МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЧЕРКАСЬКИЙ ДЕРЖАВНИЙ ТЕХНОЛОГІЧНИЙ УНІВЕРСИТЕТ

Кваліфікаційна наукова
праця на правах рукопису

Усік Павло Сергійович

УДК 621.396.93 (043.3)

ДИСЕРТАЦІЯ

МЕТОДИ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ РОЗПОДІЛЕНОЇ ОБРОБКИ ДАНИХ В КОМП'ЮТЕРНИХ СИСТЕМАХ ОПЕРАТОРІВ СТІЛЬНИКОВОГО ЗВ'ЯЗКУ

123 «Комп'ютерна інженерія»
Подається на здобуття ступеня доктора філософії

Дисертація містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

П.С. Усік

Наукові керівники:

Смірнов Олексій Анатолійович
доктор технічних наук, професор

Миронець Ірина Валеріївна
кандидат технічних наук, доцент

Черкаси – 2021

АНОТАЦІЯ

Усік П.С. **Методи підвищення ефективності розподіленої обробки даних в комп'ютерних системах операторів стільникового зв'язку**. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття ступеня доктора філософії за спеціальністю 123 «Комп'ютерна інженерія». – Черкаський державний технологічний університет, Черкаси, 2021.

Дві основні тенденції спонукають індустрію безпроводового зв'язку розвивати мережі стільникового зв'язку п'ятого покоління: стрімке збільшення попиту на безпроводові широкосмугові послуги, які потребують значно більших швидкостей передачі даних та мережі значно більшої ємності, які можуть надавати відео та інші ресурсоємні послуги; та послуги Інтернету Речей (IoT), що спонукають до необхідності масового підключення пристроїв, а також до необхідності наднадійного зв'язку з наднизькою затримкою.

Можна визначити низку різних областей застосування, де будуть використовуватись дані мережі: сюди входить V2X комунікація (комунікація транспортних засобів між собою та з іншими об'єктами інфраструктури); промислова автоматизація та комунальні програми; безпроводові медичні послуги; споживчі та бізнес-послуги віртуальної та доповненої реальності; деякі додатки розумного міста; розумні будинки та велика кількість застосувань мобільного широкосмугового зв'язку.

При цьому, із розвитком стільникових мереж з'являються нові більш досконалі мережеві архітектури для передачі даних та керування. Проте залишається ряд невирішених завдань та проблемних місць, які необхідно вирішувати та усувати відповідно.

Так, наприклад, за останні десятиліття модель хмарних потужностей та обчислень отримала широке застосування в області Інформаційних Технологій (IT). Проте, не дивлячись на свій успіх, впровадження хмарних технологій має подолати декілька проблем, з якими вони зіштовхнулись при появі Інтернету речей (IoT) та

мереж 5-го покоління. В першу чергу, це швидкий ріст кількості пристроїв IoT (такі як сенсори, виконавчі механізми, мобільні телефони та інші прилади доступу), що створюють дуже велику кількість даних, які можуть привести до перевантаження мережі, центрів обробки даних та великих фінансових витрат. По-друге, це велика фізична відстань між пристроями IoT та хмарними центрами обробки даних, що приводить до великих затримок, які можуть бути критичними для деяких чутливих до затримок програмних додатків (наприклад, потокова передача відео високої якості, інтерактивні мобільні ігри, програми доповненої реальності та інші спеціалізовані додатки), які потребують вкрай малої затримки отримання даних від кінцевого пристрою (наприклад, 10 мс або навіть 1 мс). По-третє, додаткам, які розгорнуті в хмарі, тяжко адаптуватися до змін локальних умов (наприклад, точне місцерозташування користувачів та умови роботи локальної мережі) розподілених мобільних пристроїв.

В цілях вирішення цих проблем, пов'язаних з хмарами, нещодавні дослідження представили аналогічну концепцію, яка розширює можливості хмарних розрахунків, які ближче до кінцевих користувачів (тобто на межі мережі) – Mobile Edge Computing or Multi-access Edge Computing (MEC). MEC надає новий рівень розподілених обчислювальних вузлів між пристроями кінцевих користувачів і хмарними центрами обробки даних. Тому додатки, які працюють на MEC, можуть виконувати дії, які близькі до їх користувачів, перед підключенням до хмари.

Таким чином, дана дисертаційна робота спрямована на розробку методів підвищення ефективності розподіленої обробки даних в комп’ютерних системах операторів стільникового зв’язку.

Як було показано, розрахункові потужності на межі стільників – це доволі перспективна концепція в контексті розвитку Інтернету речей, особливо для підтримки залежних від затримки додатків. Головною з основних проблем при цьому є задача по розміщенню релевантних сервісів, яка стосується рішення, в яке ж місце помістити декілька додатків згідно їх вимог до якості надання послуг QoS, це з одного боку, та обчислювальної доступності ресурсу, з іншого боку.

Для цього автором було розроблено метод оптимізації розміщення масштабованих послуг на розподілених обчислювальних ресурсах мережі стільникового оператора, що полягає у послідовному використанні моделі граничних обчислень, узагальненої моделі мережі стільникового оператора та евристичного рішення, заснованого на використанні генетичних алгоритмів.

Даний метод дозволяє зменшити рівень деградація якості обслуговування кінцевих абонентів мережі стільникового оператора, зокрема, затримки на величину до 8 мс для великої кількості абонентів та відповідно великої кількості завдань.

При цьому, ефективне використання ресурсів граничних мобільних обчислень необхідне для гарантування передбачених переваг, які тісно пов'язані з вирішенням наступних завдань:

1. Проблема розвантаження задач, яка полягає у визначенні серверів, на які слід розвантажувати задачі.

2. Проблема розподілу ресурсів додатків, яка визначає обчислювальні ресурси, які повинні бути розподілені для кожної програми, розгорнутої на граничному сервері, щоб обробити всі призначені їм завдання в межах їхніх вимог до затримки.

3. Планування завдань, що визначає порядок, в якому кожне завдання має бути оброблене в спільній програмі, дотримуючись вимог до часу обробки.

Саме ці завдання вирішувались в третьому розділі даної дисертаційної роботи. Крім того, після процесу розвантаження та планування завдань, вирішувалось завдання оптимізації використання обчислювальних ресурсів та енергії в стільникових мережах під час проведення граничних обчислень.

Обробка завантажених завдань з урахуванням їх затримок вимагає прийняття рішення про сервер MEC, на який слід завантажувати кожне із завдань, визначення ресурсів обчислення для розподілу в додатках IoT, які будуть обробляти завдання, окрім визначення порядку в якому вивантажені завдання повинні оброблятися кожним із додатків. Вирішення трьох вищезазначених завдань сильно впливає на прийняття завдань у мережу, оскільки вони безпосередньо впливають на деякі затримки, які вони зазнають.

Тому, саме для вирішення вище окреслених завдань, було удосконалено метод динамічного розвантаження та планування задач для граничних комп'ютерних систем оператора стільникового зв'язку за рахунок формування задачі змішаного цілочисельного програмування та її вирішення за допомогою декомпозиції Бендерса.

Розроблений метод надає змогу максимізувати кількість допущених та відповідно виконаних завдань на розподілених граничних ресурсах мережі стільникового оператора. При цьому швидкість виконання даного планування та розвантаження підвищилась до 10 разів.

Надалі було досліджено управління випромінюваною потужністю радіопередавачів у стільниковій багатокористувацькій системі в умовах впливу інтерференції. В даному випадку, відмінність цієї системи від вільної від завад у тому, що контроль живлення у багатокористувацькій системі в умовах впливу інтерференції головним чином розподілений через взаємодію між різними користувачами мобільних пристроїв, що дуже ускладнює задачу керування потужністю.

Тому було удосконалено метод керування випромінюваною потужністю мобільних пристроїв під час розвантаження завдань в розподіленій комп'ютерній системі граничних обчислень оператора стільникового зв'язку за рахунок послідовного використання моделі для оцінки умови необхідності розвантаження завдань в мобільній мережі та керування випромінюваною потужністю радіопередавальних пристроїв в каналах з інтерференцією на основі теорії ігор.

Даний метод дозволяє зменшити використання енергії під час використання граничних обчислень в комп'ютерних системах операторів стільникового зв'язку на величину від 5% до 40%.

Теоретичні результати, отримані в дисертаційному дослідженні, відкрили можливість виявити і запропонувати нові практичні шляхи підвищення ефективності функціонування підсистеми базових станцій стільникових мереж під час їх впровадження в Україні на основі використання нових методів керування мережею, передавання даних.

При цьому отримані результати дозволяють:

–зменшити рівень затримки в мережі стільникового оператора 5G до 8 мс;

–проводити більш ефективне керування виконанням завдань в мережі стільникового оператора із використанням граничних обчислень (до 12% підвищений рівень енергоефективності);

– запроваджувати нові сервіси для використання в стільниковій мережі.

Практична цінність дисертаційної роботи полягає в наступному:

– розроблено методику розвантаження завдань в мережі стільникового оператора за допомогою концепції граничних обчислень;

– розроблені комп'ютерні моделі проведення розподілених обчислень в комп'ютерних системах операторів стільникового зв'язку;

– розроблено відповідне алгоритмічне забезпечення для планування та розвантаження завдань в граничних комп'ютерних системах операторів стільникового зв'язку.

Матеріали дисертаційної роботи впроваджено у діяльність ТОВ «ІСП Імперіал» та у навчальний процес Центральноукраїнського національного технічного університету.

Всі отримані результати доцільно використовувати під час планування безпроводових мереж стандарту LTE та 5G, розробки обладнання та програмного забезпечення для систем передачі даних, а також в навчальному процесі.

Ключові слова: стільникова мережа, підсистема базових станцій, планування, транспортна мережа, 5G, ефективність, балансування навантаження, MEC.

ABSTRACT

Pavlo Usik. **Methods for improving the efficiency of distributed data processing in computer systems of cellular operators. –** Qualifying scientific work on the rights of the manuscript.

Thesis for the degree of Doctor of Philosophy (PhD) in the specialty 123 «Computer Engineering». – Cherkasy State Technological University, Cherkasy, 2021.

Two major trends are driving the wireless industry to develop fifth-generation cellular networks: the rapid increase in demand for wireless broadband services that require much higher data rates, and much higher capacity networks that can provide video and other resource-intensive services; and Internet of Things (IoT) services, which encourage the need to connect devices en masse, as well as the need for ultra-low latency.

You can define a number of different applications where network data will be used: this includes V2X communication (vehicle communication with each other and with other infrastructure); industrial automation and utility programs; wireless medical services; virtual and augmented reality consumer and business services; some smart city apps; smart homes and a large number of mobile broadband applications.

At the same time, with the development of cellular networks, new and more advanced network architectures for data transmission and management appear. However, there are still a number of unresolved issues and problem areas that need to be addressed and addressed accordingly.

For example, in recent decades, the model of cloud capacity and computing has been widely used in the field of Information Technology (IT). However, despite its success, the introduction of cloud technologies must overcome several of the challenges they faced with the advent of the IoT and 5th generation networks. First and foremost, it is the rapid growth in the number of IoT devices (such as sensors, actuators, mobile phones, and other access devices), which generate very large amounts of data that can lead to network congestion, data centers, and high financial costs. Second, there is a large physical distance between IoT devices and cloud data centers, leading to long delays that can be critical for some delay-sensitive software applications (e.g., high-quality video streaming,

interactive mobile games, augmented reality applications). and other specialized applications) that require extremely little delay in receiving data from the end device (for example, 10 ms or even 1 ms). Third, applications deployed in the cloud find it difficult to adapt to changes in local conditions (such as the exact location of users and LAN conditions) of distributed mobile devices.

To address these cloud-related issues, recent research has introduced a similar concept that extends the capabilities of cloud computing that is closer to end users (ie, at the network boundary) – Mobile Edge Computing or Multi-access Edge Computing (MEC). MEC provides a new level of distributed computing nodes between end-user devices and cloud data centers. Therefore, applications that run on MES can perform actions that are close to their users before connecting to the cloud. Thus, this dissertation is aimed at developing methods to improve the efficiency of distributed data processing in computer systems of cellular operators.

It has been shown that computing power at the cell boundary is a rather promising concept in the context of the development of the Internet of Things, especially to support delay-dependent applications. The main problem is the task of placing relevant services, which concerns the decision in which place to place several applications according to their requirements for the quality of QoS services, on the one hand, and the computational availability of the resource, on the other hand.

To do this, the author developed a method for optimizing the placement of scalable services on the distributed computing resources of the cellular network, which consists in the consistent use of the boundary computation model, a generalized model of the cellular network and a heuristic solution based on genetic algorithms.

This method reduces the level of degradation of the quality of service of end users of the cellular operator's network, in particular, delays of up to 8 ms for a large number of subscribers and, accordingly, a large number of tasks.

At the same time, the efficient use of resources of marginal mobile computing is necessary to guarantee the intended benefits, which are closely related to the solution of the following tasks:

1. The problem of unloading tasks, which is to determine the servers on which to unload tasks.

2. Application resource allocation problem, which determines the computational resources that must be allocated for each program deployed on the marginal server to handle all assigned tasks within their latency requirements.

3. Task planning, which determines the order in which each task must be processed in a joint program, adhering to the requirements for processing time.

These tasks were solved in the third section of this dissertation. In addition, after the process of unloading and planning tasks, the task of optimizing the use of computing resources and energy in cellular networks during the boundary calculations was solved.

Processing downloaded tasks based on their delays requires deciding on the MEC server to load each task, determining computing resources for distribution in IoT applications that will handle the tasks, in addition to determining the order in which the downloaded tasks should be processed by each application. Solving the above three tasks greatly affects the acceptance of tasks into the network, as they directly affect some of the delays they experience.

Therefore, it is to solve the above problems that the method of dynamic unloading and scheduling of tasks for the boundary computer systems of the cellular operator was improved by forming the problem of mixed integer programming and its solution by Benders decomposition.

The developed method makes it possible to maximize the number of allowed and performed tasks on the distributed marginal resources of the cellular operator's network. At the same time, the speed of this planning and unloading has increased up to 10 times.

Further, the power control of the radiated power of radio transmitters in a cellular multiuser system under the influence of interference was investigated. In this case, the difference between this system and free from interference is that the power control in a multi-user system under interference is mainly distributed through the interaction between different users of mobile devices, which greatly complicates the task of power management.

Therefore, the method of controlling the radiated power of mobile devices during the unloading of tasks in the distributed computer system of boundary calculations of the cellular operator was improved by sequential use of the model to assess the need for unloading tasks in the mobile network and control the radiated power of radio devices in interference channels. based on game theory.

This method reduces the energy consumption when using limit computing in the computer systems of cellular operators by 5 to 40%.

The theoretical results obtained in the dissertation research opened the possibility to identify and propose new practical ways to increase the efficiency of the subsystem of cellular base stations during their implementation in Ukraine based on the use of new methods of network management, data transmission.

The results obtained allow:

– reduce the level of delay in the network of the 5G cellular operator 5G to 8 ms;

– to carry out more effective management of performance of tasks in a network of the cellular operator with use of boundary calculations (to 12% the increased level of energy efficiency);

– introduce new services for use in the cellular network;

– improve the quality of service for cellular network subscribers.

The practical value of the dissertation is as follows:

– developed a method of unloading tasks in the network of the cellular operator using the concept of boundary calculations;

– developed computer models for distributed computing in computer systems of cellular operators;

– Appropriate algorithmic software has been developed for scheduling and unloading tasks in borderline computer systems of cellular operators.

The materials of the dissertation were introduced into the activities of ISP Imperial LLC and into the educational process of the Central Ukrainian National Technical University. All the obtained results should be used during the planning of wireless networks of the LTE and 5G standard, development of equipment and software for data transmission systems, as well as in the educational process.

Keywords: cellular network, base station subsystem, planning, transport network, 5G, efficiency, load balancing, MEC.

Список основних публікацій здобувача

1. Usik P., Smirnov O., Odarchenko R., Abakumova A., Kundyz M. QoE assesment technique for media delivery in 5g networks. *Problems of Infocommunications, Science and Technology (PIC S&T)*: 2019 IEEE Int. Sci.-Pract. Conf., (Kyiv, Oct. 8–11, 2019). P. 597–601 (**Scopus**).

2. Usik P., Odarchenko R., Volkov O., Simakhin V., Gospodarchuk O., Burmak Yu. 5G networks cyberincidents monitoring system for drone communications. *Actual Problems of Unmanned Aerial Vehicles Developments (APUAVD*): 2019 IEEE 5th Int. Conf., (Oct. 22–24, 2019). P. 165–169 (**Scopus**).

3. Ponomarenko O., Bulakovskaya A., Skripnichenko A., Usik P., Olenyuk A. Tomographic application-specific integrated circuits for fast radon transformation. *CEUR Workshop Proceedings.* 2020. No. 2654. P. 339–351 (**Scopus**).

4. Усік П.С., Смірнов О.А. Дослідження перспектив використання технологічних рішень в мережах 5g. *Кібербезпека та інформаційні технології*: монографія. Харків: ДІСА ПЛЮС, 2020. С. 122–135.

5. Котелянець В.В., Усік П.С., Кищенко В.В., Гнатюк В.О. Інтелектуалізована система моніторингу параметрів навколишнього середовища на базі технології інтернету речей. *Вісник інженерної академії України.* 2018. № 4. С. 133–140,

6. Усік П.С., Полігенько О.О., Одарченко Р.С., Терещенко Л.Ю., Смірнов О.А. «Інформаційна технологія та програмне забезпечення для підвищення ефективності планування підсистеми базових станцій стільникового зв'язку». *Проблеми телекомунікацій.* 2020 № 1(26). С. 83-96.

