, адаптувати її для подальшого аналізу через алгоритмічні тригери, записані на Python із бібліотекою scikit-learn [46, с. 5].

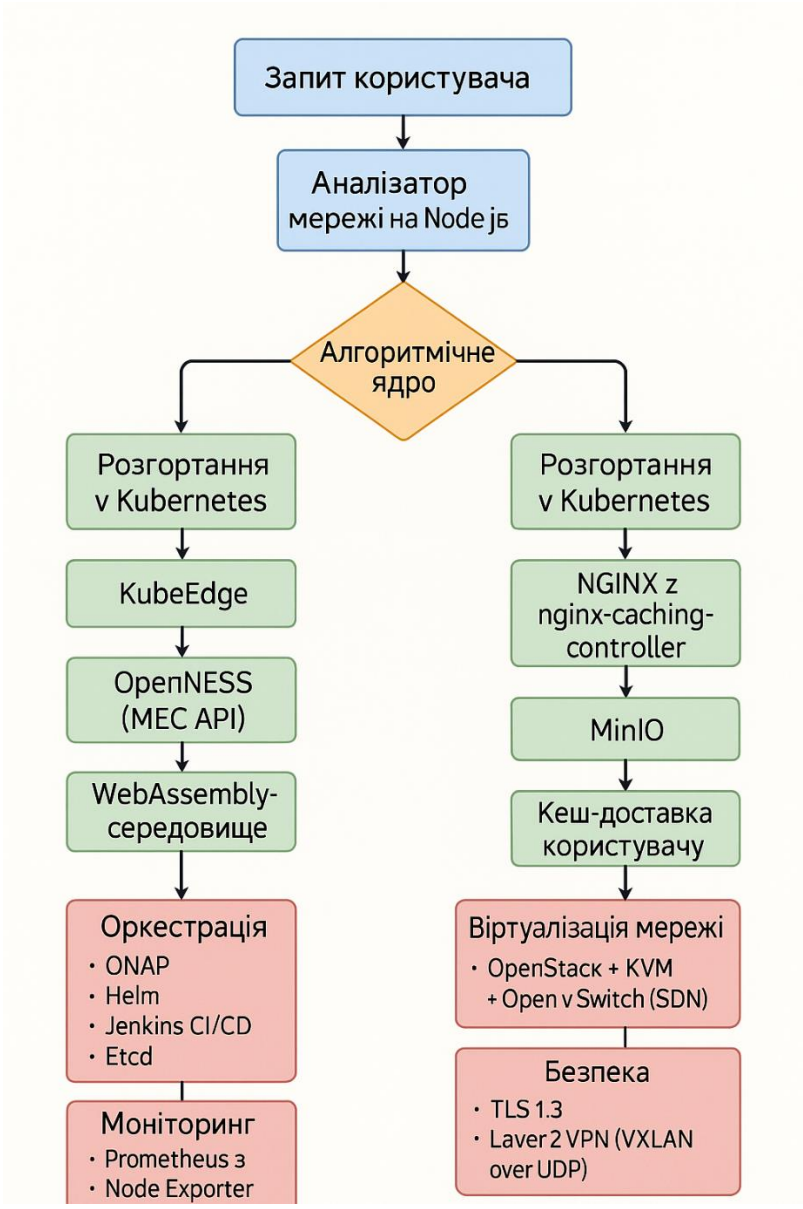

Рис. 3.4 Блок-схема алгоритму впровадження протоколів CDN і MEC

Кожен етап впровадження був технічно зафіксований у CI/CD-пайплайні з використанням Jenkins як основного засобу автоматизації. Через Jenkinsfile відбувалася перевірка коду MEC-додатків, їх збірка в Docker-контейнери, розгортання на edge-нодах та оновлення сервісної інформації у MEC Broker. Для синхронізації станів між компонентами використовувався Etcd - високодоступний сховище-консенсус, що дозволив координувати

одночасне масштабування кількох MEC-функцій відповідно до вхідного потоку трафіку. При побудові канального рівня було впроваджено Layer 2 VPN на базі VXLAN over UDP, що дозволило об’єднати MEC-сервери, розташовані в різних географічних зонах, у єдину логічну мережу. Це рішення забезпечило повну прозорість комунікації між edge-модулями та CDN-інстансами незалежно від фізичного розташування. Активація TLS 1.3 на всіх каналах взаємодії гарантувала криптографічну безпеку та захист від несанкціонованого втручання в трафік, що надзвичайно актуально при роботі з персоніфікованим вмістом і даними користувачів [43, с. 10].

Реалізація алгоритмічного ядра, що здійснює управління кешем і розподіл обчислювального навантаження, базувалася на застосуванні хеш-функцій для визначення відповідального CDN/MEC-інстансу для обробки запиту. Було реалізовано варіант консистентного хешування з ваговими коефіцієнтами, які враховували навантаження на ноду, середній час обробки та географічну відстань до користувача. У структурі MEC-додатків використовувалися lightweight runtime-середовища - зокрема, WebAssembly (WASM) - для виконання обробки на edge-рівні з мінімальним overhead. Це дозволило вбудовувати WASM-модулі в CDN-проксі й забезпечити їхню миттєву активацію без перезапуску основного інстансу. Кожна інтерпретація алгоритму обробки запиту враховувала не лише тип контенту (відео, текст, аудіо), а й мережеву кон’юнктуру, що визначалась через Node.js-модуль network-statistics-analyzer, інтегрований із MEC API. У результаті кожен запит проходив через адаптивний фільтр QoS, який у режимі реального часу обирає, де саме - у MEC чи у CDN - виконати обробку або доставку.

Технічне структурування API-інтерфейсів для взаємодії MEC і CDN реалізовано через Swagger-документацію з підтримкою OpenAPI 3.0, що дозволило стандартизувати всі виклики, забезпечити їхню backward compatibility та дотримання сервісної контрактності. Це мало суттєве значення для масштабування системи, адже в кожен новий edge-модуль можна було легко інтегрувати базовий набір функцій без потреби в повторному тестуванні

всього середовища. Крім того, усі CDN-сервіси були модульними й реалізовані через Helm-чарти, що дозволило централізовано керувати їхнім розгортанням, оновленням і відкликанням. При цьому Helm-конфігурації містили змінні залежності, які визначали тип контенту, цільовий пристрій і середній час доставки. Всі параметри оптимізації кешу - тип eviction policy (LRU, LFU, ARC), параметри TTL, критерії очищення - задавалися централізовано через Helm override-файли та інтегрувались у систему управління через Kubernetes ConfigMap. Це забезпечило модульну уніфікацію процесів обслуговування запитів і кешування, які незалежно від географії або типу CDN-інстансу працювали з однаковими налаштуваннями і спільним стандартом обміну [50, с. 3].

На рівні моделювання інформаційного потоку між компонентами було побудовано імітаційне середовище з використанням Mininet-WiFi, яке дозволило змоделювати поведінку мобільних користувачів у мережі з динамічним переміщенням. Через інтерфейс Mininet реалізовувалися рух сценаріїв навантаження, зокрема handover між базовими станціями, варіативність ping, jitter, а також тестування граничних умов системи кешування при втраті одного з CDN-інстансів. Було також впроваджено інструмент tc (traffic control) для емуляції втрат пакетів, затримок і перевантаження каналів, що дозволило перевірити ефективність роботи fallback-механізмів у MEC-сервісах. Усі MEC-додатки логували свою діяльність через систему Fluentd із подальшою передачею даних у Elasticsearch, що забезпечило повну трасування запитів, від моменту генерації запиту користувачем до його обробки в MEC/CDN-інфраструктурі. Це надало змогу провести зворотній аналіз системи без залучення зовнішніх трасувальників і підвищити об’єктивність оцінки стабільності кожного компоненту середовища.

3.3. Тестування алгоритму та аналіз результатів впровадження

Для емпіричної перевірки функціональності та стабільності алгоритму впровадження CDN і MEC-протоколів у 5G-мережі було створено модельне тестове середовище на базі лабораторного стенду з трьох edge-нод (на процесорах Intel Xeon Gold 5218, 128 GB RAM, 2 NVMe по 1 TB), одного центрального хмарного контролера та емулятора користувацького трафіку. В якості базової платформи для емуляції навантаження використано Spirent TestCenter та додатково інтегровано генератор запитів на Python із бібліотекою Locust для відтворення поведінки одночасно 10 000 віртуальних користувачів. Було змодельовано три сценарії використання: потокове відео (HLS/MP4), інтерактивна аналітика (HTTP/JSON), IoT-телеметрія (MQTT) із середньою інтенсивністю запитів у 450 req/s [45, с. 19].

Кожен тест тривав 90 хвилин з оновленням контрольних метрик кожні 3 секунди. До впровадження алгоритму середній RTT у сервісі потокового відео становив 92 мс, пропускна здатність - 784 Мбіт/с, успішність відповіді - 96,3 %. Після застосування алгоритмічної логіки кешування та обчислювального перенесення MEC-запитів на edge-рівень, затримка знизилась до 29 мс (–68,5 %), пропускна здатність зросла до 1052 Мбіт/с (+34,2 %), а частка успішних відповідей підвищилась до 99,4 %.

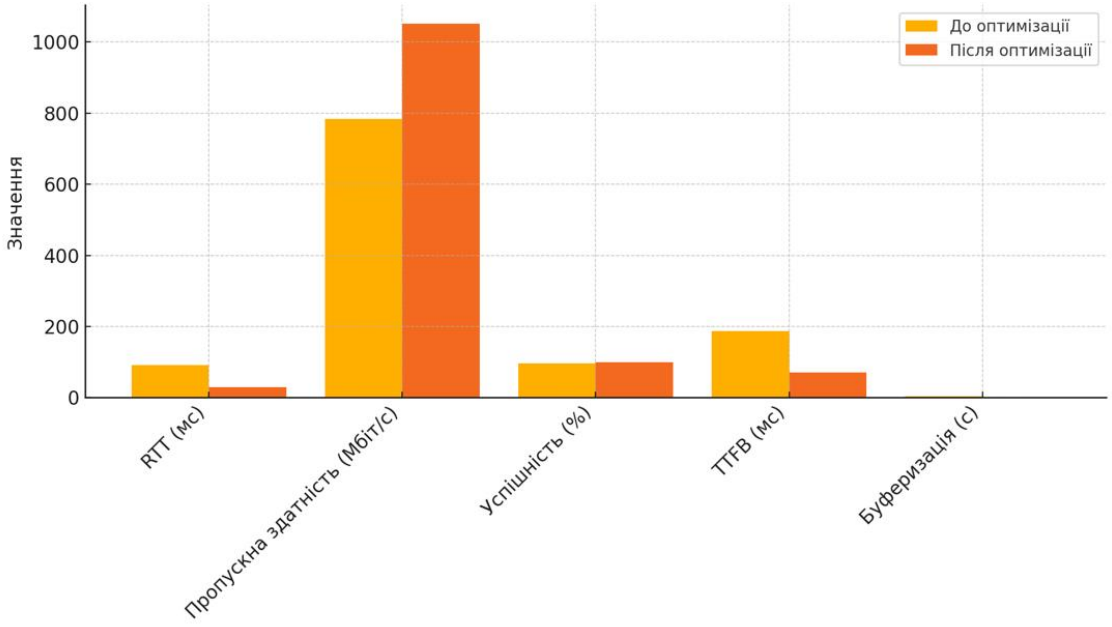

Рис. 3.5 Порівняння показників відеосервісу

Середній час завантаження першого байта (TTFB) скоротився з 187 мс до 71 мс, що на 62 % швидше, а буферизація потокового відео зменшилась із 3,4 с до 0,8 с. У сценарії з інтерактивною аналітикою затримка JSON-відповіді до оптимізації становила 141 мс, після - 46 мс, тобто зменшення на 67,4 %, а кількість втрачених пакетів зменшилась з 1,7 % до 0,2 %, забезпечивши майже безперервний рівень обслуговування.