7. Усік П.С., Смірнов О.А., Миронець І.В., Буравченко К.О., Якименко Н.М. Метод підвищення ефективності розподіленої обробки даних у комп'ютерних системах операторів стільникового зв'язку. *Вісник Черкаського державного технологічного університету.* 2020. № 4. С. 103–110.

8. Усік П.С., Полігенько О.О., Смірнов О.А. Напрямки підвищення ефективності управління підсистемою базових станцій стільникових операторів. *Проблеми розвитку глобальної системи зв'язку, навігації, спостереження та організації повітряного руху CNS/ATM*: тези доп. наук.-техн. конф., (м. Київ, 21–23 листоп. 2018 р.). Київ: НАУ, 2019. С. 32.

9. Одарченко Р.С., Мараткызы К., Усик П.С. Анализ перспектив использования сетей 5g для автоматизации производственных процессов. *Өндірістегі цифрлық технологиялар конференциясы*: Республикалық ғылыми және практикалық конференциясының жинағы=*Цифровые технологии в промышленности*: материалы респ. науч.-практ. конф.=*Digital technologies in industry*: Materials of sci. and pract. conf. Казахстан, Актау: КГУТИ им. Ш. Есенова, 2019. Каз., рус., англ. С. 42–44.

10. Усік П.С., Смірнов О.А., Якименко Н.М. Перспективи використання мережевих технологічних рішень в 5g. *Інформаційна безпека та інформаційні технології (Information Security and Information Technologies)*: II Міжнар. наук.-практ. конф., (м. Кропивницький, 2–3 квіт. 2020 р.). С. 56.

11. Усік П.С., Смірнов О.А. Підвищення ефективності функціонування підсистеми базових станцій на основі Multi-Access Edge Computing. *Інформаційні технології – 2020 (IT-2020)*: VII Всеукр. наук.-практ. конф. молодих науковців, (м. Київ, 21 трав. 2020 р.). С. 135–136.

12. Chumachenko B.S., Zaitseva N.O., Grigorenko D.K., Usik P.S. Research of the advantages and disadvantages of the network virtualization of network resources of a consistent architecture of 5g networks. *POLIT. Challenges of science today*, (Apr. 1–3, 2020). P. 99–100.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

4G – від англ. Generation – покоління

CN – Ядро мережі (англ. Core Network)

DL – Передача даних в напрямку до абонента (англ. Downlink)

eNB – Покращений вузол B (англ. Evolved Node B)

EPC – Покращене ядро пакетної мережі (англ. Evolved Packet Core Network)

HTTP – Протокол передачі гіпертексту (англ. HyperText Transfer Protocol)

IMT – International Mobile Telecommunications

IoT – Концепція Інтернету речей (англ. Internet of Things)

IP – Протокол міжмережної взаємодії (англ. Internet Protocol)

ISO – Міжнародна організація по стандартизації (англ. International Standard Organization)

LTE – Назва мобільного протоколу передавання даних (англ. Long Term Evolution)

NFV – Віртуалізація мережевих функціій (англ. Network Function Virtualization)

NR – Технологія нового радіоканалу мереж 5G (англ. New Radio)

QoS – Якість обслуговування (англ. Quality of Service)

RAN – Мережа радіодоступу (англ. Radio Access Network)

SAE – Еволюція системної архітектури LTE (англ. System Architecture Evolution)

SDN – Програмно-керовані мережі (англ. Software Defined Networks)

SDR – Програмно-кероване радіо (англ. Software Defined Radio)

TCP – Протокол транспортного рівня мережі OSI (англ. Transmission Control Protocol)

UE – Обладнання користувача (англ. User Equipment)

БС – Базова станція

ІТ – Інформаційні технології

КЗ – Канал зв’язку

МО – Мобільний оператор

МСЕ – Міжнародний союз електрозв’язку

СМО – Система масового обслуговування

ВСТУП

Актуальність теми. Дві основні тенденції спонукають індустрію безпроводового зв'язку розвивати мережі стільникового зв'язку п'ятого покоління: стрімке збільшення попиту на безпроводові широкосмугові послуги, які потребують значно більших швидкостей передачі даних та мережі значно більшої ємності, які можуть надавати відео та інші ресурсоємні послуги; та послуги Інтернету Речей (IoT), що спонукають до необхідності масового підключення пристроїв, а також до необхідності наднадійного зв'язку з наднизькою затримкою.

Можна визначити низку різних областей застосування, де будуть використовуватись дані мережі, а саме: V2X комунікація (комунікація транспортних засобів між собою та з іншими об'єктами інфраструктури); промислова автоматизація та комунальні програми; безпроводові медичні послуги; споживчі та бізнес-послуги віртуальної та доповненої реальності; деякі додатки розумного міста; розумні будинки та велика кількість застосувань мобільного широкосмугового зв'язку.

Виходячи з вищесказаного, можна стверджувати, що розвиток стільникових мереж п'ятого поколінь і підвищення їх ефективності є задачею актуальною та перспективною.

При цьому, із розвитком стільникових мереж з'являються нові більш досконалі мережеві архітектури для передачі даних та керування. Проте залишається ряд невирішених завдань та проблемних місць, які необхідно вирішувати та усувати відповідно.

Так, наприклад, за останні десятиліття модель хмарних потужностей та обчислень отримала широке застосування в області Інформаційних Технологій (IT). Проте, не дивлячись на свій успіх, впровадження хмарних технологій має подолати декілька проблем, з якими вони зіштовхнулись при появі Інтернету речей (IoT) та мереж 5-го покоління. В першу чергу, це швидкий ріст кількості пристроїв IoT (такі як сенсори, виконавчі механізми, мобільні телефони та інші прилади доступу), що

створюють дуже велику кількість даних, які можуть привести до перевантаження мережі, центрів обробки даних та великих фінансових витрат. По-друге, це велика фізична відстань між пристроями IoT та хмарними центрами обробки даних, що приводить до великих затримок, які можуть бути критичними для деяких чутливих до затримок програмних додатків (наприклад, потокова передача відео високої якості, інтерактивні мобільні ігри, програми доповненої реальності та інші спеціалізовані додатки), які потребують вкрай малої затримки отримання даних від кінцевого пристрою (наприклад, 10 мс або навіть 1 мс). По-третє, додаткам, які розгорнуті в хмарі, тяжко адаптуватися до змін локальних умов (наприклад, точне місцерозташування користувачів та умови роботи локальної мережі) розподілених мобільних пристроїв.

З метою вирішення цих проблем, пов'язаних з хмарами, нещодавні дослідження представили аналогічну концепцію, яка розширяє можливості хмарних розразхунків, які ближче до кінцевих користувачів (тобто на межі мережі) – Mobile Edge Computing or Multi-access Edge Computing (MEC). MEC надає новий рівень розподілених обчислювальних вузлів між пристроями кінцевих користувачів і хмарними центрами обробки даних. Тому додатки, які працюють на MEC, можуть виконуватись дії, які близькі до їх користувачів, перед підключенням до хмари. Це значно знижує навантаження на мережу, забезпечує більш оперативну відповідь і надає змогу отримувати локальну контекстну інформацію найбільш ефективним способом.

Останнім часом з'явилось багато наукових праць вітчизняних та здебільшого закордонних дослідників (Климаш М.М., Лемешко О.В., Одарченко Р.С., S. Zhou, Y. Chang, Robert Schober, Albert Banches, Tuyen X. Tran, Luca Foschini та багато інших [1-8], присвячених дослідженням оцінки та підвищення ефективності використання MEC для різних застосувань для підвищення надійності та динамічності використання їх у мережах 5G. Проте дана технологія розподілених обчислень зовсім не позбавлена недоліків, які необхідно нівелювати для її повсюдного розгортання.

Таким чином, розробка методів підвищення ефективності розподіленої обробки даних в комп'ютерних системах операторів стільникового зв'язку є важливою науково-технічною задачею, спрямованою на вдосконалення якості обслуговування абонентів сучасних стільникових мереж і забезпечення вимог до мереж наступних поколінь.

Вищезгадана задача, яка вирішувалась в даній дисертаційній роботі, обумовлює її **актуальність.**

Зв'язок роботи з науковими програмами, планами, темами. Тема дисертаційної роботи та обраний напрямок досліджень безпосередньо пов'язаний з реалізацією положень «Стратегії розвитку інформаційного суспільства в Україні» (затверджена Кабінетом Міністрів України від 15 травня 2013 року) з міжнародними програмами, зокрема, Horizon 2020 (ICT-19-2019 Advanced 5G validation trials across multiple vertical industries, ICT-18-2018: 5G for cooperative, connected and automated mobility, ICT-07-2017: 5G PPP Research and Validation of critical technologies and systems). Основні наукові результати отримано в рамках науково-дослідних робіт: шифр Дельфін, тема «Дослідження можливостей розгортання мереж урядового радіозв'язку на базі концепції 5G», держбюджетної НДР МОН України «Розробка методів підвищення оперативності передачі та захисту інформації у телекомунікаційних системах» (0115U003103), НДР «Розробка методів підвищення безпеки телекомунікаційних мереж» (№ ДР 0112U006630).

Роль автора в зазначених науково-дослідних роботах, у яких дисертант був безпосереднім виконавцем, полягає в аналізі існуючих методів безпроводової передачі інформації, моделюванні та дослідженні роботи розглянутих мереж зв'язку, розробці методів підвищення ефективності функціонування концепції MEC у безпроводових мережах зв'язку.

Мета роботи. Метою дисертаційної роботи є підвищення ефективності розподіленої обробки даних в комп'ютерних системах операторів стільникового зв'язку.

Досягнення поставленої мети передбачає розв’язання таких задач:

1. Розробити метод оптимізації розміщення масштабованих послуг на розподілених обчислювальних ресурсах мережі стільникового оператора.

2. Удосконалити метод динамічного розвантаження та планування задач для граничних комп’ютерних систем оператора стільникового зв’язку.

3. Удосконалити метод керування випромінюваною потужністю мобільних пристроїв під час розвантаження завдань в розподіленій комп’ютерній системі граничних обчислень оператора стільникового зв’язку.

Об'єктом дослідження є процес передавання та обробки даних у комунікаційних мережах комп’ютерних систем операторів стільникового зв’язку.

Предметом дослідження є методи та моделі передавання та обробки даних у комп’ютерних системах операторів стільникового зв’язку.

Методи дослідження. Для досягнення поставлених цілей в дисертаційній роботі використано: методи теорії інформації та передавання сигналів – для аналізу методів передавання інформації у стільникових мережах четвертого та п’ятого поколінь; методи теорії розповсюдження електромагнітних хвиль – для дослідження процесу затухання електромагнітного поля та визначення рівня інтерференції; методи теорії телетрафіку – для генерування та дослідження розподілу навантаження на мережу; методи комп’ютерного моделювання – для перевірки адекватності розроблених моделей та алгоритмів; математичної статистики – для обробки отриманих експериментальним шляхом та під час комп’ютерного моделювання статистичних даних, теорія множин – для опису множин мережевих функцій тощо.

Наукова новизна отриманих результатів. У роботі отримані такі нові наукові результати.

1. Вперше розроблено метод оптимізації розміщення масштабованих послуг на розподілених обчислювальних ресурсах мережі стільникового оператора, що на відміну від відомих, за рахунок використання моделі граничних обчислень, узагальненої моделі мережі стільникового оператора та евристичного рішення, заснованого на використанні генетичних алгоритмів, дозволяє зменшити рівень

деградація якості обслуговування кінцевих абонентів мережі стільникового оператора.

2. Удосконалено метод динамічного розвантаження та планування задач для граничних комп'ютерних систем оператора стільникового зв'язку, який за рахунок формування задачі змішаного цілочисельного програмування та її вирішення за допомогою декомпозиції Бендера, надає змогу максимізувати кількість допущених та відповідно виконаних завдань на розподілених граничних ресурсах мережі стільникового оператора.

3. Удосконалено метод керування випромінюваною потужністю мобільних пристроїв під час розвантаження завдань в розподіленій комп'ютерній системі граничних обчислень оператора стільникового зв'язку, який за рахунок послідовного використання моделі для оцінки умови необхідності розвантаження завдань в мобільній мережі та керування випромінюваною потужністю радіопередавальних пристроїв в каналах з інтерференцією на основі теорії ігор, дозволяє зменшити використання енергії під час використання граничних обчислень в комп'ютерних системах операторів стільникового зв'язку.

Вищенаведені наукові результати дають можливість вирішити проблему підвищення ефективності функціонування стільникових мереж зв'язку.

Практичне значення отриманих результатів. Теоретичні результати, отримані в дисертаційному дослідженні, відкривають можливість виявити і запропонувати нові практичні шляхи підвищення ефективності функціонування підсистеми базових станцій стільникових мереж під час їх впровадження в Україні на основі використання нових методів керування мережею, передавання даних.

При цьому отримані результати дозволяють:

–зменшити рівень затримки в мережі стільникового оператора 5G до 8 мс;

–проводити більш ефективне керування виконанням завдань в мережі стільникового оператора із використанням граничних обчислень (до 12% підвищений рівень енергоефективності);

– запроваджувати нові сервіси для використання в стільниковій мережі.

Практична цінність дисертаційної роботи полягає в наступному:

– розроблено методику розвантаження завдань в мережі стільникового оператора за допомогою концепції граничних обчислень;

– розроблені комп'ютерні моделі проведення розподілених обчислень в комп'ютерних системах операторів стільникового зв'язку;

– розроблено відповідне алгоритмічне забезпечення для планування та розвантаження завдань в граничних комп'ютерних системах операторів стільникового зв'язку.

Матеріали дисертаційної роботи упроваджено у діяльність ТОВ «ІСП Імперіал» та у навчальний процес Центральноукраїнського національного технічного університету. Використання результатів дисертаційної роботи підтверджено відповідними актами впровадження.

Особистий внесок здобувача. Основні положення й результати дисертаційної роботи отримані автором самостійно. З робіт, що опубліковані у співавторстві, використовуються результати, отримані особисто здобувачем. У роботах, опублікованих у співавторстві, автору дисертації належить: вибір параметрів QoS, які найбільше впливають на оцінку QoE [9], розробка архітектури серверів MEC, що дозволило розгортати систему моніторингу ближче до кінцевих користувачів (БПЛА) [10], розробка архітектури системи [11], дослідження перспектив використання технологічних рішень в мережах 5G [12], розробка архітектури системи моніторингу параметрів навколишнього середовища на базі технології інтернету речей [13], розробка технічних рішень по удосконаленню підсистеми базових станцій операторів стільникового зв'язку [14], розробка методу підвищення ефективності розподіленої обробки даних у комп'ютерних системах операторів стільникового зв'язку [15], аналіз основних напрямків підвищення ефективності управління підсистемою базових станцій стільникових операторів [16], аналіз можливості використання мереж 5G для автоматизації виробничих процесів [17], аналіз сучасних технологічних рішень, які можна використовувати в мережі 5G [18], аналіз технології Multi-Access Edge Computing та можливостей її використання для оптимізації роботи підсистеми базових станцій оператора стільникового зв'язку

[19], дослідження переваг використання віртуалізації мережевих ресурсів в мережах 5G [20].

Апробація матеріалів дисертації. Основні теоретичні та практичні результати дисертаційної роботи доповідались і обговорювались на таких конференціях і семінарах: Міжнародна науково-практична конференція «Безпека інформації в інформаційно-телекомунікаційних системах» (Київ, 2018-2020 рр.); VIII Міжнародна науково-технічна конференція "Комп'ютерні системи і мережні технології" (Київ, НАУ, 2019 р.); Міжнародна науково-технічна конференція "ITSEC" (Київ, НАУ, 2019 р.); Міжнародна науково-практична конференція молодих учених і студентів "Політ. Сучасні проблеми науки" (Київ, НАУ, 2019 р., 2020 р.); Всеукраїнська науково-практична конференція «Перспективні напрями захисту інформації» (Одеса, 2019, 2020 р.р.), Автоматика та комп'ютерно-інтегровані технології у промисловості, телекомунікаціях, енергетиці та транспорті: всеукраїнська науково-практична інтернет-конференція (Кропивницький, 2019, 2020 рр.), Перший Міжнародний семінар з кібергігієни і управління конфліктами в глобальних інформаційних мережах (Київ, НАУ, 2019 р.), IEEE International Scientific-Practical Conference «Problems of Infocommunications Science and Technology (PIC S&T)» (Харків, ХНУРЕ, 2019 р.).

Публікації. За матеріалами дисертаційної роботи опубліковано 12 наукових праць, у тому числі: 1 розділ колективної монографії [12], 3 статті у фахових виданнях [13-15], які входять в перелік наукових видань, затверджений МОН України, 1 стаття у періодичних виданнях [11], які включені до науково-метричної бази Scopus, інших 2 праці [9, 10], які включені до науково-меторичної бази Scopus, матеріали доповідей на науково-технічних конференціях – 5 [16-20].

Структура і зміст роботи. Дисертаційна робота складається зі вступу, чотирьох розділів, висновків, списку використаних джерел (вкінці кожного розділу основної частини дисертації) та додатків і має: 119 сторінок основного тексту, 24 рисунки, 10 таблиць, 2 сторінки додатків. Список використаних джерел містить 130 найменування і займає 13 сторінок. Загальний обсяг дисертаційної роботи – 154 сторінки.