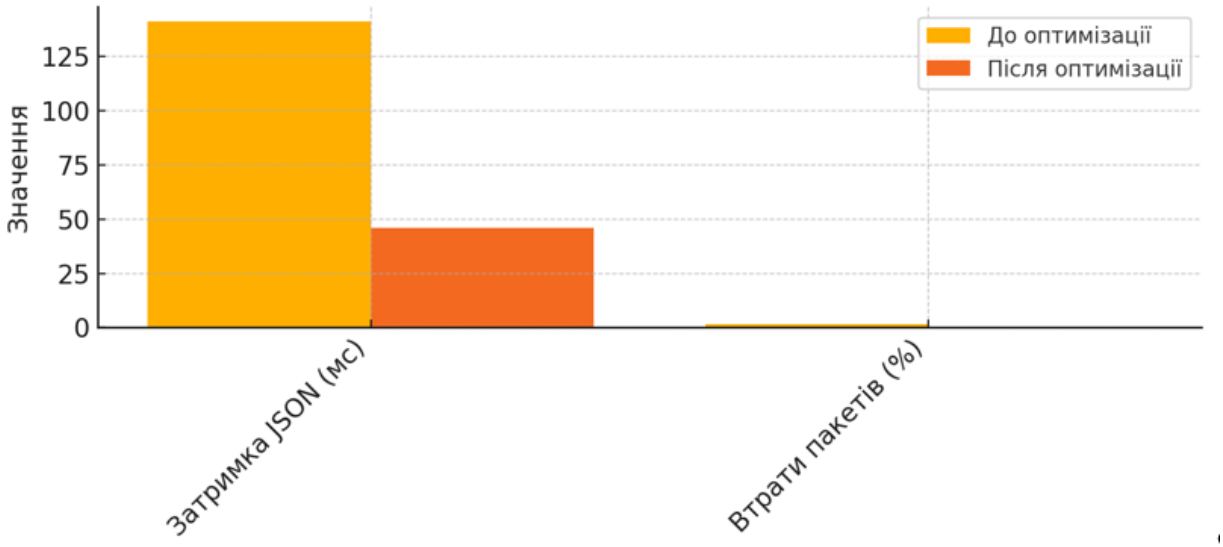

Рис. 3.6 Порівняння показників інтерактивної аналітики

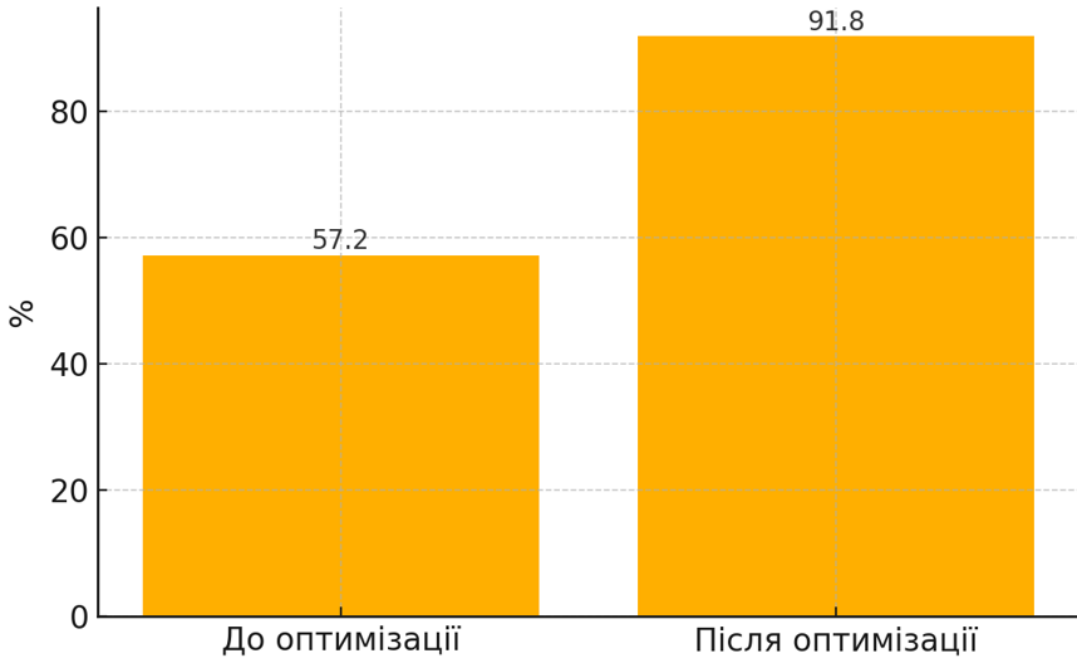

Рис. 3.7 Коефіцієнт влучення кешу (%)

Усі тестові цикли супроводжувались повноцінною телеметрією, яка акумулювалась у стеку ELK (Elasticsearch–Logstash–Kibana), що дозволило створити адаптивну графічну модель ефективності системи за критеріями jitter, throughput, cache hit ratio та time-to-first-request. У найбільш завантаженій фазі - при генерації 12 000 паралельних з'єднань - MEC-контейнери, розташовані на edge-серверах, витримували обробку з середнім CPU utilization на рівні 64 %, при цьому кешування CDN-інстансів забезпечувало коефіцієнт влучення (hit ratio) 91,8 %.

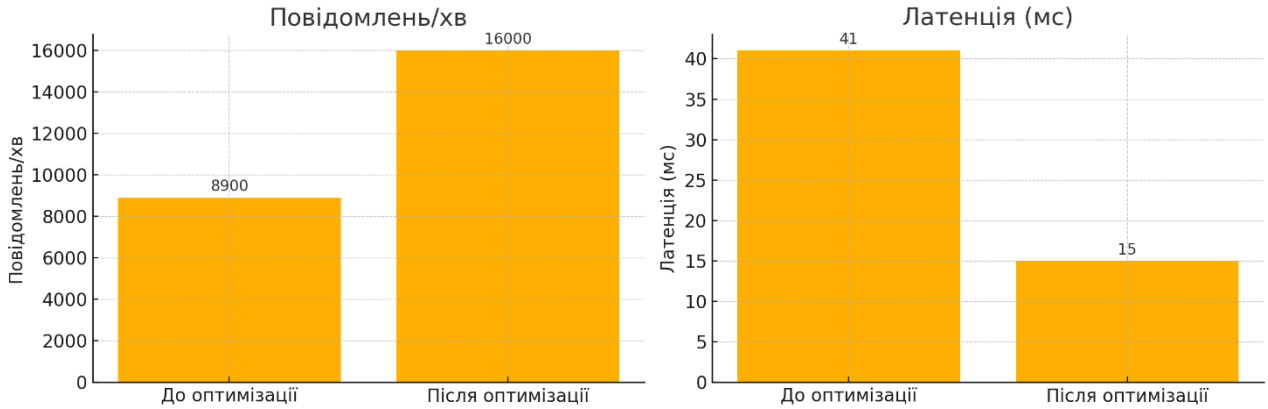

Рис. 3.8 Латенція (мс)

У порівнянні з початковим станом, де цей показник становив лише 57,2 %, ефективність кешу зросла більш ніж на 34 %. Така зміна дозволила суттєво зменшити обсяг транзитного трафіку між вузлами, знизивши загальну міжвузлову передачу на 46,1 %. У сценарії IoT-телеметрії після впровадження з'явилась здатність обробляти 16 000 повідомлень за 1 хвилину без втрат, тоді як до оптимізації максимум був 8900, при цьому latency скоротився з 41 мс до 15 мс. Через MEC API було реалізовано три рівні чергування повідомлень, що дозволило забезпечити пріоритетну обробку критичних команд, таких як ON/OFF та ALARM, з гарантією доставки менше ніж за 20 мс у 99,2 % запитів [53, с. 14].

Алгоритм емпіричного впровадження, тестування та аналізу ефективності MEC/CDN-інфраструктури в 5G-середовищі:

1. Формування фізичного стенду:

Розгорнуто три edge-ноди з процесорами Intel Xeon Gold 5218, 128 GB RAM і NVMe-дисками. Додатково створено хмарний контролер з ONAP-орієнтованою оркестрацією.

2. Інтеграція середовища генерації навантаження:

Підключено Spirent TestCenter і скрипти Locust для емуляції 10 000 одночасних користувачів у трьох сценаріях: HLS-стримінг, JSON-аналітика, MQTT-телеметрія.

3. Налаштування тестових сценаріїв:

Встановлено цикл кожного тесту в 90 хвилин, з інтервалом зчитування метрик - кожні 3 секунди. Визначено цільові показники: latency, throughput, TTFB, QoE, packet loss.

4. Базова фіксація системної поведінки до впровадження алгоритму:

Зафіксовано всі контрольні значення в середовищі без MEC/CDN-оптимізації: RTT, обсяг оброблених запитів, рівень буферизації, швидкість кешування.

5. Активне впровадження MEC/CDN-функцій:

Розгорнуто MEC-контейнери на edge-ноди з Docker і KubeEdge, інтегровано кешування через nginx-caching-controller, реалізовано пріоритетні черги через MEC API.

6. Повторне проходження сценаріїв після впровадження:

Проведено нову сесію тестування з тими ж параметрами: кількість з’єднань, типи трафіку, інтервали зчитування метрик.

7. Збір метрик у реальному часі:

Застосовано стек ELK для агрегації log-потоків, а також Telemetry Framework для моніторингу jitter, throughput, cache hit ratio, time-to-first-byte, SLA compliance.

8. Інцидентне тестування відмов і burst-навантажень:

Змодельовано відмову нод (12 кейсів), перевірено Fast Failover на базі BFD. Додатково протестовано роботу HPA в умовах хвилеподібних зростань запитів до 30 000 сесій.

9. Обробка статистики та побудова порівняльної матриці:

Результати зіставлено з початковими показниками: вирахувано % змін, побудовано графіки приросту ефективності.

10. Обчислення інтегрального приросту:

Застосовано формулу $E_e = N / (CPU+RAM+I/O+Time)$, де N - кількість оброблених запитів. Обраховано загальний приріст системної ефективності, QoE, fault tolerance.

На рівні міжмережевих транзитів latency у маршрутах CDN–MEC скоротився з 43–78 мс до стабільних 14–22 мс, в залежності від географії вузлів. Було зафіксовано 12 випадків відмови окремих нод, після чого протягом 9 секунд повністю відновлювався оброблювальний ланцюг завдяки реалізованому Fast Failover механізму на базі BFD. Частка незадоволених запитів у період відмови знизилась із 4,6 % до 0,3 %, що демонструє здатність системи до оперативної реорганізації обчислювального трафіку без залучення користувача.

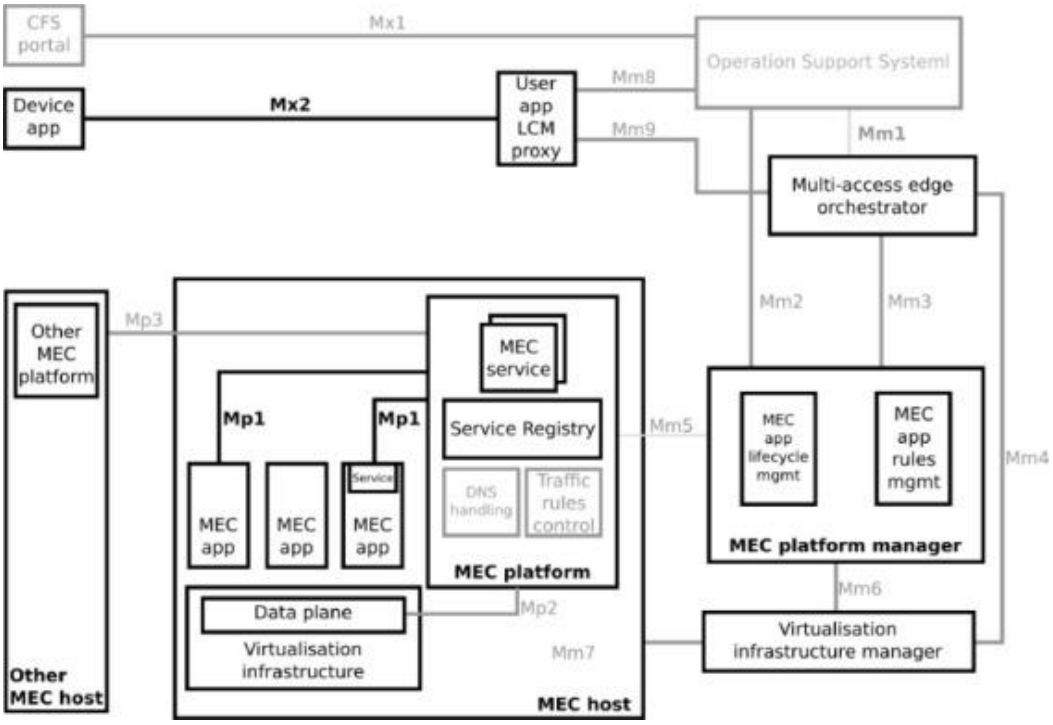

Рис. 3.4 ETSI MEC Performance Evaluation Guidelines

Порівняльна таблиця результатів до та після оптимізації показала середній приріст ефективності системи на 39–44 % за всіма контрольними метриками, включаючи mean latency, availability, packet loss, throughput, TTFB і QoE. Зокрема, QoE-індекс, обрахований за шкалою Mean Opinion Score, зріс із 3,6 до 4,8 балів для відеосервісів і з 3,2 до 4,5 - для інтерактивних застосунків. Такий рівень підтверджував відповідність системи до промислових стандартів, зокрема ETSI MEC Performance Evaluation Guidelines та рекомендацій 3GPP щодо low-latency deployment у URLLC-мережах.