Список використаних джерел у Вступі

1. Климаш М.М., Гордійчук-Бублівська О.В., Чайковський І.Б., Урікова О.М. Дослідження ефективності розподілених інфокомунікаційних систем на основі оброблення великих обсягів даних. Вісник Університету «Україна», № 2 (23), 2019. – С. 29-39

2. Mobile network architecture evolution toward 5G / Rost P., Banchs A., Berberana I. et al. *IEEE Communications Magazine.* 2016. Vol. 54, Iss. 5. P. 84–91

3. Maksymyuk, Taras, Mykhailo Klymash, and Minho Jo. "Deployment strategies and standardization perspectives for 5G mobile networks." *2016 13th International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science (TCSET)*. IEEE, 2016.

4. ZHOU, Siyu, et al. The MEC-based architecture design for low-latency and fast hand-off vehicular networking. In: *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*. IEEE, 2018. p. 1-7.

5. DING, Zhiguo, et al. Delay minimization for NOMA-MEC offloading. *IEEE Signal Processing Letters*, 2018, 25.12: 1875-1879.

6. TRAN, Tuyen X., et al. Collaborative mobile edge computing in 5G networks: New paradigms, scenarios, and challenges. *IEEE Communications Magazine*, 2017, 55.4: 54-61.

7. CARELLA, Giuseppe A., et al. Prototyping nfv-based multi-access edge computing in 5G ready networks with open baton. In: *2017 IEEE Conference on Network Softwarization (NetSoft)*. IEEE, 2017. p. 1-4.

8. Tran, T., Navrátil, D., Sanders, P., Hart, J., Odarchenko, R., Barjau, C., ... & Gomez-Barquero, D. (2020). Enabling multicast and broadcast in the 5G core for converged fixed and mobile networks. *IEEE Transactions on broadcasting*, *66*(2), 428-439.

9. Usik P., Smirnov O., Odarchenko R., Abakumova A., Kundyz M. QoE assesment technique for media delivery in 5g networks. *Problems of Infocommunications, Science and Technology (PIC S&T)*: 2019 IEEE Int. Sci.-Pract. Conf., (Kyiv, Oct. 8–11, 2019).

P. 597–601 (**Scopus**).

10. Usik P., Odarchenko R., Volkov O., Simakhin V., Gospodarchuk O., Burmak Yu. 5G networks cyberincidents monitoring system for drone communications. *Actual Problems of Unmanned Aerial Vehicles Developments (APUAVD*): 2019 IEEE 5th Int. Conf., (Oct. 22–24, 2019). P. 165–169 (**Scopus**).

11. Ponomarenko O., Bulakovskaya A., Skripnichenko A., Usik P., Olenyuk A. Tomographic application-specific integrated circuits for fast radon transformation. *CEUR Workshop Proceedings.* 2020. No. 2654. P. 339–351 (**Scopus**).

12. Усік П.С., Смірнов О.А. Дослідження перспектив використання технологічних рішень в мережах 5g. *Кібербезпека та інформаційні технології*: монографія. Харків: ДІСА ПЛЮС, 2020. С. 122–135.

13. Котелянець В.В., Усік П.С., Кищенко В.В., Гнатюк В.О. Інтелектуалізована система моніторингу параметрів навколишнього середовища на базі технології інтернету речей. *Вісник інженерної академії України.* 2018. № 4. С. 133–140,

14. Усік П.С., Полігенько О.О., Одарченко Р.С., Терещенко Л.Ю., Смірнов О.А. «Інформаційна технологія та програмне забезпечення для підвищення ефективності планування підсистеми базових станцій стільникового зв'язку». *Проблеми телекомунікацій.* 2020 № 1(26). С. 83-96.

15. Усік П.С., Смірнов О.А., Миронець І.В., Буравченко К.О., Якименко Н.М. Метод підвищення ефективності розподіленої обробки даних у комп'ютерних системах операторів стільникового зв'язку. *Вісник Черкаського державного технологічного університету.* 2020. № 4. С. 103–110.

16. Усік П.С., Полігенько О.О., Смірнов О.А. Напрямки підвищення ефективності управління підсистемою базових станцій стільникових операторів. *Проблеми розвитку глобальної системи зв'язку, навігації, спостереження та організації повітряного руху CNS/ATM*: тези доп. наук.-техн. конф., (м. Київ, 21–23 листоп. 2018 р.). Київ: НАУ, 2019. С. 32.

17. Одарченко Р.С., Мараткызы К., Усик П.С. Анализ перспектив использования сетей 5g для автоматизации производственных процессов. *Өндірістегі цифрлық технологиялар конференциясы*: Республикалық ғылыми және

практикалық конференциясының жинағы=*Цифровые технологии в промышленности*: материалы респ. науч.-практ. конф.=*Digital technologies in industry*: Materials of sci. and pract. conf. Казахстан, Актау: КГУТИ им. Ш. Есенова, 2019. Каз., рус., англ. С. 42–44.

18. Усік П.С., Смірнов О.А., Якименко Н.М. Перспективи використання мережевих технологічних рішень в 5g. *Інформаційна безпека та інформаційні технології (Information Security and Information Technologies)*: II Міжнар. наук.-практ. конф., (м. Кропивницький, 2–3 квіт. 2020 р.). С. 56.

19. Усік П.С., Смірнов О.А. Підвищення ефективності функціонування підсистеми базових станцій на основі Multi-Access Edge Computing. *Інформаційні технології – 2020 (IT-2020)*: VII Всеукр. наук.-практ. конф. молодих науковців, (м. Київ, 21 трав. 2020 р.). С. 135–136.

20. Chumachenko B.S., Zaitseva N.O., Grigorenko D.K., Usik P.S. Research of the advantages and disadvantages of the network virtualization of network resources of a consistent architecture of 5g networks. *POLIT. Challenges of science today*, (Apr. 1–3, 2020). P. 99–100.

РОЗДІЛ 1
ДОСЛІДЖЕННЯ ПЕРСПЕКТИВ ВИКОРИСТАННЯ КОНЦЕПЦІЇ MEC В МЕРЕЖАХ 5G

Стільникові мережі зв’язку стали одними із найбільш поширених телекомунікаційних мереж. Кожен із нас вже не зможе уявити свого життя без доступу до мережі. При цьому дані мережі продовжують розвиватись стрімкими темпами. Так, вже стандартизовано мережі 5G та вже відбувається запуск перших комерційних мереж.

Поряд із розвитком телекомунікаційних технологій збільшуються й об’єми даних, які передаються, вимоги до якості надання послуг тощо. Все це спонукає до запровадження нових прогресивних рішень. Тому метою даного розділу є аналіз сучасного стану розвитку стільникових мереж, аналіз нових технологічних рішень, визначення їх недоліків та формулювання задач дисертаційного дослідження.

1.1 Дослідження основних драйверів та вимог до мереж 5G

1.1.1. Існуючі системи зв’язку та їх недоліки

Мережі мобільного радіозв’язку спочатку були призначені для передачі мови з використанням аналогових каналів для передачі. Потім, з поступовим розвитком цифрових технологій, в умовах виникнення потреби у значно більшій кількості каналів, в 90-х роках двадцятого сторіччя з'явилися цифрові системи другого покоління – 2G (рис. 1.1) [1]. З’явились принципово нові послуги, такі як обмін текстовими повідомленнями (SMS) і одночасний доступ до даних за технологією комутації каналів.

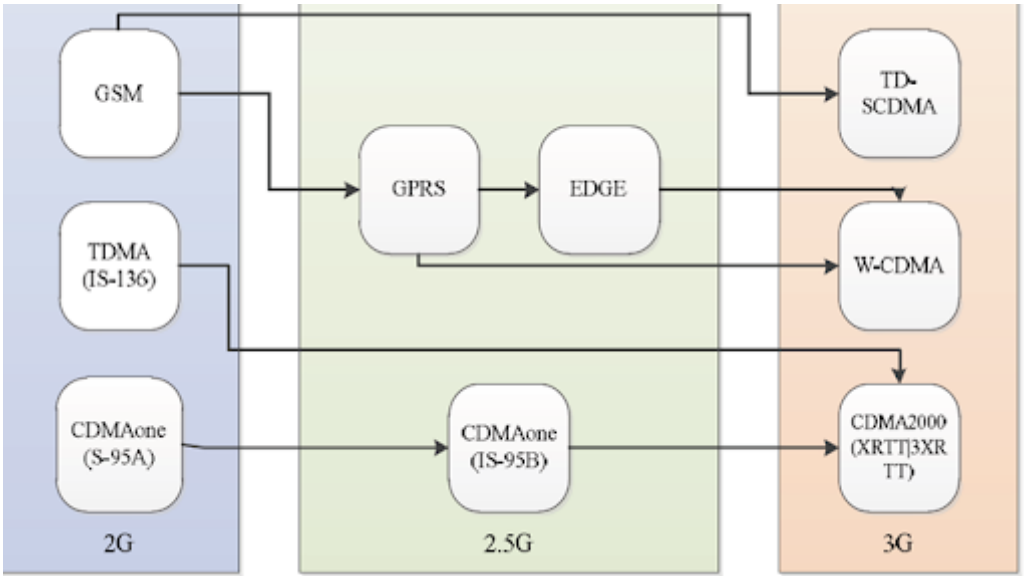


Рис. 1.1. Еволюція стільникових мереж зв’язку

Послуги низької швидкості передачі даних, які забезпечували системи 2G були вже зовсім не в змозі забезпечити значний попит на доступ до ресурсів мережі Інтернет. Це призвело до попиту на нові стандарти 3G, які еволюціонували, щоб забезпечити швидке обслуговування даних і більше можливостей для передачі голосу [2]. Недавня (4G) система стільникового зв'язку LTE була розроблена, щоб забезпечити високу пропускну здатність і сервіс високої швидкості передачі даних для стільникових мультимедіа. Якщо дивитися з історичної точки зору, кожен із стандартів стільникового зв'язку розвивався навколо набору ключових випадків використання [3, 4]:

– Мережі першого покоління (1G) – основне призначення: голосові послуги.

– Мережі другого покоління (2G) – основне призначення: поліпшення голосового зв'язку та обміну текстовими повідомленнями.

– Мережі третього покоління (3G) – основне призначення: вбудовані голосові послуги, а також доступний Інтернет із використанням стільникових мереж.

– Мережі четвертого покоління (4G) – основне призначення: забезпечення високої пропускної здатності для стільникового мультимедіа.

На рис.1.2 показана еволюція систем стільникового зв'язку, на основі нових комунікаційних потреб. Усі очікують на наднизьку затримку, а також збільшення

швидкості передачі даних. Це основні напрямки наукових досліджень в області сучасних стільникових мереж.

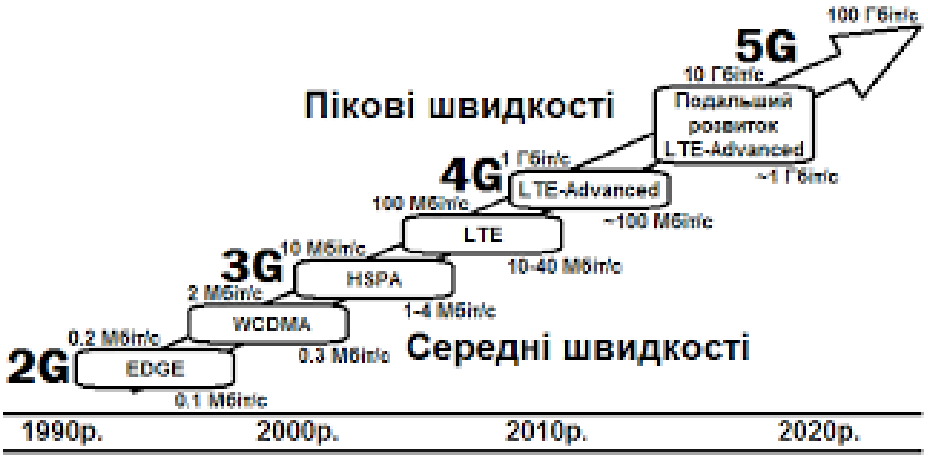


Рис. 1.2. Еволюція мобільних мереж на основі потреб користувачів

5G має надати можливість досягнути досить низького рівня затримки, а радіоінтерфейс, при цьому, не стане вузьким місцем, навіть для найскладніших випадків використання (Use Cases) [5]. Нові мережі стільникового зв’язку очевидно не будуть використовуватися виключно для спілкування людей. Натомість, прогнозується стрімке зростання попиту на використання зв'язку з пристроями, котрі відносяться до категорії «Інтернет речей». Крім цього, пристрої будуть не тільки дистанційно керуватися і управлятися людьми, але й взаємодіяти один з одним (M2M). Тому будуть вимагатись більш надійні канали зв'язку, що зможуть забезпечити наднизькі затримки в передачі [6].

Взаємодія з людиною також буде більш вимогливою в майбутньому – для системи 2G, основний акцент був зроблений на голос, де вимоги до затримок були викликані чутними обмеженнями людини, в порядку 100 мілісекунд. Для мультимедійних додатків, людське око більш чутливе і необхідні затримки менше 10 мілісекунд. «Тактильна взаємодія» означає все більш широке використання сенсорних інтерфейсів, де іноді можна спостерігати вимоги до затримки менше однієї мілісекунди.

"Досвід Gigabit" означатиме прийом даних і швидкість передачі гігабіт на секунду для користувачів і пристроїв. Знову ж таки, це не означає впровадження мереж з високою пропускною здатністю у всьому світі, але центри великих міст стануть першими місцями, де буде відчуватися потреба в новій системі. Загальне зростання попиту на швидкості передачі даних користувачів і пропускну здатність мережі як і раніше є основною рушійною силою технологічного розвитку – підвищення потужності мереж потребують більш високої продуктивності, стільникового ущільнення і доступу до нових, більш широких носіїв в новому спектрі. Частина зростання виробничих потужностей, звичайно, можуть бути виконані з існуючими системами, але у 2021 року, межі вже досягнуті і постала необхідність у технології 5G [7].

1.1.2. Рушійні сили розвитку мереж 5G

5G – це п’яте покоління стільникових мереж, що пропонує нові можливості, які створять можливості для людей, бізнесу та суспільства [7, 8].

5G зробить набагато більше, ніж просто значно покращить ваше мережеве з’єднання. Воно відкриває нові можливості, дозволяючи нам пропонувати новаторські рішення, які охоплюють все суспільство.

Уявіть, як мільярди підключених пристроїв збирають та обмінюються інформацією в режимі реального часу для зменшення дорожньо-транспортних пригод; або рятувальні програми, які можуть полетіти завдяки гарантованим з'єднанням без затримок; або виробничі лінії настільки передбачувальні, що можуть запобігти перебоям задовго до їх виникнення. 5G буде поєднувати в собі існуючі технології радіодоступу з новими, оптимізованими для конкретних смуг частот і розгортання мережі, сценаріїв і варіантів використання. 5G буде використовувати принципово нову мережеву архітектуру, засновану на технологіях Network Function Virtualization (NFV) і Software Defined Networking (SDN). При цьому, слід відзначити, що програмованість матиме ключове значення для досягнення високого рівня гнучкості, так як стільниковим операторам потрібно буде підтримувати нові

сервіси зв’язку, що висуватимуться до них з широкого кола користувачів, пристроїв, компаній з різних галузей промисловості та інших організацій. Мережі 5G повинні бути програмованими, програмно забезпеченими і управлятися цілісно, щоб забезпечити різноманітний і вигідний спектр послуг.

Багато постачальників послуг стикаються як з цифровою трансформацією, так і з переходом з 4G на 5G одночасно. Тривають глобальні трансформаційні ініціативи. У міру того як формуються випадки використання послуг 5G, нові вимоги до BSS та OSS стають очевидними.

Багато з цих нових вимог стосуються як цифрової трансформації, так і 5G [8]:

– Надання та забезпечення послуг у реальному часі.

– Поліпшення BSS для декількох доменів та декількох партнерів.

– Високо масштабована, розподілена оркестрація в режимі реального часу.

– Управління QoS в режимі реального часу із закритим циклом та гарантованим аналітичним сервісом.

– Наскрізне управління гібридними віртуальними та фізичними мережами через доступ, агрегацію та основну інфраструктуру, включаючи 4G, 5G та фіксовані мережі.

– Нарізання та спільне використання мережі в режимі реального часу через бездротову та фіксовану інфраструктуру [9].

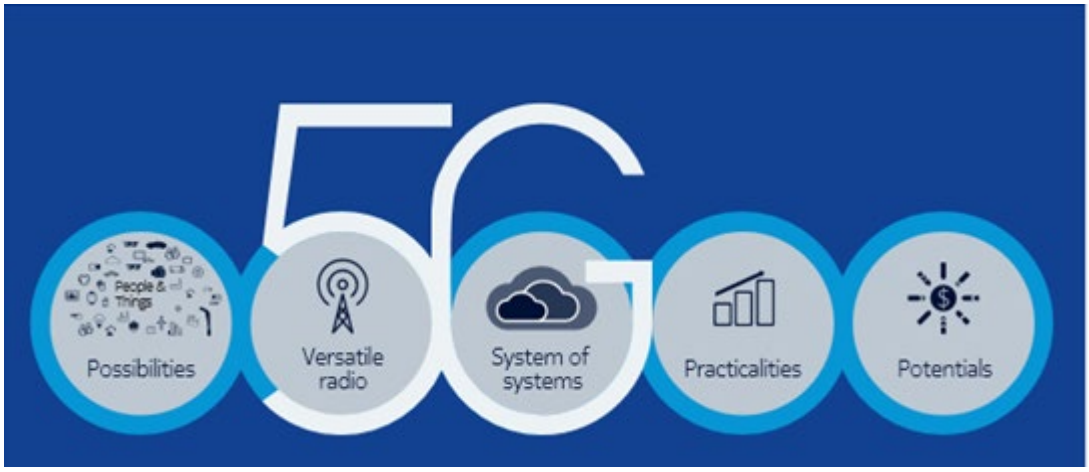


Рис. 1.3. Ключові рушійні сили 5G [10]

Найбільша різниця між 5G і успадкованими проектними вимогами є різноманітність сценаріїв використання, котрі мережі 5G повинні підтримувати в порівнянні з сучасними мережами, які були розроблені в першу чергу для забезпечення високошвидкісного широкосмугового стільникового зв'язку. Проте, 5G буде для людей і речей, які можуть бути широко розділені на три категорії використання (рис. 1.4) [11]:

1. Розширений промисловий IoT. Виробники вже широко використовують датчики IoT для моніторингу продуктивності та оптимізації як виробництва, так і логістики. Менша затримка та підвищена гнучкість бездротового зв'язку дозволять їм подальше впорядковувати свою інфраструктуру, будувати взаємопов’язані та напівавтоматизовані інтелектуальні заводи та збільшувати видимість у всіх своїх ланцюгах поставок.

2. Більше даних у реальному часі для кращих рішень. Будь то роздрібне середовище, електромережа чи заклад охорони здоров’я, 5G може надавати надійні дані в режимі реального часу, які можуть допомогти організації прийняти кращі рішення. Оскільки мережа 5G може передавати стільки інформації так швидко, мережа 5G може збирати та обробляти дані з кількох джерел, щоб люди могли фактично бачити та вирішувати потенційні проблеми, коли вони трапляються, а не намагатися визначити, що пішло не так. Це дозволяє швидко оптимізувати та динамічно приймати рішення, що відображає реальну ситуацію на місцях.

3**. Автономні транспортні засоби.** Самостійних автомобілів, можливо, поки що тут немає у великій кількості, але відсутність потужної мережі 5G є однією з причин, чому вони не є більш звичним видовищем на дорозі. Оскільки мережі 5G особливо ефективні при передачі даних між рухомими об'єктами, це буде вкрай важливо для успіху автономних мереж транспортних засобів, яким потрібно буде передавати величезну кількість інформації між транспортними засобами. Інші потреби в підключенні, такі як дистанційна діагностика, оновлення операційної системи, прогнозне обслуговування, оплата автомобілів та управління парком, важко масштабувати без потужності мереж 5G.

4. Додатки Smart City. Ще одним захоплюючим додатком для 5G є технологія розумного міста. Міста у всьому світі експериментують із цифровими рішеннями, які можуть мінімізувати затори на дорогах, покращити безпеку та зробити державні служби більш ефективними. Розумні датчики IoT можуть потенційно швидко передавати дані через мережі 5G, щоб попередити міських чиновників про проблеми, повідомити пасажирів про стан дорожнього руху або навіть повідомити людей про вільні місця для паркування.

5. Покращення мереж охорони здоров'я. Мало яка з галузей промисловості більше схильна до техногенних зривів, ніж сектор охорони здоров'я. Обмеження даних давно створюють перешкоду для взаємодії медичних служб, але мережі 5G можуть зробити можливим швидке та легке передавання зображень з високою роздільною здатністю між постачальниками послуг. Збільшена пропускна здатність мереж 5G буде особливо корисною для лікарень, що дозволить їм розширити використання та гнучкість підключених пристроїв IoT без шкоди для продуктивності. Послуги телемедицини також можна зробити швидшими та надійнішими завдяки підключенню 5G, особливо коли мова йде про поширення цих послуг на сільські райони та інші віддалені місця. Замість розпилення, зернистих відеоз'єднань, телемедицина 5G може забезпечити високоякісне потокове передавання відео для поліпшення взаємодії пацієнта та лікаря.

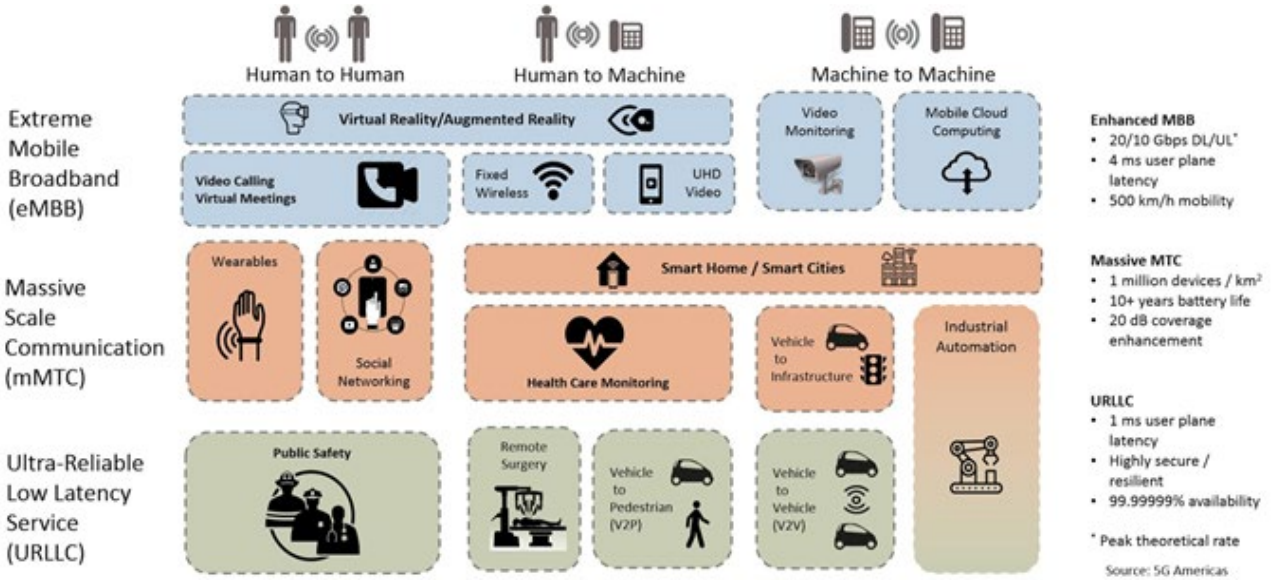


Рис. 1.4. Випадки використання 5G

Нове 5-е покоління стільникових пристроїв буде побудовано навколо двох основних принципів проектування, якими керуються всі вимоги і технічні рішення [12].

Ключовим принципом проектування мереж 5G є **гнучкість та динамічність,** для підтримання невідомих раніше випадків використання, які неминуче виникнуть в майбутньому. Декілька факторів сприяють динамічності мережі – здатність мережі швидко пристосовуватися до мінливого попиту. Впровадження радіотехнологій 5G та запуск нових послуг є двома головними факторами, що підштовхують до необхідності більш динамічних мереж і, отже, до необхідності більш гнучкого транспортного літака. Однак існує багато інших факторів, що сприяють динамічності мережі [13]:

– **динамічність ресурсів**: швидке додавання та усунення ресурсів зв'язку, обчислень та зберігання;

– **динамічність трафіку:** реагування на коливання шаблонів трафіку, що виникають внаслідок переміщення / міграції користувачів або варіацій активності користувачів;

– **динамічність послуг:** реагування на схеми використання послуг із дуже різними потребами в ресурсах;

– **збої та сервісні вікна:** можливість перенаправити трафік та мінімізувати вплив простою;

– погодні умови: управління впливом дощу або туману на ефективність роботи мікрохвильових або вільно-космічних оптомереж.

Думки експертів різняться щодо того, як розвиватимуться архітектури доступу та транспорту для задоволення майбутніх вимог до мобільних пристроїв, і зокрема, як вони будуть забезпечувати підтримку малих стільників [14-16]. Спадкові мережі, як правило, складаються з окремих філій для фіксованого (житлові / ділові послуги) та мобільного доступу. Постійне ущільнення мобільних мереж, ймовірно, призведе до використання декількох різних дрібноклітинних транспортних технологій – кожна з яких адаптована до конкретних умов мережі.

Зокрема, все більш розповсюдженим буде прийняття технологій бездротового зворотного / фронтального зв'язку, таких як NLOS, для забезпечення підключення до нових дрібноклітинних сайтів. У той же час інфраструктура фіксованого доступу та її широка доступність будуть і надалі корисними для забезпечення підключення до малих стільникових мереж, підштовхуючи до необхідності зближення фіксованої та мобільної мереж.

1.1.3. Вимоги до архітектури мережі 5G

Нова мережева архітектура буде мати важливе значення для задоволення потреб в період після 2020, для керування складними багатошаровими та мульти-технологічними мережами, а також для досягнення вбудованої гнучкості [13]. Ера мереж 5G буде програмованою, з керуванням програмного забезпеченням і цілісним управлінням.

Існуюча архітектура має кілька архітектурних проблем. Вони пов’язані із необхідністю забезпечення наступних вимог [17].

Збільшена швидкість передачі даних. Збільшення пропускної здатності ставить перед архітектурними проблемами як радіодоступ, так і сегменти базової мережі. Поки що один з життєздатних підходів до збільшення пікової швидкості передачі даних накопичується, кілька провайдерів разом і використовують їх одночасно для відправки або отримування даних, що належать до одного потоку.

Зверніть увагу, що це найкращий пік швидкості передачі даних, який дуже різниться від швидкості передачі даних в Гбіт / с, необхідної для 5G. Тому що це практично неможливо досягти на основі наявності частот.

На додаток до проблем мережі радіодоступу, основна мережу також потрібно переробити та оптимізувати (наприклад, видалити будь-які потенційно вузькі місця для передачі трафіку), оскільки вузли мережі LTE-SAE в даний час структуровані в ієрархічному порядку, де трафік проходить через серію обслуговуючих шлюзів (S-GW) і мережеві шлюзи пакетної передачі даних (P-GW). Для більш ефективної обробки трафіку на площині даних мережа повинна стати більш рівною і більше

розподілятися, щоб уникнути зайвого трафіку користувача, досягаючи мережевих вузлів, розташованих віддалено в центрі мережі.

Знижена наскрізна затримка. Сьогодні для LTE це зазвичай займає кілька десятків мілісекунд для IP-пакету, надісланого із програми, що працює в пристрої для переходу вгору і вниз по стеку протоколів LTE, перш ніж нарешті будуть доставлені необхідні дані.

Масовий зв'язок. Існуючі мобільні мережі LTE-SAE розроблені та оптимізовані в основному для даних користувачів. Тобто система є більш ефективною в обробці трафіку площини даних, ніж трафіку площини управління.

Останнім часом і в найближчі роки інша схема руху є і надалі спостерігатиметься для зв'язку машинного типу.

Ці пристрої та речі роблять дуже частими та епізодичними спроби передати невеликий обсяг даних, що призводить до величезного обсягу загального та одночасного сигналу руху на площині управління та малої кількості загального трафіку користувача.

Гарантована якість досвіду. Останнім часом докладено багато зусиль для створення гнучкої і програмованої мобільної мережі. Водночас, техніки щодо збору величезної кількості мережевих даних та аналізу зібраних даних в режимі реального часу, також дозрівають. Беручи повні переваги гнучких та програмованих мереж, стає можливим налаштувати та оптимізувати мережеву продуктивність на вимогу в режимі реального часу.

Вища доступність. Віртуалізація та декомпозиція основних програмних функцій з основним обладнанням та завуальованим середовищем дає кілька переваг. Однією з головних переваг є висока доступність, де функцію можна мігрувати та гнучко відтворюватися на інших фізичних ресурсах.

Більш висока ефективність. Очікується, що мережа 5G забезпечить вищу швидкість передачі даних у будь-якому місці, що врешті-решт призводить до вищих загальних витрат на розгортання.

Крім того, послуги 5G та варіанти використання є надзвичайно жорсткими до технічних вимог, які потенційно можуть підвищити вартість окремих функцій

мережі, що розгортаються. Тому індивідуальне функціонування мережі та загальна інфраструктура повинні бути високовитратоефективними.

Таким чином, очікується зміна мережевої архітектури (рис. 1.5) [18].

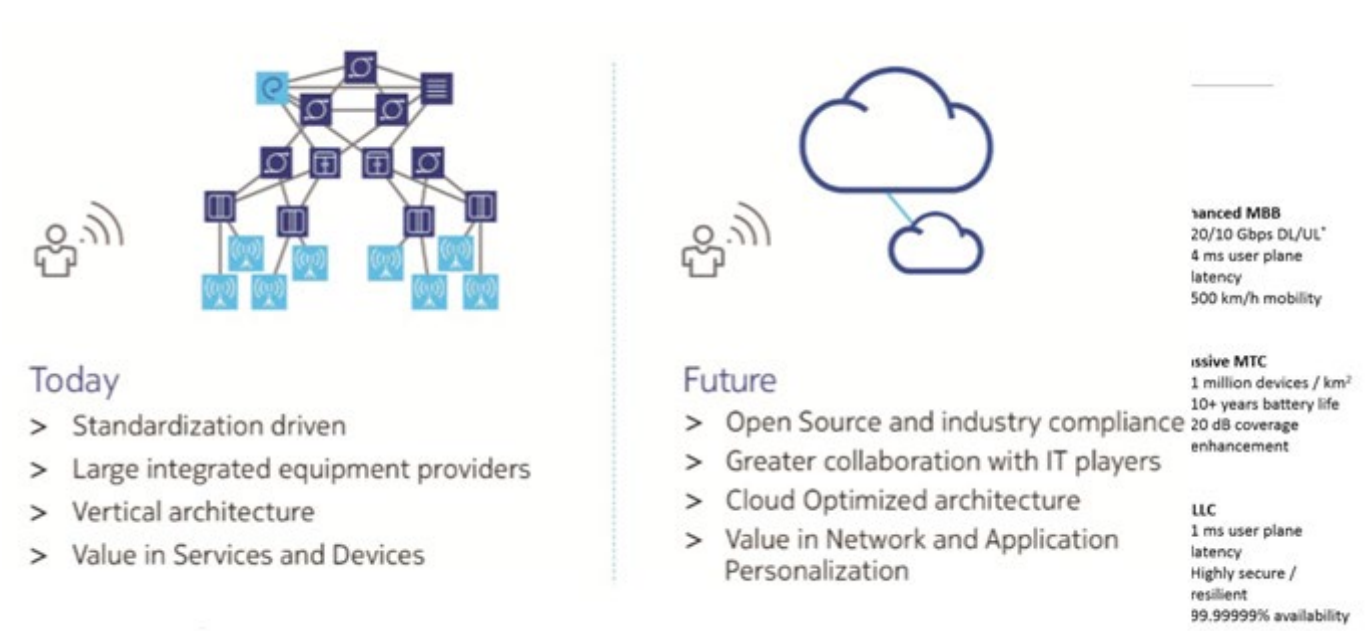


Рис. 1.5. Еволюція мережевої архітектури 5G

Останні вимоги епохи мереж 5G в 2020 році і за її межами сформували архітектурне бачення майбутньої мережевої архітектури “Cognitive and cloud Optimized Network Evolution” (CONE) [18].

CONE (рис. 1.6) охоплює фундаментальні зміни у восьми мережевих доменах, як мобільних, так і фіксованих, що поєднує топологічний вигляд радіодоступу та основних мереж з функціональним видом архітектури функцій мережі. Архітектура, показана на рис. 1.6, відображає мережі мобільного доступу для стислості [19].

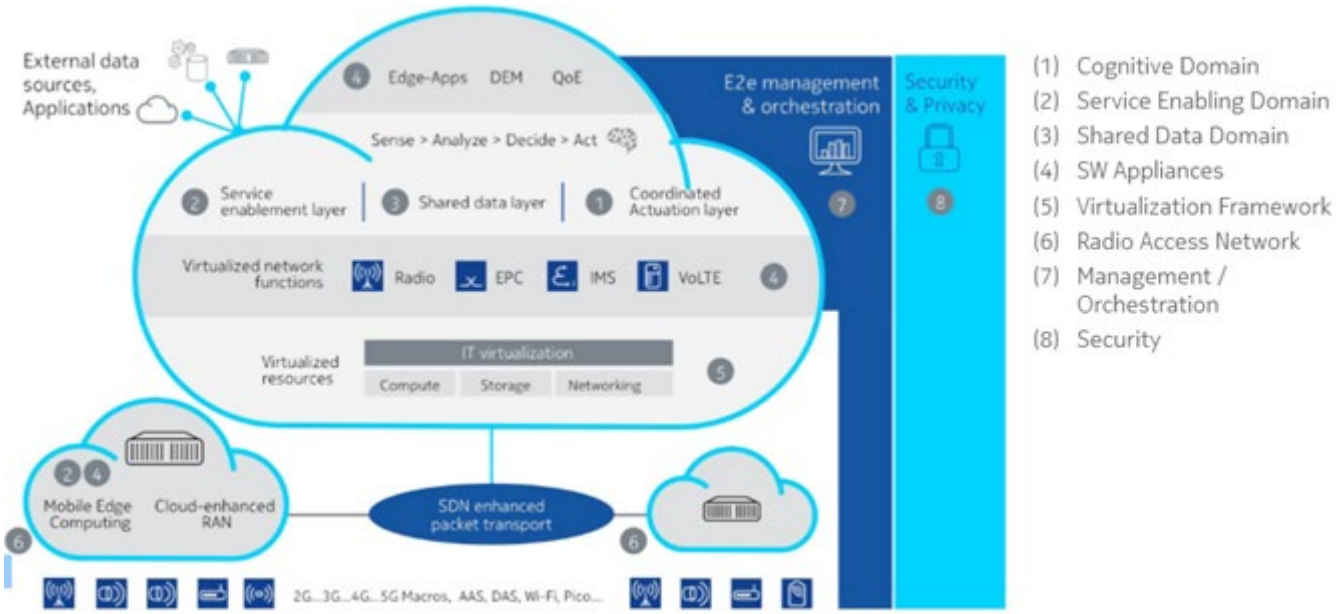


Рис. 1.6. Мережева архітектура CONE

Когнітивний домен. Когнітивний домен збирає дані та події з мережі та, за необхідності, із зовнішніх джерел, таких як соціальні мережі. Ці дані можна обробляти в режимі реального часу для отримання відповідної інформації, а також зберігати в автономному режимі для подальшої обробки. Програми отримують доступ до даних через домен спільних даних. Аналіз даних виявляє та інтерпретує різні закономірності, наприклад, щоб знайти першопричину проблем та виявити будь-які наслідки для клієнтів або бізнесу оператора. Статистичні дані цього аналізу повідомляються та автоматично перекладаються у відповідні дії. Вони виконуються координованим доменом активації для запобігання конфліктам між діями, ініційованими різними програмами.