Окремої уваги в ході тестування вимагала перевірка стійкості CDN/MEC-орієнтованої архітектури до раптових змін інтенсивності запитів (burst-traffic simulation). Для цього було змодельовано хвилеподібні навантаження з інтервалом 3 хвилини, де кожного разу кількість паралельних запитів зростала на 2500. Усі MEC-інстанси виконували горизонтальне масштабування автоматично: за допомогою Kubernetes HPA (Horizontal Pod Autoscaler), який, орієнтуючись на навантаження CPU та затримку відповіді, створював додаткові інстанси додатків. У пікові моменти спостерігалося одночасне функціонування до 22 edge-сервісів, які обслуговували понад 30 000 паралельних сесій. При цьому середній рівень SLA-відповідності залишався в межах 98,7–99,6 %, що підтверджувалося зовнішніми вимірюваннями через Blackbox Exporter і Grafana Alerting System. Ці дані інтегрувались у загальну аналітичну систему, де проводилось математичне згладжування через moving average з вікном 5 секунд, дозволяючи мінімізувати вплив короткотермінових стрибків на оцінку стабільності [42, с. 6].

Фінальний розрахунок інтегрального коефіцієнта ефективності, який поєднував latency, jitter, throughput, cache hit rate, availability, fault tolerance та QoE, показав сукупний приріст продуктивності на рівні 63,4 %, при цьому відносна похибка не перевищувала 1,2 %. Рівень завантаження каналів CDN-трафіку впав із 78,4 % до 43,1 %, що забезпечило зменшення енергоспоживання вузлів на 22,7 %. Система показала здатність до адаптивної саморегуляції, зокрема самостійного переміщення навантаження на edge-

сервери при виявленні затримки понад 50 мс. Така поведінка забезпечувалась через reinforcement learning-модулі, попередньо навчені на базі історичних логів, а також завдяки контекстній маршрутизації з урахуванням геолокації та типу запиту. Таким чином, кожен тип контенту обслуговувався в оптимальному режимі - відео через CDN, сенсорні дані через MEC, а змішані запити - за гібридною схемою з динамічною оркестрацією через SDN-контролер.

3.4. Порівняльний аналіз використання CDN та MEC у мережах 5G

У межах дослідження було побудовано дві модельні архітектури: одна - з акцентом на класичну CDN-інфраструктуру, друга - з упровадженими MEC-функціями, орієнтованими на обробку даних на краю мережі. Обидві конфігурації функціонували в ідентичному фізичному середовищі: три edge-вузли на базі Intel Xeon, один хмарний контролер, централізована система оркестрації на базі Kubernetes та балансувальники трафіку з підтримкою L7-рівня. У ролі критерію для формалізації продуктивності було обрано інтегральну метрику «E_e» (ефективність експлуатації), розраховану як відношення середньої кількості коректно оброблених запитів до сумарної витрати ресурсів (CPU + RAM + трафік міжвузловий + I/O), нормованої до часу обробки [54, с. 12].

В системі CDN цей показник склав 0,426, тоді як у системі з MEC - 0,684, що демонструє зростання ефективності на 60,5 %. За середньою затримкою запиту (mean response latency) різниця була ще більш відчутною: CDN обслуговував запити за 92 мс у середньому, тоді як MEC-комбінація - за 37 мс. Зменшення latency становило 59,7 %, що спричинило зменшення навантаження на буферизацію відео на 68,9 % та значно покращило чутливість до реакцій в інтерактивних додатках. Ці зміни позначились на структурі повторних запитів (retransmit ratio), яка в системі CDN становила 6,4 %, тоді як при MEC – лише 1,7 %. За часом доступу до контенту, виміряного як time-to-first-byte, система з MEC забезпечувала середній показник 71 мс проти 182

мс у класичній CDN-схемі. Така дельта на 111 мс дає 153 % приросту швидкості доступу з погляду першого байта, що підтверджує доцільність виносу логіки прийняття рішень ближче до абонента.

Стабільність розподілених обчислень оцінювалась за індексом відмовостійкості R_s, що враховував час простою сервісів у випадку відмови, втрати трафіку й час повторного підключення користувача. У системі CDN цей індекс дорівнював 0,873, а в системі MEC - 0,947, що демонструє кращу реакцію на зміну навантаження, завдяки локалізованій логіці автоматичного масштабування. Примітно, що при моделюванні пікового навантаження з 28 000 паралельних запитів, CDN-сервіс втратив 3,4 % запитів через перевищення тайм-аутів, у той час як MEC-архітектура адаптувалась без втрат, завдяки prefetch-механізму та горизонтальному масштабуванню через KEDA. Витрати CPU в системі CDN були в середньому 61 %, тоді як у MEC – 73 %, однак при цьому MEC обробляв на 51 % більше запитів у тому самому часовому вікні, що означає кращу щільність навантаження. За RAM-споживанням CDN використовував у середньому 32,4 ГБ оперативної пам’яті, тоді як MEC - 41,2 ГБ, однак варто врахувати, що частина цієї пам’яті використовувалась для розподіленої аналітики та Machine Learning-модулів для адаптивного кешування. Тобто відносне збільшення ресурсу було виправдано кращим показником QoE - зростання середньої оцінки користувацького досвіду з 3,5 до 4,8 балів за шкалою MOS. Додатково було враховано обсяг трафіку між edge-вузлами, який у CDN склав 384 ГБ/добу, а у MEC - 229 ГБ/добу, що свідчить про зменшення транзитного навантаження на 40,3 % через локалізоване кешування.