Домен надання послуг. Використовуючи домен активації послуг (SED), оператори можуть дозволити контрольований та захищений доступ до своїх мереж уповноваженим третім сторонам, дозволяючи їм розгортати інноваційні, пізнавальні програми та послуги для мобільних споживачів, підприємств та вертикальних сегментів. (SDK) для сприяння розробці, забезпеченню, тестуванню, звітуванню, аналітиці та інтеграції в платформу. Наприклад, можливість пропускної здатності в Mobile Edge Computing вже забезпечує керований доступ до мережі для будь-якого типу додатків, щоб забезпечити найкращу продуктивність.

Спільний домен даних. Спільний домен даних (SDD) діє як загальне сховище даних, що забезпечує спільний доступ до даних для програм із різних доменів операторів. Метою є усунення специфічних для додатків силосів даних, коли це можливо, та забезпечення послідовного набору ключових показників ефективності (KPI). Окрім гнучкого доступу до даних, SDD також має механізми безпеки та конфіденційності, що забезпечують доступ до захищених даних лише авторизованим програмам. Автентифікація та авторизація, а також обмеження доступу людей залежно від їхніх ролей забезпечують безпеку даних та конфіденційність.

Програмне забезпечення. В архітектурі CONE всі фізичні ресурси, необхідні для реалізації будь-якого елемента мережі, віртуалізовані та пропонуються «як послуга», доступна через менеджер інфраструктури. Усі мережеві функції та послуги побудовані з програмного забезпечення, яке може бути сховищем абонентів, голосовим сервером, LTE-SW, брандмауером на транспортному рівні IP, функціональністю SON та багатьма іншими. Наприклад, Nokia впровадила елементи EPC, Voice over LTE та інші послуги зв'язку у повністю віртуалізованому середовищі. Віртуалізація є ключовим фактором, що сприяє архітектурі бездротових мереж наступного покоління. Бачення полягає в максимальному використанні платформ обробки загального призначення для розміщення різноманітних мережевих функцій від Core до RAN у вигляді функцій віртуальної мережі (VNF).

Структура віртуалізації. Домен віртуалізації включає площину виконання та площину автоматизації. Площина виконання складається з набору ресурсів (обчислювальних, мережевих, сховищних), до яких можуть отримати доступ мережеві функції через рівень віртуалізації. Площина автоматизації забезпечує автоматичне управління площиною виконання за допомогою таких операцій, як створення, видалення та масштабування функцій мережі та розподіл базових ресурсів.

Мережа радіодоступу. Незважаючи на те, що RAN залишається ключовим фактором повсюдного бездротового підключення, протягом найближчих років очікуються значні зміни. Нове 5G радіо інтегрує існуючі та нові технології як

доповнення до LTE. 5G включатиме існуючі системи, такі як LTE-Advanced та Wi-Fi, а також революційні технології для надщільного розгортання, зв'язку машинного типу, високонадійного зв'язку та мінімальних затримок. Важливо звести до мінімуму кількість нових ефірних інтерфейсів, щоб забезпечити ідеальну взаємодію нових радіостанцій між собою та існуючими технологіями. Для задоволення значно збільшеної пропускної здатності та надзвичайно низьких вимог до затримок потрібні дуже щільні мережі на додаток до більшого спектру. Однак економічно недоцільно будувати надщільні мережі скрізь, і низька затримка та / або гігабітний зв’язок знадобляться лише у певних випадках. Масивні системи MIMO використовуватимуть новий і рясний спектр в смугах см-хвилі та мм-хвилі. Тим часом нові програми, що вимагають затримки в мілісекунді або менше, вимагатимуть обчислювальної потужності, розміщеної поруч із користувачем. Це перетворить існуючі макросайти на невеликі центри обробки даних, конфігурація стала практичною завдяки використанню спектра мм-хвилі та см-хвилі для швидкого транспорту. Незважаючи на те, що обчислювальні ресурси наближені до радіо, функції управління та координації працюють більш централізовано, щоб забезпечити безперебійну інтеграцію з широкосмуговою мережею. Nokia Networks вже показала прототипи для дизайну радіосигналу 5G, такі як використання хвилі мм і хвилі відстеження спектра та променя мобільних користувачів. Управління радіоресурсами Nokia Networks в режимі реального часу для систем 5G продемонструвало безперебійне зв’язок між 4G і 5G і показує вдосконалений погляд на те, як 4G можна інтегрувати в 5G.

Оркестрація та управління. У повністю хмарній мережі існує кілька рівнів та типів оркестровки. Сюди входить організація послуг для послуг, розроблених оператором і пропонованих користувачам; управління хмарними послугами та доступними ресурсами, що здійснюється домену віртуальної інфраструктури менеджера. І, як абсолютно новий об’єкт, мережевий функціональний оркестратор запроваджений та визначений ETSI NFV.

Безпека та конфіденційність. Вкрай важливо вжити заходів для захисту мережі від загроз. Ці загрози потрібно швидко виявляти, швидко діяти, щоб

зменшити їхній ефект. Для такого швидкого реагування потрібна багатовимірна, цілісна та цілісна архітектура, яка “поєднує крапки” між безпекою та конфіденційністю та мережевими подіями. Обмін інформацією про загрози, порушення та пов'язані з ними рішення є критично важливим не лише з клієнтами та екосистемними партнерами, але й з регуляторами та потенційно з конкурентами. Ця відкритість необхідна, оскільки всі домени зазнають впливу кіберзагроз. Основними принципами, на яких базується архітектура безпеки та конфіденційності, є постійна пильність; підвищена автоматизація збору, аналізу та реагування на дані; та оцінка загроз поза межами мережі.

1.2. Дослідження ключових технологій для забезпечення вимог 5G

Принципи проектування нових комунікаційних мереж та мереж 5G необхідні для задоволення вимог та для надання політикам рішень ключових питань. У цьому розділі представлені принципи проектування систем 5G для задоволення вимог та ролі 5G [17, 20]. Це забезпечує читачеві загальну картину всієї системи 5G. У цій главі розглядаються нові підходи до проектування та ключові проблеми проектування системи 5G. Він розглядає нову радіостанцію 5G, яка була визначена у стандарті Проекту партнерства третього покоління. 5G NR може застосовувати різні фази, конфігурації та сценарії. Стільникові мережі будуть поступово розвиватися, і старе та нове мережеве обладнання будуть співіснувати протягом певного періоду. У цьому розділі також розглядаються ключові технічні засоби 5G. Ці методи покращують ключові показники ефективності 5G і дозволяють системі 5G відповідати високому рівню вимог.

Взагалі ключові принципи архітектури мережі 5G полягають в наступному [21]:

– Поділ мережевих вузлів на елементи, що забезпечують роботу протоколів User Plane і елементи, що забезпечують роботу протоколів Control Plane. Це значно збільшує гнучкість в частині масштабування і розгортання (допускаючи централізоване і децентралізоване розміщення окремих складових мережевих вузлів).

– Поділ на мережеві шари (Network Slicing), ґрунтуючись на послугах та вимогах до них, що надаються конкретним групам кінцевих користувачів.

– Віртуалізація мережевих функцій – VNF (Virtual Network Functions).

– Підтримка паралельного доступу до централізованих і децентралізованих локальних служб, що дозволяє реалізовувати концепції хмарних (fog computing) і прикордонних (edge computing) обчислень.

– Конвергентна архітектура, що об'єднує різні типи мереж доступу (AN – Access Network) – 3GPP (New Radio – NR) і не 3GPP (WiFi та ін.) з використанням єдиної опорної мережі (CN – Core Network).

– Використання єдиних алгоритмів і процедур автентифікації для різних мереж доступу.

– Підтримка мережевих функцій stateless, де обчислювальний ресурс відділений від ресурсу зберігання.

– Підтримка роумінгу.

Мережеві функції 5G взаємодіють наступним чином [22]:

– одні мережеві функції (наприклад, AMF) надають змогу іншим авторизованим мережним функціям отримувати доступ до їх сервісів;

– взаємодія точка-точка (наприклад, інтерфейс N11), що може відбуватись між будь-якими двома мережевими функціями ядра мережі (наприклад, AMF і SMF).

При цьому, всі мережеві функції площини управління мають користуватись тільки сервісно-орієнтованими інтерфейсами для взаємодії між собою.

Взагалі, архітектура ядра мережі 5G включає в себе наступні основні мережеві функції (NF) [22]:

– функція управління доступом і мобільністю (AMF – Access and Mobility Management Function);

– функція управління сесіями (SMF – Session Management Function);

– функція передачі даних користувачів (UPF – User Plane Function);

– модуль керування даними користувачів (UDM – Unified Data Management);

– уніфікована база даних (UDR – Unified Data Repository);

– система зберігання неструктурованих даних (UDSF – Unstructured Data Storage Function);

– функція вибору мережевого шару (NSSF – Network Slice Selection Function);

– функція управління політиками (PCF – Policy Control Function);

– функція забезпечення взаємодії з зовнішніми додатками (NEF – Network Exposure Function);

– репозитарій мережевих функцій (NRF – NF Repository Function);

– функція додатків (AF – Application Function);

– функція підтримки обміну короткими текстовими повідомленнями за допомогою протоколу NAS (SMSF – SMS Function);

– функція взаємодії з не-3GPP мережами доступу (N3IWF – Non-3GPP InterWorking Function);

Для забезпечення вимог до мереж 5G в розглянутій вище архітектурі необхідно буде використовувати принципово нові технології, такі, наприклад, як SDN, NFV, SDR, MEC та інші. Розглянемо їх більш детально в наступному підрозділі.

1.2.1. Концепція SDN

Програмно-конфігурована мережа (SDN) – це нова архітектура, яка є динамічною, керованою, економічно ефективною та адаптується, що робить її ідеальною для сучасних застосувань з високою пропускною спроможністю. Ця архітектура роз'єднує функції управління мережею та переадресації, дозволяючи керуванню мережею стати безпосередньо програмованим, та пропонує базову інфраструктуру, яку потрібно абстрагувати для програм та мережевих служб.

SDN засноване на використанні наступних принципів [23]:

– розділення площини управління (control plane) та площини передачі даних (data plane). В мережах SDN відбувається розділення процесів передачі інформаційного трафіку та трафіку управління;

– уніфікований, єдиний, незалежний від вендорів інтерфейс взаємодії між рівнем управління та рівнем передачі даних;

– централізоване управління мережею, яке відбувається із використанням контролера;

– можливість програмування мережі. Метою SDN є можливість застосовувати додатки, що програмним чином будуть впливати на всю мережу. Вони нададуть змогу підвищити надійність мережі шляхом надання нових засобів безпеки, оптимізувати процес маршрутизації трафіку та процес визначення та надання пріоритетів, що забезпечить покращену якість обслуговування [24];

– віртуалізація всіх можливих фізичних ресурсів мережі.

Надалі наведемо порівняння мереж SDN із традиційними IP-мережами.

Традиційна мережа використовує пристрої із фіксованими функціями, такі як комутатор або маршрутизатор (рис. 1.8). Кожен із цих пристроїв має певні функції, які добре працюють разом і підтримують мережу. Якщо функції мережі реалізовані як апаратні конструкції, то, як правило, її швидкість підвищується [25].

Гнучкість є постійною перешкодою для традиційних мереж. Мало API-інтерфейсів доступні для забезпечення, і більшість комутаційних апаратних та програмних засобів є пропрієтарними. Традиційні мережі часто добре працюють із запатентованим програмним забезпеченням, але це програмне забезпечення не може бути швидко змінено за необхідності.

Традиційні мережі побудовані за наступним принципом [26]:

1. Функції традиційних мереж реалізуються насамперед на виділених пристроях, що використовують один або кілька комутаторів, а також маршрутизаторів та контролерів доставки додатків.

2. Функціональні можливості традиційних мереж значною мірою реалізовані у виділеному обладнанні, такому як специфічні інтегральні схеми (ASIC). Одним з негативних аспектів цієї традиційної апаратно-орієнтованої мережі є її обмеження.

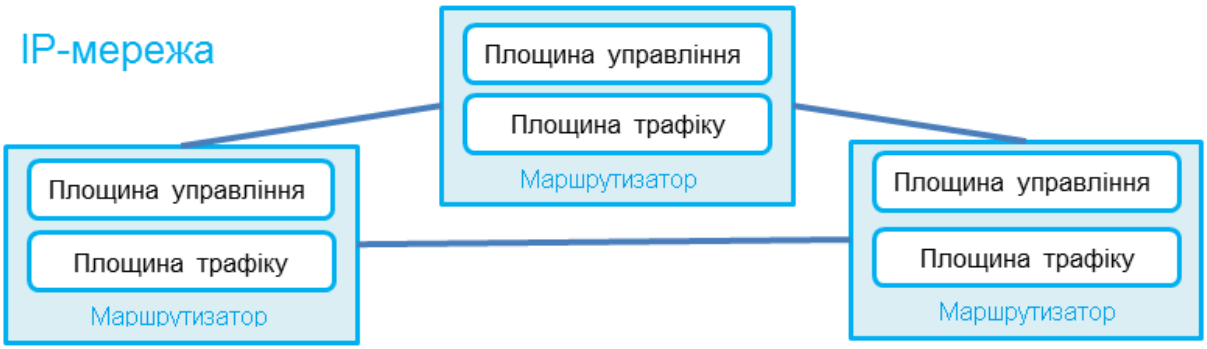


Рис.1.8. Спрощена архітектура IP-мережі

Найбільш помітна різниця між SDN та традиційними мережами полягає в тому, що SDN заснована на програмному забезпеченні, тоді як традиційна мережа зазвичай працює на апаратному забезпеченні [27]. Оскільки SDN заснована на програмному забезпеченні, вона є більш гнучкою, що дозволяє користувачам більше контролювати та полегшувати управління ресурсами практично по всій площині управління.

І навпаки, традиційні мережі використовують комутатори, маршрутизатори та іншу фізичну інфраструктуру для створення зв'язків та запуску мережі.

Контролери SDN мають інтерфейс, який взаємодіє з API. Завдяки такому спілкуванню розробники програм можуть безпосередньо програмувати мережу, на відміну від використання протоколів, необхідних для традиційних мереж (рис. 1.9).

SDN дозволяє користувачам використовувати програмне забезпечення для емулювання нових пристроїв замість використання фізичної інфраструктури, тому IT-адміністратори можуть направляти мережеві шляхи та попереджувати організацію мережевих послуг. На відміну від традиційних комутаторів, SDN також має можливість краще спілкуватися з пристроями, що використовують мережу.

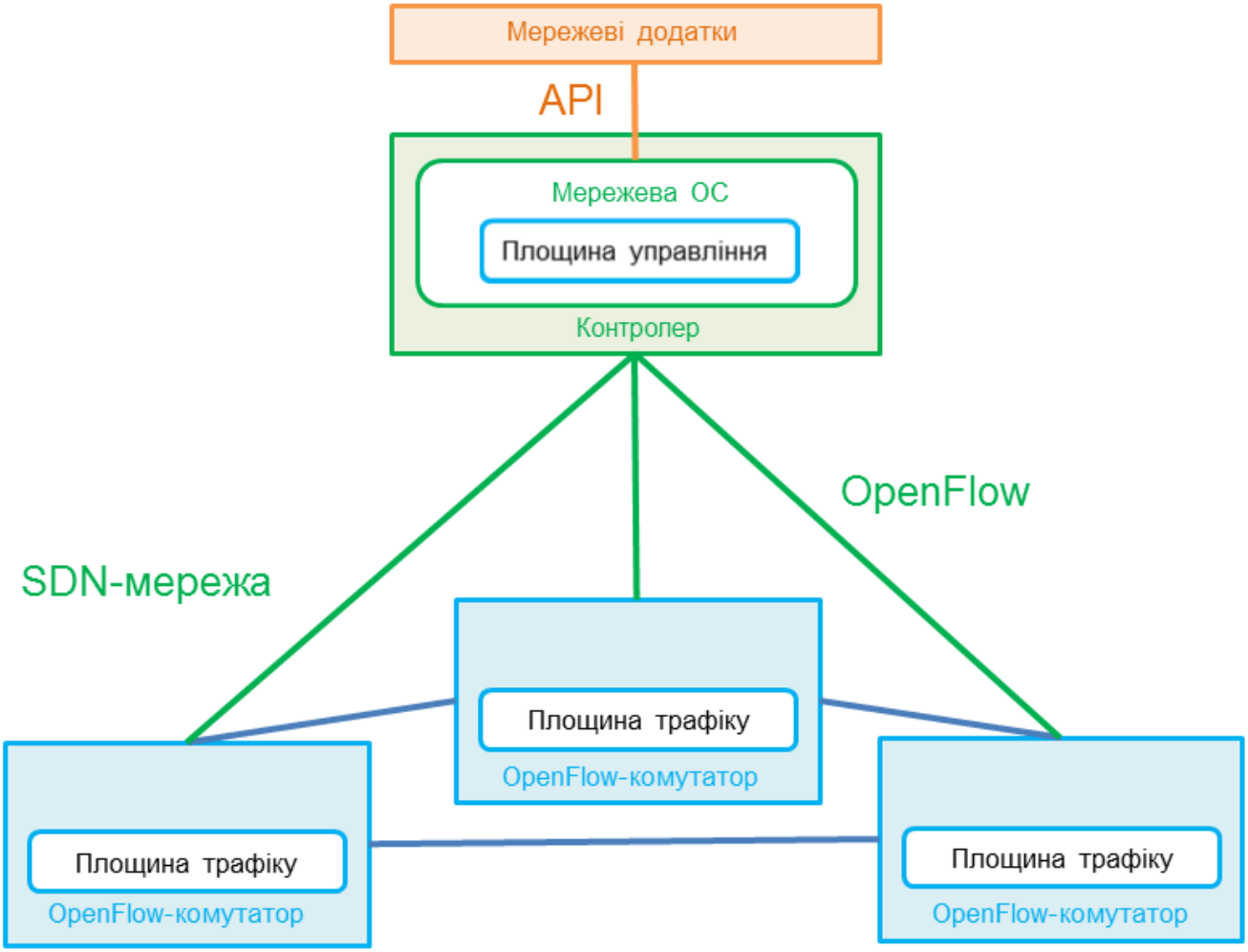


Рис.1.9. Спрощена архітектура SDN-мережі

Віртуалізація уособлює основну різницю між SDN та традиційними мережами. Коли SDN віртуалізує всю вашу мережу, вона генерує абстрактну копію вашої фізичної мережі та дозволяє вам надавати ресурси з централізованого місця.

На відміну від традиційної мережі, фізичне розташування площини управління заважає IT-адміністратору контролювати потік трафіку.

За допомогою SDN площина управління стає програмною, що дозволяє отримати доступ до неї через підключений пристрій. Цей доступ дозволяє IT-адміністраторам управляти потоком трафіку з більшою деталізацією через централізований користувальницький інтерфейс (UI). Це централізоване місцезнаходження надає користувачам більший контроль над тим, як працюють їхні

мережі та як вони налаштовані. Можливість швидко обробляти різні конфігурації мережі із централізованого інтерфейсу особливо корисна для сегментації мережі.

SDN став популярною альтернативою традиційній мережі, оскільки він дозволяє IT-адміністраторам надавати ресурси та пропускну здатність за потреби, не вимагаючи вкладення додаткової фізичної інфраструктури. Традиційні мережі потребують нового обладнання, щоб збільшити пропускну здатність мережі. Парадигма для SDN порівняно з традиційними мережами може бути розроблена узагальнено: одна вимагає більше обладнання для розширення, а інша – лише натискання клавіш.

В архітектурі програмно-конфігурованих мереж виділяють три основні рівні, що вказані на рис. 1.10 [28]:

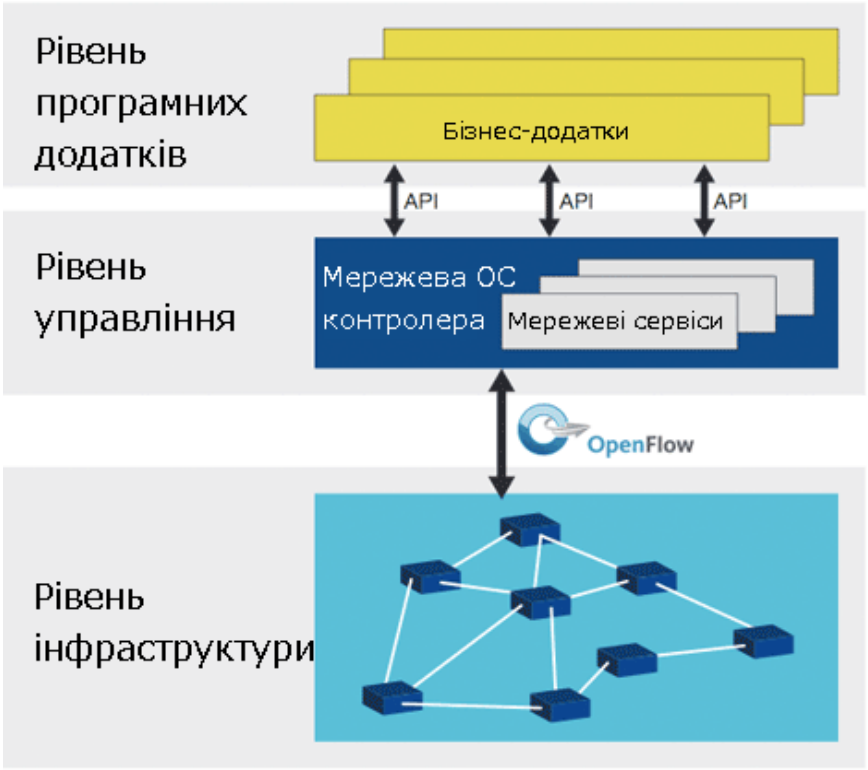


Рис.1.10. Архітектура SDN

– **інфраструктурний рівень**, який представляє набір мережевих пристроїв (комутаторів, маршрутизаторів і відповідно каналів передачі даних);

– **рівень управління,** який охоплює мережеву операційну систему, що здатна забезпечити додаткам надання мережевих сервісів, і відповідно програмний інтерфейс для управління мережевими пристроями і всією мережею;

– **рівень додатків**, що забезпечує гнучке та більш ефективне управління мережею.

Рівень додатків SDN дозволяє забезпечувати програмування мережі як єдиного цілого, а адміністраторам не доведиться при цьому займатися налаштуванням окремих пристроїв. Цей рівень пов'язаний з інтерфейсом контролера API (Application Programming Interface) – прикладним програмним інтерфейсом, що використовується сучасними програмістами, а відповідно й полегшує написання програмного забезпечення для контролерів.

Таким чином, на рівні управління працює контролер з встановленою на нього мережевою операційною системою. Контролер характеризується повним баченням мережі, а це дозволяє надавати пристроям мережі вказівки щодо обробки трафіку.

На рівні ж інфраструктури знаходяться відносно прості пристрої, яким вже не потрібно працювати з десятками протоколів. Достатньо просто слідувати інструкціям контролера, а тому, відповідно вони можуть бути простішими і дешевшими.

Для взаємодії між рівнем управління та рівнем інфраструктурних рішень використовується протокол OpenFlow. Він використовується для управління мережевими комутаторами і маршрутизаторами за допомогою контролера мережі. Це управління замінює або доповнює працюючу на комутаторі (маршрутизаторі) вбудовану програму, що здійснює побудову маршруту, створення карти комутації і т. д. Контролер може бути використаний для управління таблицями потоків комутаторів, на підставі яких приймається рішення про передачу прийнятого пакета на конкретний порт комутатора. Таким чином, в мережі формуються прямі мережеві з'єднання з мінімальними затримками передачі даних і відповідно необхідними параметрами якості обслуговування.

Пристрої, що підтримують роботу протоколу OpenFlow складаються з трьох компонент [28]:

– таблиця потоків (flow table);

– безпечний канал (secure channel);

– протокол OpenFlow.

В результаті, така архітектура (рис. 1.10) надає ряд переваг [29]:

– **Централізоване управління мережею.** SDN допомагає централізувати управління та забезпечення підприємствами, пропонуючи єдину точку зору на всю мережу. SDN також може прискорити надання послуг та підвищити оперативність надання доступу до віртуальних та фізичних мережевих пристроїв у центральному місці.

– **Цілісне управління підприємством.** Мережі повинні задовольняти зростаючий попит на обробку запитів. SDN допомагає IT-відділу налаштувати конфігурацію мережі, не впливаючи на неї. Крім того, на відміну від простого протоколу керування мережею (SNMP), SDN підсилює управління фізичними та віртуальними комутаторами та мережевими пристроями, що надходять від центрального контролера.

– **Більший рівень безпеки.** Віртуальні машини створюють складні завдання для брандмауерів та фільтрації вмісту, що також ускладнюється персональними пристроями. Створивши центральний пункт управління для регулювання інформації про безпеку та політику для вашого підприємства, контролер SDN швидко стає позитивним фактором для IT-відділу будь-якого підприємства.

– **Зниження експлуатаційних витрат.** Деякі переваги SDN, такі як ефективне адміністрування, покращення використання сервера та вдосконалений контроль віртуалізації, можуть подвійно допомогти зменшити операційні витрати. Оскільки багато питань регулярного адміністрування мережі можна автоматизувати та централізувати, SDN також може допомогти зменшити експлуатаційні витрати та збільшити адміністративні заощадження.

– **Економія обладнання та зменшення капітальних витрат.** Прийняття SDN допомагає оживити старі мережеві пристрої та спрощує процес оптимізації

обладнання. Дотримуючись вказівок контролера SDN, старе обладнання можна перепрофілювати, в той час як дешеве обладнання можна розгорнути для оптимального ефекту.

– **Хмарні ресурси.** Використання SDN для абстрактних хмарних ресурсів допомагає спростити процес їх об'єднання. Контролери SDN можуть керувати всіма мережевими компонентами, що складають масивні платформи ЦОД.

– **Послідовна та своєчасна доставка контенту.** Однією з великих переваг SDN є можливість маніпулювати трафіком даних. Простіше мати якісний сервіс для передачі голосу через Інтернет (VoIP) та передачі мультимедіа, якщо ви можете керувати та автоматизувати трафік даних. SDN також допомагає у програванні високоякісних відео, оскільки SDN покращує швидкість реагування мережі і, отже, створює поліпшену взаємодію з користувачами (UX).

1.2.2. Концепція NFV

Віртуалізація мережевих функцій – Network Functions Virtualization (NFV) – це принципово нова концепція, яка використовує технології віртуалізації на рівні мережі і відповідно мережевих функцій [30]. NFV вважають також еволюцією мереж SDN (Software-Defined Networking). SDN разом із NFV можуть бути застосованими на мережевому рівні, рівні інфраструктури і рівні сервісів користувача. При цьому, контролер SDN може бути вбудований в систему управління NFV, також може бути реалізована як віртуальна або навіть апаратна платформа. Концептуальне поєднання архітектур NFV і SDN на якісному рівні описано в документі Європейського інституту телекомунікаційних стандартів (ETSI): Network Functions Virtualisation (NFV); Ecosystem; Report on SDN Usage in NFV Architectural Framework [31].

В першу чергу, NFVi (NFV Infrastructure) – інфраструктура, яка реалізує концепцію віртуалізації втратити зв'язок із мережею. Це мережева і обчислювальна платформа, на якій базується віртуалізація, рівень самої віртуалізації. VNFs (Virtualised Network Functions) – віртуалізовані функції, наприклад: Firewall (FW),

DHCP, Mobility Management Entity (MME), Evolved Packet Core (EPC), Serving Gateway (SGW) і ін. Система MANO (NFV Management and Orchestration) – система комплексного управління та моніторингу. vCPE (virtual Customer Premises Equipment) – віртуалізація кінцевих пристроїв замовника в операторському хмарі (контейнер VNF). Гнучка схема надання даних сервісів бізнесу – це основна рушійна сила для переходу провайдера на концепцію NFV (на думку Infonetics Research Survey, 2014 року) [32].

Розподілена схема NFV включає наступні варіанти базування компонент [33]:

– в ЦОД провайдера послуги – вся обробка відбувається в ЦОД провайдера;

– в замовника сервісу – частина обробки відбувається на майданчику замовника сервісу;

– змішаний режим – поєднання обох варіантів локалізації.

Перелічимо основні цілі, які можуть бути досягнуті в результаті переходу на концепцію NFV [34]:

– прискорення інноваційних процесів в наданих сервісах за допомогою програмного розгортання і впровадження мережевих функцій та наскрізних послуг;

– поліпшення експлуатаційної ефективності в результаті спільної автоматизації і скорочення операційних процедур, а також зниження енергоспоживання за рахунок міграції робочих навантажень і відключення невикористаного обладнання;

– стандартизація інтерфейсів між мережевими функціями і їх керуючими об'єктами і можливість надання мережевих елементів різними гравцями (наприклад, VNFaaS-провайдер 1 на базі IaaS-провайдера 2);

– підвищення ефективності капіталів і загальної гнучкості мережевий архітектури завдяки відходу від апаратних реалізацій.

Таким чином, наприклад, стосовно до мережевих операторів послуг і їх клієнтам забезпечуються наступні переваги [35]:

– скорочення операційних капітальних витрат і операційних витрат (CAPEX / OPEX) за рахунок зниження вартості обладнання і зниження споживання енергії;

– скорочення «виходу на ринок» для розгортання нових мережевих сервісів, поліпшення віддачі від інвестицій в нові послуги;

– велика гнучкість для збільшення, зменшення або розвитку послуг; відкритість для ринку віртуальних пристроїв і «чистих» учасників;

– можливість апробування та впровадження нових інноваційних послуг з меншим ризиком; відхід від частих EoL / EoS, спрощення RMA та інших процедур за рахунок скорочення різноманітності парку апаратних пристроїв.

1.3. Аналіз можливостей використання MEC для мереж 5G

Розроблені різні сценарії, а ефективність ЕС порівнюється та аналізується із системами хмарних обчислень. Виходячи з результатів, підкреслюється ефективність роботи ЕСС та вирішуються проблеми, що стосуються різних мережевих систем.

Multi-Access Edge Computing (MEC) [36] пропонує розробникам додатків та постачальникам контентних можливостей хмарних обчислень та середовищу IT-сервісу можливість проведення високопродуктивних обчислень на межі мережі. Це середовище характеризується наднизькою затримкою та високою пропускною здатністю, а також доступом до інформації в радіомережі в реальному часі, яку можна використовувати додатками.

MEC забезпечує нову екосистему та ланцюжок вартості. Оператори можуть відкрити свою мережу радіодоступу (RAN) для уповноважених третіх сторін, що дозволяє їм гнучко та швидко розгортати інноваційні програми та послуги до мобільних абонентів, підприємств та вертикальних сегментів [37].

На рис. 1.11 приведено архітектуру «Все в одному».

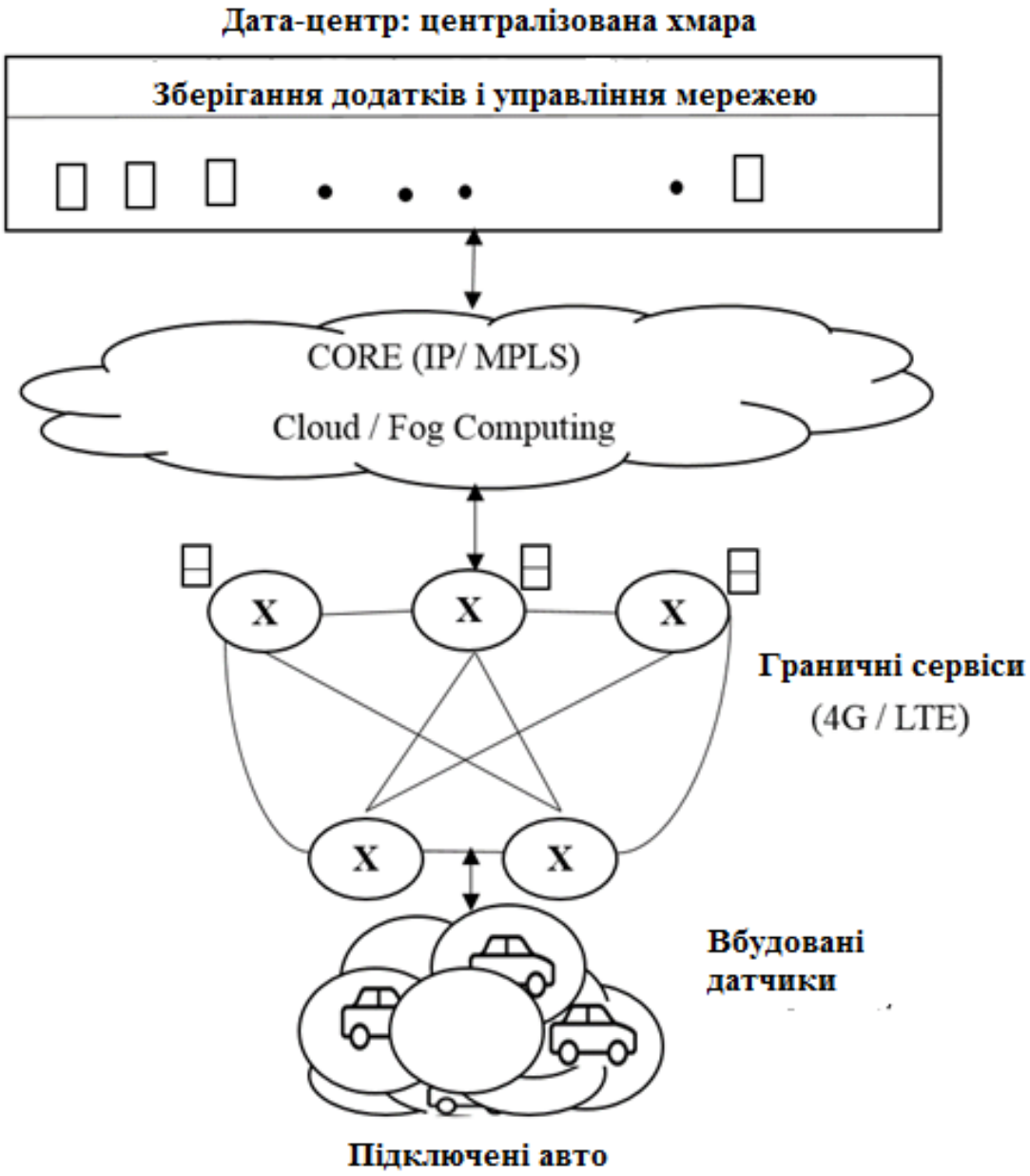


Рис. 1.11. Обчислювальна архітектура «Все в одному»

MEC – це природний розвиток в еволюції мобільних базових станцій та конвергенції IT та телекомунікаційних мереж. Багаторазовий обчислювач Edge дозволить отримати нові вертикальні бізнес-сегменти та послуги для споживачів та клієнтів підприємств. Сюди включають [38]:

– відеоаналітика;
– послуги локації;
– Інтернет-речі (IoT);

– доповнена реальність;
– оптимізований локальний розподіл вмісту;
– кешування даних.