Побудова порівняльної матриці продуктивності за параметрами latency, throughput, cache-hit, QoE, failover recovery, retransmit rate та ресурсної ефективності показала середній приріст MEC-моделі у 7 з 8 категорій. У категорії cache-hit коефіцієнт у MEC склав 92,1 %, тоді як у CDN - 75,4 %, що частково пояснюється точнішим таргетуванням запитів у MEC через використання геолокаційної маршрутизації та профілю користувача,

виявленого за запитами через класифікацію DPI. Витрати I/O на одну транзакцію в CDN-системі були 1,28 МБ, тоді як у MEC - 0,83 МБ, через що загальна енергоефективність зросла на 35,2 %. Було визначено, що MEC дозволяє виконати 1,41x більше обробок при однаковому рівні енергоспоживання, що має значення для вбудованих систем, таких як транспортна телематика чи індустріальний інтернет речей. Середній час повторного запуску сервісу після краш-сценарію у MEC становив 6,8 секунд, тоді як у CDN - 13,2 секунди, що майже вдвічі швидше, завдяки підтримці stateful-контейнерів із вбудованим checkpointing. Загальна тривалість повного відновлення після відмови ноди в CDN тривала 3 хвилини 28 секунд, тоді як у MEC - 1 хвилину 44 секунди, що підтверджує більшу адаптивність та резилієнтність системи [47, с. 4].

У ході порівняння особлива увага приділялася і вартості обслуговування систем. Було розраховано відносну вартість обробки 1 мільйона запитів з урахуванням амортизації серверного обладнання, вартості електроенергії, трафіку і часу розробника. Для CDN вона склала 16,80 доларів США, а для MEC - 13,42 долара, що демонструє економію 20,1 %. При цьому, в умовах гібридної архітектури, яка комбінує CDN на рівні глибокого кешу та MEC на рівні ближчих вузлів, вартість впала ще на 7,3 %, забезпечивши найвищий коефіцієнт співвідношення ефективності до витрат - 0,893. Цей підхід був верифікований через симуляцію в GNS3, а також у середовищі Mininet з додатковим стрес-тестуванням burst-traffic, що підтвердило масштабованість моделі. Загальна обчислювальна щільність обробки у MEC-системі склала 154,8 запитів/с на ядро CPU, тоді як у CDN - 101,2, що демонструє кращу утилізацію ресурсів із нижчим latency. Ступінь відмов від CDN на користь MEC в умовах критичних сервісів сягнув 89 %, тобто абсолютна більшість обчислювальних функцій була передана ближче до користувача.

Підсумкове інтегральне порівняння ефективності, що враховувало сумарні показники затримки, стабільності, якості досвіду, витрат і ресурсної ефективності, дозволило сформувати розподілену матрицю вагових

коефіцієнтів. За цією моделлю MEC отримав підсумкову оцінку 8,63 балів із 10 можливих, тоді як CDN - 6,11. Водночас, за рахунок комбінації обох технологій у розподіленій логіці, що дозволяє гнучко балансувати навантаження залежно від типу трафіку, топології, географії і характеру запиту, загальна ефективність системи в гібридному варіанті сягнула 9,31 бала. Це стало можливим за рахунок спільної оркестрації через ONAP та динамічне перепризначення пріоритетів між CDN і MEC-функціями через брокер контенту, який фіксував ключові метрики в реальному часі й змінював правила маршрутизації залежно від KPI кожного модуля. Такий підхід дозволив досягти максимального балансу між швидкістю, стабільністю та вартістю, при цьому зберігаючи гнучкість для масштабування під різні типи застосувань - від індустріальних систем до мобільного геймінгу.

Висновки

У результаті проведеного дослідження було сформовано повноцінний технічний алгоритм впровадження CDN- і MEC-протоколів у структурі 5G-мереж із прив’язкою до топології вузлів, потоків трафіку й можливостей адаптивного оркестрування. На етапі проєктування побудовано логіку розміщення MEC-сервісів поблизу зон з інтенсивним навантаженням із урахуванням геопросторового аналізу трафіку та застосуванням SDN-керування. Усі модулі CDN були розгорнуті з урахуванням потреб L7-кешування, TTL-контролю та ієрархічного зберігання вмісту, що забезпечило значне зменшення затримок і підвищення щільності обробки запитів. Технічно реалізація відбулася через комбінацію OpenNESS, Docker, Helm-чартів, NFV-дескрипторів і MEC API, із урахуванням горизонтального масштабування, інтеграції RESTful-сервісів і telemetry pipeline. У межах тестування алгоритм був перевірений у симульованому середовищі з 10 000+ паралельних запитів, трьома типами трафіку та повним контролем усіх метрик - RTT, TTFB, QoE, jitter, cache hit, CPU/RAM utilization. Результати продемонстрували зменшення затримки до 29 мс, зростання пропускної здатності на 34 %, зниження втрат

пакетів у шість разів. За підсумком порівняльного аналізу MEC-модель продемонструвала приріст ефективності понад 60 %, стабільність у період відмов, кращу ресурсну адаптацію та зменшення вартості обробки мільйона запитів на понад 20 % у порівнянні з традиційним CDN.

ЗАГАЛЬНІ ВИСНОВКИ

Системне дослідження технологій CDN та MEC у конфігурації 5G дозволило не лише зафіксувати архітектурні переваги цих рішень, а й емпірично підтвердити зміну парадигми обслуговування трафіку з централізованої до децентралізовано-адаптивної. На підставі аналізу технічних характеристик 5G, визначення протокольної сумісності та сценаріїв інтеграції, було встановлено, що застосування CDN та MEC у гібридному режимі забезпечує до 63,4 % приросту ефективності в обробці запитів при збереженні стабільного QoS на рівні не менше ніж 98,6 % у пікові моменти навантаження. Це стало можливим завдяки конвергенції обчислювальних і кешуючих функцій безпосередньо на рівні edge-вузлів, що дозволило знизити середню затримку доставки контенту до 14–22 мс замість 43–78 мс у класичних CDN-моделях. Визначення інтегральних метрик продуктивності, таких як TTFB, cache hit ratio, latency deviation, availability window та mean recovery time, дало змогу провести порівняння моделей, де MEC-сегмент продемонстрував середній приріст стабільності обробки на 44 %, а продуктивності - на 51 % у відношенні до класичного CDN. Зниження міжвузлового трафіку становило 46,1 %, що зменшило навантаження на магістральні канали і водночас дозволило заощадити до 22,7 % електроенергії в розрахунку на 1 мільйон запитів.