Це однозначно дозволяє програмам використовувати локальний вміст та інформацію в режимі реального часу за умови існування локальної мережі доступу. За допомогою розгортання різних служб та кешування вмісту на межі мережі, основні мобільні мережі позбавляються від подальших перевантажень і можуть ефективно обслуговувати інші більш глобальні цілі.

MEC дозволяє операторам стільникового зв'язку відкривати свою радіодію, що реагує на легітимні додатки третьої сторони, таким чином, як розробники пропонують і встановлюють контент. Для них MEC надає стандартизовану відкриту середу з максимальною можливістю виходу, мінімальними затримками та збереженням у реальному часі оперативної інформації – актуальні робочі мережі, що знаходяться в роботі (рис. 1.12) [39].

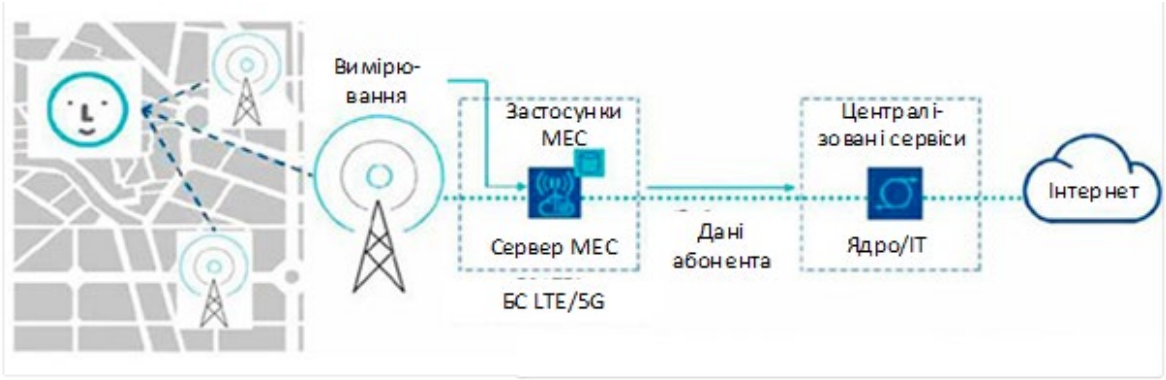


Рис. 1.12. Архітектура мережі із підтримкою MEC

Використовуючи архітектуру Multi-Access Edge Computing, оператори передають можливість розширення на кордонах, використовуючи нове покоління, в CDN, DNS, 5G та телеметричних сервісах IoT. Вони надають змогу запропонувати геолокаційні служби, відеоаналітику, послуги доповненої реальності та інформацію для підключених автомобілів. Більше того, у оператора буде можливість

представити інфраструктуру за сервісною моделлю сторонніх організацій. Це важливо для середнього та малого бізнесу, яка відкрита для сучасних технологій.

Multi-Access Edge Computing ідеально підходить для того, щоб забезпечити готовність наявної інфраструктури для розширення послуги високої пропускної здатності з ультрамалими затримками і в майбутньому плавно перейти до мереж 5G.

Переваги використання MEC. Оскільки численні бізнес-сервіси були перенесені з хмари на границю мережі, завдяки своїм найсучаснішим обчислювальним послугам, економія масштабу значно покращилася, надаючи величезні переваги бізнес-сервісам, які надають інфраструктуру, а також підприємствам, які її використовують. Нижче наведено деякі суттєві переваги підходу MEC [39, 40]:

– Використовуючи MEC, швидше проводити аналітику даних, що покращує загальну ефективність додатків у режимі реального часу.

– Вартість впровадження центру обробки даних значно зменшується шляхом вибору вузлів граничних серверів.

– Зменшується мережевий трафік.

– Наявність однієї точки відмови та адаптація розподілених обчислень.

– Можливість віртуалізації мережевих ресурсів.

– Покращена масштабованість мережі.

– Покращення параметрів QoS за рахунок мінімізації відстані, на яку відбувається передача даних.

– Підвищений рівень надійності, за рахунок встановлення додатків в безпосередній близькості від кінцевого пристрою.

– Менш складне управлінні обладнанням MEC.

Таким чином, підсумовуючи, можна стверджувати, що застосування MEC перетворюється на основну технологію, яка підвищує ефективність та продуктивність бізнесу та промислового сектору.

1.4 Дослідження недоліків MEC

Граничні обчислення з множинним доступом (MEC) – це мережева архітектура, яка забезпечує доступ у режимі реального часу, пропускну здатність, низьку затримку доступу до інформації радіомережі, що дозволяє операторам відкривати свої мережі в принципово новій екосистемі [41].

MEC дозволяє кілька типів (режимів) доступу на краю. Переваги цієї технології виходять за рамки мобільних, перетворюються на технології Wi-Fi та фіксованого доступу. Він передбачає розміщення обчислювальних ресурсів та ресурсів зберігання ближче до кінцевого споживача або підприємства, і, як очікується, він стане головним фактором для можливостей 5G, покращуючи доставку даних та відповідно досвід роботи з додатками.

Шлях між пристроєм кінцевого користувача та місцем розміщення даних часто довгий і недосконалий. MEC скорочує мережевий шлях, знижуючи затримку, збільшуючи надійність та покращуючи загальну ефективність мережі.

Хоча MEC працює протягом останніх кількох років і надалі буде частиною підготовки та розгортання 5G, спостерігачі галузі вже визначили п'ять основних проблем [37, 42], які необхідно вирішувати найближчим часом:

1. *Місце розташування*. Чим ближче обчислювальні ресурси знаходяться до користувача, тим складніше стає питання більш зручного місця розташування цих ресурсів. Мобільні програми користувачів, розумні міста та автономні транспортні засоби потребуватимуть більшої географічної доступності обчислювальних ресурсів для покращення процесу доставки даних. Облаштування декількох десятків регіональних центрів обробки даних або кілька сотень центральних офісів – це одна, більш проста задача, а от розміщення обчислювальної техніки на тисячах окремих баштових майданчиків (базових станціях), це інша більш складна задача, що суттєво підвищує вартість та складність розгортання. Тому дуже важливою є розробка нових методів, що дозволять більш ефективно планувати місце розташування обчислювальних ресурсів MEC ближче до користувача.

2. *Енергоспоживання*. Створення центрів обробки даних з каналами подвійного енергоживлення дозволяє забезпечити стійкість у випадку, якщо по одному із каналів зменшується потужність живлення, або навіть зникає. Однак, залежно від розміру крайової обчислювальної установки та місця її розташування, не завжди можливо отримати достатню кількість каналів подвійної потужності, щоб буде відповідати існуючим стандартам та вимогам для установок МЕС.

3. *Фізичне місце розташування*. Переміщення обчислювальних ресурсів до центральних офісів чи приміщень клієнта може означати або необхідність в дооснащенні цих середовищ спеціалізованими пристроями (живлення, сервери тощо), або роботу в місцях, дуже далеких від ідеального. Невеликим центрам обробки даних з незначними обчислювальними потужностями, особливо тим, які можуть бути розміщені на самих вежах стільникового зв'язку, доведеться працювати в набагато складніших фізичних умовах.

4. *Розташування оптичної інфраструктури та інших мереж*. Крайні обчислювальні засоби потребуватимуть доступу до волоконних мереж і, швидше за все, до точки догляду за іншими мережами. Отже, капітальний ремонт (надсилання мережевих даних по виділеному маршруту з метою отримання даних там раніше або тому, що це коштує менше) є також важливою науково-технічною задачею. При цьому слід звертати увагу на те, що МЕС може різко зменшити кількість віддалених робіт, які необхідно проводити на сайтах. Справа, зрештою, полягає у зменшені затримки в мережах 5G, роблячи необхідною локальну обробку даних. Отримати потрібну ємність та необхідні зв'язки між собою в заданому місці обчислень може бути набагато складнішим завданням, ніж у відносно невеликій кількості великих центрів обробки даних.

5. *Операційна діяльність*. Чим менше осередків обробки крайових даних, тим більше їм потрібно буде працювати. Віддалене управління, моніторинг та безпека будуть вирішальними для зменшення витрат на розгортання. Оскільки МЕС – це відносно нова технологія, яка не перезріла з точки зору найкращих практик та знань, то можливість працювати з центрами обробки даних все ще потребує певного дотику з боку доволі досвідчених людей. Це робить ці крайові середовища – які

потенційно є досить маленькими, але численними за кількістю – потенційно дуже дорогими для роботи.

6. *Конфіденційність та кібербезпека*. МЕС відіграє ключову роль у наданні послуги, чутливої до затримки, для різних різнорідних мережевих інтелектуальних програм, однак, це може вирішити проблеми, пов'язані із безпекою та конфіденційністю системи. Деякі з основних проблем коротко узагальнені як наступні:

Сервери МЕС поширюються на межі мережі, що робить систему більш вразливою до різних загроз безпеці. Існуючі стандарти шифрування не застосовуються в ЕС через обмеження ресурсів, що існують на серверах. Щоб забезпечити надійний захист від загроз та атак безпеці, необхідно спроектувати схему автентифікації в невеликій вазі, де сервери ЕС автентифікують кінцеві пристрої без будь-якої затримки. Ще одне питання для ЕС – це проблема управління довірою між граничним сервером та кінцевими вузлами. Оскільки граничні сервери розподіляються по всій мережі, обчислення довіри з одного сервера МЕС не може передавати довіру іншим серверам МЕС. Оскільки мобільність вузлів висока в розподілених мережах, таких як VANET та MANET, вузли будуть стикатися з різними граничними серверами, і тому їх потрібно періодично перевіряти. Для цього потрібно надійну систему управління довірою інтегрувати в середовище МЕС, яка здатна обробляти довіру як із серверів, так і з кінцевих вузлів.

На додаток до цього, підтримка конфіденційності даних є настільки ж складною для ЕС, оскільки обробка інформації висувається на границя мережі. Отже, розумні програми створюватимуть більший обсяг персоналізованої інформації та відомостей про місцезнаходження, які легко можуть бути порушені через відкритість у навколишньому середовищі. Таким чином, в систему повинна бути включена надійна схема захисту даних та перевірки довіри, яка може значно захистити географічну точність розташування та особисті дані користувачів.

1.5 Формування напрямків наукових досліджень

Обчислювальна технологія з граничним доступом (MEC) нещодавно стала новою парадигмою для полегшення доступу до розширених обчислювальних можливостей на межі мережі, в безпосередній близькості до кінцевих пристроїв, тим самим забезпечуючи широкий спектр послуг, залежних від затримок, які вимагають різні нові вертикальні галузі.

Вибір пристроїв для MEC є критичним для різних мережевих сценаріїв. Наприклад, у VANET пристрої MEC можуть бути транспортним засобом або спеціалізованим граничним сервером [43, 44]. Якщо транспортні засоби будуть обрані як граничні пристрої, обчислення розподіляються, але вартість впровадження буде високою. З іншого боку, якщо в мережі є виділений граничний сервер, вона може зіткнутися з проблемами в обробці зростаючих потреб кінцевих пристроїв. Таким чином, щоб мати ефективну систему MEC, додаток має включати ефективну схему управління ресурсами, яка повинна бути достатньо складною для управління як граничними серверами, так і підключеними пристроями.

Вибір обчислень серед граничних пристроїв – ще один складний параметр. У динамічній мережі обчислення через кілька граничних вузлів потрібно вивантажувати розподіленим чином. Без розподіленої схеми робоче навантаження стає упередженою, що з часом збільшує навантаження в деяких системах і виснажує їх системи живлення. Для забезпечення енергоефективної системи розподілу робочого навантаження необхідна ретельна розробка політики у поєднанні з ефективною організацією обчислень та управління.

Задача оптимізації розміщення масштабованих послуг на розподілених обчислювальних ресурсах мережі стільникового оператора. Автоматизований розподіл завдань між хмарою та краєм є складною задачею [45]. Через певні технологічні обмеження в аспекті обчислення та зберігання MEC не виключає повністю послуг хмарних обчислень, оскільки деякі обчислення все ще проводяться на хмарних серверах для підвищення надійності системи. У MEC потрібно включити надійну схему планування завдань, яка повинна належним чином

розподіляти завдання на граничні та хмарні сервери, не впливаючи на продуктивність системи [46].

Зниження накладних комунікацій для досягнення QoS в MEC також є складним завданням. Без будь-якої стандартизації мережі та протоколів системи MEC можуть страждати від проблем, пов'язаних із мережею, наприклад, перевантаженості мережевого трафіку та відмови в обслуговуванні [47]. Для систем MEC повинні бути розроблені ефективні мережеві протокол та стандарти, щоб забезпечити безперебійну роботу без будь-якого відставання в мережі.

Управління мобільністю в MEC також є складним завданням. Пристрої, що використовуються в мережах високої мобільності, наприклад, MANET та VANET, будуть стикатися з частими хендоверами. Як результат, на обробку даних та прийняття рішень можуть значно вплинути затримки, викликані цим. У пристроях MEC повинні бути включені надійні схеми співпраці для ефективного вирішення таких питань мобільності.

Забезпечення безпеки та конфіденційності в системі EC також є досить складним завданням. Із обчисленнями, висунутими на межі мережі, інформація стає вразливою до різних загроз та атак. У систему повинні бути включені ефективні системи управління довірою для вирішення питань безпеки та запобігання можливим шкідливим вторгненням / атакам.

Розміщення віртуальних машин (VM) – добре вивчена тема в хмарних обчисленнях. В [48] автори розглядують методи розміщення та міграції віртуальних машин в хмарному середовищі. В [49] пропонують використовувати таксономію для класифікації цих рішень. Але ці підходи до традиційних централізованих хмарним обчисленням не враховують, що обчислювальне edge середовище є більш розподіленою, різнорідною, чутливою до затримок і мають обмежені ресурси.

Розвантаження обчислень значить перенесення задач з одного приладу на зовнішню платформу, наприклад, edge і хмарні обчислення. В результаті це дозволяє виконувати інтенсивні обчислення додатків в приладах з обмеженими ресурсами, що одночасно зменшує енергоспоживання. Окрім того, важливою частиною розвантаження є рішення розвантажувати чи ні. В [50] дослідження

авторів, що стосується розвантаження обчислювання в контексті сценаріїв мобільних edge обчислень. В цій роботі ми не зацікавлені в цьому процесі прийняття рішення щодо розвантаження, але розміщення служб і розвантаження обчислень можуть розглядатись як додаткові проблеми.

Деякі роботи вирішують проблему розміщення сервісів в контексті egde обчислювань. В [51] автори мають намір оптимізувати проблему розміщення і переміщення додатків в архітектурі MEC з декількома ієрархічними рівнями для мінімізації загальних експлуатаційних витрат. Обмеження цієї роботи заключається в припущенні наявності достатніх ресурсів для всіх додатків. В [52] автори досліджують проблеми розподілення базової станції, розміщення VM і розподілення задач для медицинських додатків в MEC, щоб мінімізувати загальну вартість, яка б задовольняла максимальну допустиму затримку додатків. Тим не менш, автори вивчають тільки розгортання додатків на базових станціях. В [53, 54] також вирішують проблеми розміщення віртуальних машин і балансування навантаження в MEC. Хоч ціллю роботою є мінімізація середнього часу відповіді на запит, в ньому не враховується крайні терміни часу на відповідь, особливо для додатків, які чутливі до затримок.

Що стосується порушення QoS, то в [55] автори пропонують стратегію розміщення і міграції VM, щоб мінімізувати енергозберігаюче споживання енергії вузлами MEC але з порушенням вимог скрізної затримки додатків. Передбачається також, що вузли MEC мають однакові можливості використовування ресурсів, а віртуальні машини мають однакові потреби в ресурсах. [56] вивчає можливість прийняття рішення про планування і розміщення віртуальних машин в MEC з ціллю максимізації доходу постачальника інфраструктури, мінімізування коливань погодження на рівні обслуговування (SLA) і забезпечення справедливого розподілення ресурсів між постачальниками послуг. Не дивлячись на те, що робота розслідує порушення SLA з точки зору часу відповіді, вважається що час обробки відповідальна тільки за затримку відповіді, яка ігнорує мережеву затримку.

Небагато робіт пропонують рішень, які засновані на генетичному алгоритмі. [57-59] вивчає розміщення додатків в ієрархічній розподіленій архітектурі fog-a, а

не в хмарі, які задовольняють при цьому крайній термін виконання додатку. Але пропоноване кодування хромосоми може генерувати необхідні рішення, тоді при розрахунку придатності додається штраф за порушення цих випадків. Для рішення проблеми генерації недопустимих рішень [60] пропонує кодування зі зміщеним рандомним ключем для забезпечення стійкого розміщення критично важливих додатків в георозподілених хмарах. Недолік цієї роботи полягає в тому, що в ній не враховується необхідність затримки додатків.