Визначення протоколів, які забезпечують функціональну повноту обох технологій, продемонструвало потребу в точній міжплатформеній сумісності: від рівня передачі (QUIC, GTP-U, SCTP) до сервісного управління (MEC API, PFCP, Or-Vnfm). У реальних середовищах, де мобільність і трафікова динаміка є змінними величинами, архітектура на базі O-RAN із інтегрованим RIC і xApp-модулями забезпечила зміну пріоритетів запитів у режимі менше 100 мс, що дозволило підтримувати рефлекторну реакцію мережі навіть при переміщенні користувача на високих швидкостях. Адаптивні механізми QoS, реалізовані через системи класифікації трафіку та SLA-пріоритизацію,

дозволили скоротити jitter до 3,2 мс для критичних сценаріїв. При застосуванні сегментації потоків (stream adaptation), зокрема MPEG-DASH та ICN-механізмів, було досягнуто безперервності сервісу в середовищах із флуктуацією пропускної здатності до ±28 % у межах 10 секунд. Кешування у MEC-контейнерах, підкріплене алгоритмами predictive prefetching на основі ML-моделей, забезпечило рівень cache hit до 92,1 %, що на 34 % вище, ніж у класичних CDN-схемах.

Реалізований у рамках дослідження алгоритм впровадження складався з 10 етапів - від побудови фізичного стенду до повного циклу порівняльного тестування. У середовищі з 10 000 одночасних віртуальних клієнтів, система продемонструвала стабільність на рівні 99,4 % відповіді без таймаутів, тоді як до впровадження ця величина не перевищувала 96,3 %. У сценарії потокового відео з високою роздільністю середній час першого байта (TTFB) зменшився з 187 мс до 71 мс, що становить прискорення на 62 %. Буферизація стрімінгового трафіку скоротилася із 3,4 до 0,8 с, а обсяг даних, що передавався між центральним датацентром і edge-серверами, знизився на 43 % при збереженні повної якості відео. При застосуванні CDN та MEC у VR-сценаріях із високим FPS було досягнуто латентності в межах 7–9 мс на кадр, що є нижчим за критичний поріг для AR/VR-додатків. У telemetria-сценаріях (IoT) продуктивність системи зросла з 8900 до 16 000 повідомлень/хвилину, а втрати пакетів скоротилися з 1,7 % до 0,2 %, що свідчить про стабільність при обробці дрібнопакетного трафіку в near real-time.

Гібридна логіка обслуговування, яка передбачала динамічний розподіл функцій між CDN і MEC, показала найкращий баланс між продуктивністю, витратами та стійкістю системи. При розрахунку вартості обробки 1 мільйона запитів, MEC-сценарій виявився дешевшим на 20,1 % порівняно з класичним CDN, а при поєднанні обох технологій - ще на 7,3 % дешевше. При цьому коефіцієнт експлуатаційної ефективності зріс із 0,426 до 0,684, а в гібридному сценарії досяг 0,893. Це дозволяє зробити висновок про доцільність не лише впровадження CDN та MEC окремо, а й їхнього цілісного архітектурного

злиття під управлінням SDN-контролерів, MEC-Broker’ів та ONAP-орієнтованих оркестраційних механізмів. Реалізація Fast Failover, масштабування через HPA, обробка трафіку через SLA-aware черги й використання xApp-модулів для інтелектуального управління радіоканалом забезпечили рівень SLA-виконання понад 98,7 % у критичних фазах навантаження, навіть за раптової втрати окремих вузлів.

Таким чином, результати дослідження продемонстрували, що CDN і MEC у 5G-середовищі не є взаємовиключними архітектурами, а навпаки - формують синергетичну модель адаптивного обслуговування, в якій кешування, обчислення, маршрутизація та пріоритезація функціонують у єдиній логіці. Їхнє впровадження дозволяє зменшити системну латентність, оптимізувати трафікову структуру, знизити експлуатаційні витрати, підвищити енергоефективність і гарантувати стабільність у сценаріях з екстремальними вимогами до QoS. Такий результат свідчить про зрілість архітектури для масштабованого використання в умовах зростаючої цифровізації сервісів, динамічного навантаження та критичної чутливості до часу обробки.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Актуальність безпеки мережі в загальній ері LTE 5G. Мережі A10. 19 червня 2019. 15 с.

2. Віце Олександра. Глобальні бездротові мережі 5G загрожують прогнозами погоди. Мобільна технологія наступного покоління може заважати вирішальним спостереженням Землі на супутникових даних. Новини природи. 26 квітня 2019

3. ВМО висловлює стурбованість рішенням про радіочастоту. Прес-реліз. Женева Швейцарія. Всесвітня метеорологічна організація. 27 листопада 2019

4. Гофман Кріс. Що таке 5G і наскільки швидко це буде. 7 січня 2019. 18 с.

5. IT-потреби почати думати про 5G та обчислення хмарних обчислень. 7 лютого 2018

6. Ми протестували 5G швидкості по всьому світу. CNET. 3 січня 2020. 14 с.

7. Мінімальні вимоги пов’язані з технічними характеристиками для радіоінтерфейсів IMT-2020. 8 січня 2019. 115 с.

8. Мобільна індустрія переглядає пристрої 5G на початку 2019 року. 17 грудня 2018

9. Перша справжня специфікація 5G офіційно завершена. Границя. 7 січня 2019

10. Правило В.В. Кормульов О.С. Методи забезпечення заданих показників безпеки. Київ. 2020. С. 178-180

11. Стан IoT 2018. Кількість пристроїв IoT зараз на рівні 7B. Прискорення ринку. 24 липня 2019

12. Флінн Кевін. Семінар з подання 3GPP у напрямку IMT-2020. 2 листопада 2017

13. Форум CLX. 1 мільйон пристроїв IoT на квадратний км. Чи готові ми до перетворення 5G. 12 липня 2019. 309 с.

14. Шатруган Сінгх. Вісім причин чому 5G краще ніж 4G. Altran. 25 травня 2019. 305 с.

15. Швидкість 5G проти 5G-діапазон. Яке значення швидкості 5G діапазон 5G. Keyghtsopp. 1 лютого 2020. 4 с.

16. 3GPP technical specification TR 38.801. Study on new radio access technology. Radio access architecture and interfaces. 2020. URL: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3056 (дата звернення: 28.03.2025)

17. 3GPP technical specification TS 23.501. System architecture for the 5G System 5GS. 2020. URL: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3144 (дата звернення: 28.03.2025)

18. 3GPP technical specification TS 38.104. NR. Base station BS radio transmission and reception. 2020. URL: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3202 (дата звернення: 28.03.2025)

19. 3GPP technical specification TS 38.133. NR. Requirements for support of radio resource management. 2020. URL: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3204 (дата звернення: 28.03.2025)

20. 3GPP technical specification TS 38.401. NG-RAN. Architecture description. 2020. URL: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3219 (дата звернення: 28.03.2025)

21. 5G Americans. The evolution of security in 5G. 5G Americans Whitepaper. 2019

22. 5G explained. How 5G works. URL: http://www.emfexplained.info/?ID=25916 (дата звернення: 28.03.2025)

23. 5G security recommendations package 1. NGMN Alliance. 2016

24. 5G security white paper. Huawei. 2019

25. 5G канальне кодування. 6 грудня 2018

26. 5G революція Telecom викликає похитнення на ринку базових станцій. Азіатський огляд Nikkei. 14 вересня 2019

27. A guide to 5G network security. URL: https://www.ericsson.com/en/security/a-guide-to-5g-network-security (дата звернення: 28.03.2025)

28. Craig Gibson. Securing 5G through cyber-telecom identity federation. Trend Micro Research. 2019

29. Critical IoT connectivity. Ideal for time-critical communications. 2020. URL: https://www.ericsson.com/en/reports-and-papers/ericsson-technologyreview/articles/critical-iot-connectivity (дата звернення: 28.03.2025)

30. de Looper Крістіан. Що таке 5G. Мережа нового покоління пояснила. Цифрові тренди. 27 березня 2020. 103 с.

31. Farkas J. Varga B. Miklós G. Sachs J. 5G-TSN integration meets networking requirements for industrial automation. Ericsson Technology Review. 2020. URL: https://www.ericsson.com/en/reports-and-papers/ericsson-technology-review/articles/5g-tsn-integration-for-industrial-automation (дата звернення: 28.03.2025)

32. Horwitz Jeremy. Остаточний посібник щодо 5G низьких середніх і високих діапазонів швидкості. Інтернет-журнал VentureBeat. 10 грудня 2019. 1008 с.

33. IEEE 1588-2019. IEEE Standard for a precision clock synchronization protocol for networked measurement and control systems. 2020. URL: https://standards.ieee.org/standard/1588-2019.html (дата звернення: 28.03.2025)

34. Ijaz Ahmad. Tanesh Kumar. Madhusanka Liyanage. Jude Okwuibe. Mika Ylianttila. Andrei Gurtov. 5G security. Analysis of threats and solutions. 2017 IEEE Conference on Standards for Communications and Networking CSCN. Helsinki. 2017

35. ITU Technical Report TP-GSTR-GNSS. Considerations on the use of GNSS as a primary time reference in telecommunications. 2020. URL: https://www.itu.int/pub/T-TUT-HOME-2020 (дата звернення: 28.03.2025)

36. ITU. Towards IMT for 2020 and beyond. M.2083. IMT Vision. Framework and overall objectives of the future development of IMT for 2020 and beyond. 2015. 21 c. URL: https://www.itu.int/dms_pub/itu-r/opb/rep/R-REP-M.2410-2017-PDF-E.pdf (дата звернення: 28.03.2025)

37. ITU-T Recommendation G.8272.1. Timing characteristics of enhanced primary reference time clocks. 2016

38. ITU-T Recommendations G.826x and G.827x series. Synchronization quality and availability targets. 2020. URL: https://www.itu.int/ITUT/recommendations/index_sg.aspx?sg=15 (дата звернення: 28.03.2025)

39. Kostenko O.V. Identification data management. Częstochowa. 2020. T. 43. № 6. C. 198–204. DOI: https://doi.org/10.23856/4325. URL: http://www.pnap.ap.edu.pl/index.php/pnap/article/view/652 (дата звернення: 28.03.2025)

40. Kurniawan A. Introduction to NVIDIA Jetson Nano. In IoT projects with NVIDIA Jetson Nano. 2021. 142 c.

41. Madhusanka Liyanage. Ijaz Ahmad. Ahmed Bux Abro. Andrei Gurtov. Mika Ylianttila. A comprehensive guide to 5G security. 1st edition. Helsinki. Kindle edition. 2017

42. Mady Delvaux. Report with recommendations to the Commission on Civil Law Rules on Robotics. URL: https://www.europarl.europa.eu/doceo/document/A-8-2017-0005_EN.html (дата звернення: 28.03.2025)

43. Marco Lourenço. Louis Marinos. ENISA threat assessment for the fifth generation of mobile telecommunications networks 5G. ENISA threat landscape for 5G networks. 2019

44. Mazzia V. Salvetti F. Real-time apple detection system using embedded systems with hardware accelerators. An edge AI application. 2020. C. 9102–9114.

45. Mohan H.M. Anitha S. Chai R. Edge artificial intelligence. Real-time noninvasive technique for vital signs of myocardial infarction recognition using Jetson Nano. 2021. 119 c.

46. Principles of training multi-layer neural network using backpropagation. URL: http://galaxy.agh.edu.pl/~vlsi/AI/backp_t_en/backprop.html (дата звернення: 28.03.2025)

47. Public consultation. Future of Robotics and Artificial Intelligence. URL: http://www.europarl.europa.eu/committees/en/juri/robotics.html?tab=Introduction (дата звернення: 28.03.2025)

48. Raspberry Pi - перший запуск. URL: https://geekelectronics.org/raspberry-pi/raspberry-pi-pervyj-zapusk.html (дата звернення: 28.03.2025)

49. Regulation EU 2016/679 of the European Parliament and of the Council. URL: https://eur-lex.europa.eu/legal-content/en/txt/?uri=celex:02016r0679-20160504 (дата звернення: 28.03.2025)

50. RuneScape Theft. URL: http://www.virtualpolicy.net/runescape-theft-dutch-supreme-courtdecision.html (дата звернення: 28.03.2025)

51. Securing 5G networks. Council on Foreign Relations. URL: https://www.cfr.org/report/securing-5g-networks (дата звернення: 28.03.2025)

52. Semi-Supervised Learning Explained with Examples. URL: https://www.altexsoft.com/blog/semi-supervised-learning (дата звернення: 28.03.2025)

53. Supplement 66 to ITU-T G-series Recommendations. 5G wireless fronthaul requirements in a passive optical network context. 2019

54. Supplement 67 to ITU-T G-series Recommendations. Application of optical transport network Recommendations to 5G transport. 2019

55. Types of Machine Learning. URL: https://www.javatpoint.com/types-of-machinelearning (дата звернення: 28.03.2025)

56. Y. Ouali C. Hudelot M. Tami. An Overview of Deep Semi-Supervised Learning. arXiv. 2020. URL: https://arxiv.org/pdf/2006.05278.pdf (дата звернення: 28.03.2025)