Щоб подолати деякі обмеження, які пов'язані з проблемами в раніше описаних роботах, в цій роботі необхідно розглядати розміщення послуг в автономному режимі в розподіленій гетерогенній і обмеженими ресурсами edge обчислювальними, які за допомогою масштабованих додатків, які направлені на мінімізацію порушення QoS. В автономному випадку міграція додатків і мобільність користувачів не розглядатимуться. Окрім того, набори додатків і обчислювальних вузлів, що відомі завчасно та які не змінюються протягом розміщення.

Як результат, розробляючи систему MEC для мережі, дослідники та практикуючі виробники повинні враховувати ці вищезгадані проблеми для досягнення стійкої масштабованості та надійності системи. Оскільки вимоги та виклики різні для кожної мережі, такі як MANET, VANET та IoT, дизайн MEC повинен бути адаптований до індивідуальних вимог.

Задача завантаження задач та планування їх виконання. Пристрої Інтернету речей (IoT), будучи дуже поширеними та підключеними, можуть вивантажувати свої обчислювальні завдання, які обробляються програмами, розміщеними на серверах MEC через обмежену кількість акумуляторних, обчислювальних і пам’ятних можливостей. Такі програми IoT, що надають послуги для завантажених завдань пристроїв IoT, розміщуються на крайових серверах з обмеженими обчислювальними можливостями [61].

Враховуючи неоднорідність вимог до завантажених завдань (різні обчислювальні вимоги, затримка тощо) та обмежені можливості MEC, повинно прийматися рішення про завантаження задач (завдання на призначення додатків) та планування (порядок їх виконання), що стало складною проблемою комбінаторного

характеру. Тому в даній дисертаційній роботі необхідно вирішувати задачу динамічного розвантаження та планування задач. Необхідно математично сформулювати цю задачу і в силу її складності розробити нове рішення для розкладання на основі техніки Бендерса.

Задача підвищення енергоефективності під час розвантаження завдань. В останні роки, як вже було показано, з розвитком smart мобільних пристроїв виникає та привертає багато уваги до себе все більше і більше додатків, що мають високу обчислювальну потужність, вимагають великої кількості ресурсів та споживають дуже багато енергії [62 – 64], наприклад: інтерактивні онлайн ігри, розпізнавання жестів та облич, доповнена реальність та інші. Проте, завдяки таким можливостям обчислення, об'єм пам'яті та енергія мобільних пристроїв завжди обмежена, тому якість досвіду від обчислення мобільними девайсами є незадовільною.

В цьому випадку, головне призначення граничних обчислень – це зменшення енергетичного споживання та затримки, ретельно спроектувавши схему розвантаження завдання та спосіб управління ресурсами [62], що досліджувалося як у однокористувацькій, так і у багатокористувацькій системі. [65] досліджує рішення задачі щодо розвантаження у багатокористувацькій системі на основі теорії ігор; у даному алгоритмі, користувачі мобільних пристроїв вирішують чи завантажувати задачу на сервери чи ні, та які канали зв'язку використовуються розподіленим способом. В [66] пропонується оптимальна політика розвантаження для додатків, які включають графіки залежності послідовних компонентів та мобільні пристрої з підтримкою мульті-радіо (*multi-radio*). Дана політика може мінімізувати енергетичне споживання мобільних пристроїв починаючи з часу виконання, який не перевищує заданий поріг та відсоток переданих даних. В [67] запропоновано ефективну обчислювальну модель для системи з граничними обчисленнями, яка поєднує обчислення та загальну комунікацію для покращення продуктивності системи. У цій системі лише один користувацький вузол, один допоміжний вузол та одна точка доступу, що додаються до одного граничного серверу. Більше алгоритмів, спрямованих на підвищення ефективності роботи даного типу систем, можна знайти в таких дослідженнях, як [63 – 66]. У цих дослідженнях автори

класифікують алгоритми у різні категорії; більше того, пропонуються подальші роботи та виклики в предметній галузі.

З точок зору, що запропоновані у [63, 64], не тільки рішення з розвантаження, але також управління ресурсами є важливим для підвищення продуктивності вищерозглянутих систем. Управління ресурсами включає розподіл каналів зв'язку, контроль можливостей процесора, контроль потужності передачі та інше [68]. У попередніх роботах, розподілення каналу зв'язку та контроль можливостей процесора у системі з граничними обчисленнями були широко досліджені; проте, дослідження проблеми контролю потужності (тобто управління потужністю передачі) в системі з граничними обчисленнями тільки починається. Вже було запропоновано декілька алгоритмів контролю потужності для таких систем. Для прикладу, у [68], розглянуто дослідження контролю потужності у системі для однокористувацького сценарію; Автори [69] намагалися зменшити затримку обслуговування за допомогою контролю потужності передачі у *cloudlets*; у цьому алгоритмі, користувачі передають дані у *cloudlet* у циклічній формі, таким чином втручання між різними користувачами ігнорується, що дорівнює системі одного користувача; у [66] та [69] спільно вивчалося контроль радіо та обчислювальних ресурсів (що включають потужність передачі) у багатокористувацьких системах. Проте у [64] та [70], втручання, яке викликане передачею інших користувачів, що є вирішальним та має великий вплив на продуктивність багатокористувацьких систем [65, 71], не враховується; більше того методологія цих двох алгоритмів є централізованою, що не завжди є ефективно у розповсюджених системах, таких як бездротова сенсорна мережа та IoT додатки. [65] та [71] вивчають продуктивність системи у сценарії втручання (*interference-aware*); проте, потужність передачі користувачів мобільних пристроїв не може бути встановлена у цих двох алгоритмах.

Тому дуже важливою задачею, яка вирішувалась в даній дисертаційній роботі є управління випромінюваною потужністю користувальницьких пристроїв під час розвантаження завдань в MEC.

Висновки до розділу 1

Дослідження, проведені в першому розділі, надали змогу отримати наступні результати.

1. Було проаналізовано ключові напрямки та рушійні сили розвитку стільникових мереж зв'язку п'ятого покоління. Відзначимо значне збільшення швидкостей передавання даних, зменшення затримки, підвищення економічної ефективності, енергозбереження тощо. Разом з тим, слід констатувати неможливість забезпечити необхідний рівень якості надання послуг абонентам без розгортання сучасних технологічних рішень, таких наприклад, як MEC.

2. Було проаналізовано ключові технологічні рішення, які використовуються у стільникових мережах 5G. Серед них було виділено концепцію MEC, як основу для зменшення наскрізної затримки в мережі 5G.

3. Було визначені основні переваги та недоліки концепції MEC, а також було продемонстровано неабияку перспективність досліджуваної технології.

4. Було проведено ретельний аналіз напрацювань за напрямом досліджень MEC. Було проаналізовані основні проблеми, які виникають, або будуть виникати під час розгортання даного типу систем. На основі цього аналізу було сформульовано наукові завдання, які вирішувались в даній дисертаційній роботі (розділи 2-4).

Список використаних джерел у першому розділі

1. Маковеева М. М., Шинаков Ю. С. Системы связи с подвижными объектами: учеб. пособие для вузов. Москва: Радио и связь, 2002. 440 с.

2. Гельгор А. Л., Попов Е. А. Сотовые сети мобильной связи стандарта UMTS: учеб. пособие. Санкт-Петербург: Изд-во Политехн. ун-та, 2011. 227 с.

3. Широкополосные беспроводные сети передачи информации / Вишневский В. М., Ляхов А. И., Портной С. Л., Шахнович И. В. Москва: Техносфера, 2005. 592 с.

4. Бабков В. Ю., Цикин И. А. Сотовые системы мобильной радиосвязи: учеб. пособие. 2-е изд., перераб. и доп. Санкт-Петербург: БХВ-Петербург, 2013. 432 с.: ил.

5. Тихвинский В. О., Бочечка Г. С. Перспективы сетей 5G и требования к качеству их обслуживания. *Электросвязь.* 2014. № 11. С. 40–43.

6. Сети 5G/IMT-2020 & IoT-основа цифровой трансформации / Бутенко В., Веерпалу В., Девяткин Е., Федоров Д. *Электросвязь.* 2018. № 12. С. 4–9.

7. Gupta A., Jha R. K. A survey of 5G network: Architecture and emerging technologies. *IEEE access.* 2015. Vol. 3. P. 1206–1232.

8. Design considerations for a 5G network architecture / Agyapong P. K., Iwamura M., Staehle D. et al. *IEEE Communications Magazine.* 2014. Vol. 52, Iss. 11. P. 65–75.

9. Network function virtualization in 5G / Abdelwahab S., Hamdaoui B., Guizani M., Znati T. *IEEE Communications Magazine.* 2016. Vol. 54, Iss. 4. P. 84–91.

10. What is 5G? The business guide to next-generation wireless technology [Електронний ресурс] – Режим доступу: https://www.zdnet.com/article/what-is-5g-the-business-guide-to-next-generation-wireless-technology/

11. 5G Connectivity will Enable New Use Cases, 31 травня 2020 року [Електронний ресурс] – Блог. Режим доступу: https://www.connectivity.technology/2020/05/5g-connectivity-will-enable-new-use.html

12. Mobile network architecture evolution toward 5G / Rost P., Banchs A., Berberana I. et al. *IEEE Communications Magazine.* 2016. Vol. 54, Iss. 5. P. 84–91.

13. Network slicing in 5G: Survey and challenges / Foukas X., Patounas G., Elmokashfi A., Marina M. K. *IEEE Communications Magazine.* 2017. Vol. 55, Iss. 5. P. 94–100.

14. Samdanis K., Costa-Perez X., Sciancalepore V. From network sharing to multi-tenancy: The 5G network slice broker. *IEEE Communications Magazine.* 2016. Vol. 54, Iss. 7. P. 32–39.

15. Володина Е. Е., Девяткин Е. Е., Суходольская Т. А. Перспективные радиотехнологии (сети 5g/imt-2020, интернет вещей) в социально-экономическом

развитии страны. *Мобильный бизнес: перспективы развития и реализации систем радиосвязи в России и за рубежом.* 2018. С. 135–138.

16. Iwamura M. NGMN view on 5G architecture. *Proceedings of the IEEE 81st Vehicular Technology Conference* (*VTC-spring'15)*, (Glasgow, May 11–14, 2015). P. 1–5.

17. Morgado, A., Huq, K. M. S., Mumtaz, S., & Rodriguez, J. (2018). A survey of 5G technologies: regulatory, standardization and industrial perspectives. *Digital Communications and Networks*, *4*(2), 87-97.

18. A Look At The 5G Opportunity [Електронний ресурс] / Matt Bohles // Режим доступу: https://seekingalpha.com/article/4204568-look-5g-opportunity

19. Nokia Networks FutureWorks Network architecture for the 5G era Nokia Networks white paper Network architecture for the 5G era [Електронний ресурс] – Режим доступу: https://docplayer.net/9409582-Nokia-networks-futureworks-network-architecture-for-the-5g-era-nokia-networks-white-paper-network-architecture-for-the-5g-era.html

20. New paradigm of 5G wireless internet / Chih-Lin I., Han S., Xu Z. et al. *IEEE Journal on Selected Areas in Communications.* 2016. Vol. 34, No. 3. P. 474–482.

21. Optimising 5G infrastructure markets: The business of network slicing / Bega D., Gramaglia M., Banchs A. et al. *The 36th IEEE International Conference on Computer Communications (INFOCOM 2017)*, (Atlanta, GA, USA, May 1–4, 2017). P. 1–9.

22. A multi-access core: Ericsson's dual-mode 5G Core [Електронний ресурс] – Режим доступу: https://www.ericsson.com/en/core-network/5g-core

23. Are we ready for SDN? Implementation challenges for software-defined networks / Sezer S., Scott-Hayward S., Chouhan P. K. et al. *IEEE Communications Magazine.* 2013. Vol. 51, Iss. 7. P. 36–43.

24. Towards an elastic distributed SDN controller / Dixit A., Hao F., Mukherjee S. et al. *Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking.* 2013, August. P. 7–12.

25. Opendaylight: Towards a model-driven SDN controller architecture / Medved J., Varga R., Tkacik A., Gray K. *Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks.* 2014, June. P. 1–6.

26. Advanced study of SDN/OpenFlow controllers / Shalimov A., Zuikov D., Zimarina D. et al. *Proceedings of the 9th Central & Eastern European Software Engineering Conference in Russia.* 2013, Oct. P. 1–6.

27. What is software defined networking (SDN)? [Електронний ресурс] – Режим доступу: https://www.ibm.com/services/network/sdn-versus-traditional-networking

28. Сетевые технологии SDN [Електронний ресурс] – Режим доступу: https://habr.com/ru/company/muk/blog/251959/

29. Network function virtualization: Challenges and opportunities for innovations / Han B., Gopalakrishnan V., Ji L., Lee S. *IEEE Communications Magazine.* 2015. Vol. 53, Iss. 2. P. 90–97.

30. Network function virtualization (NFV) [Електронний ресурс] – Режим доступу: https://www.etsi.org/technologies/nfv

31. Infonetic research: Unified communications survey highlights enterprises' shift towards mobility [Електронний ресурс] – Режим доступу: https://www.cnbc.com/id/100134423

32. Ashwood-Smith P., Mohammadi M. A. A., Evelyne R. O. C. H. *U.S. Patent No. 9,847,915*. Washington, DC: U.S. Patent and Trademark Office, 2017.

33. A comprehensive survey of network function virtualization / Yi B., Wang X., Li K., Huang M. *Computer Networks.* 2018. Vol. 133. P. 212–262.

34. Li Y., Chen M. Software-defined network function virtualization: A survey. *IEEE Access.* 2015. Vol. 3. P. 2542–2553.

35. Survey on multi-access edge computing for internet of things realization / Porambage P., Okwuibe J., Liyanage M. et al. *IEEE Communications Surveys & Tutorials.* 2018. Vol. 20, Iss. 4. P. 2961–2991.

36. Multi-access edge computing: Open issues, challenges and future perspectives / Shahzadi S., Iqbal M., Dagiuklas T., Qayyum Z. U. *Journal of Cloud Computing.* 2017. Vol. 6, Iss. 1. P. 1–13.

37. Developing software for multi-access edge computing / Reznik A., Arora R., Cannon M. et al. *ETSI White Paper.* 2017. No. 20.

38. Multi-access edge computing: A survey / Tanaka H., Yoshida M., Mori K., Takahashi N. *Journal of Information Processing.* 2018. Vol. 26. P. 87–97.

39. Intelligent offloading in multi-access edge computing: A state-of-the-art review and framework / Cao B., Zhang L., Li Y. et al. *IEEE Communications Magazine.* 2019. Vol. 57, Iss. 3. P. 56–62.

40. Efficient next generation emergency communications over multi-access edge computing / Markakis E. K., Politis I., Lykourgiotis A. et al. *IEEE Communications Magazine.* 2017. Vol. 55, Iss. 11. P. 92–97.

41. Maksymyuk, Taras, Mykhailo Klymash, and Minho Jo. "Deployment strategies and standardization perspectives for 5G mobile networks." *2016 13th International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science (TCSET)*. IEEE, 2016.

42. Kekki, S., Featherstone, W., Fang, Y., Kuure, P., Li, A., Ranjan, A., ... & Scarpina, S. (2018). MEC in 5G networks. *ETSI white paper*, *28*, 1-28.

43. Software defined cooperative data sharing in edge computing assisted 5G-VANET / Luo G., Zhou H., Cheng N. et al. *IEEE Transactions on Mobile Computing*. 2019.

44. Joint node selection and resource allocation for task offloading in scalable vehicle-assisted multi-access edge computing / Pham X. Q., Nguyen T. D., Nguyen V., Huh E. N. *Symmetry.* 2019. Vol. 11, Iss. 1. P. 58.

45. Tran T. X., Pompili D. Joint task offloading and resource allocation for multi-server mobile-edge computing networks. *IEEE Transactions on Vehicular Technology.* 2018. Vol. 68, Iss. 1. P. 856–868.

46. Deep learning empowered task offloading for mobile edge computing in urban informatics / Zhang K., Zhu Y., Leng S. et al. *IEEE Internet of Things Journal.* 2019. Vol. 6, No. 5. P. 7635–7647.

47. James J., Verma B. Efficient VM load balancing algorithm for a cloud computing environment. *International Journal on Computer Science and Engineering.* 2012. Vol. 4, Iss. 9. P. 1658.

48. Mobile edge computing: A taxonomy / Beck M. T., Werner M., Feld S., Schimper S. *Proceedings of the Sixth International Conference on Advances in Future Internet.* 2014, November. P. 48–55.

49. Gavrilovska L., Rakovic V., Denkovski D. Aspects of resource scaling in 5G-MEC: Technologies and opportunities. *2018 IEEE Globecom Workshops (GC Wkshps).* 2018, December. P. 35–41.

50. End-to-end performance evaluation of MEC deployments in 5G scenarios / Virdis A., Nardini G., Stea G., Sabella D. *Journal of Sensor and Actuator Networks.* 2020. Vol. 9, Iss. 4. P. 57.

51. Collaborative mobile edge computing in 5G networks: New paradigms, scenarios, and challenges / Tran T. X., Hajisami A., Pandey P., Pompili D. *IEEE Communications Magazine.* 2017. Vol. 55, Iss. 4. P. 54–61.

52. Mec-idc: joint load balancing and power control for distributed internet data centers / Rao L., Liu X., Ilic M., Liu J. *Proceedings of the 1st ACM/IEEE International Conference on Cyber-Physical Systems.* 2010, April. P. 188–197.

53. Online geographical load balancing for energy-harvesting mobile edge computing / Wu H., Chen L., Shen C. et al. *2018 IEEE International Conference on Communications (ICC).* 2018, May. pp. 116974 - 117017.

54. Energy-efficient task offloading, load balancing, and resource allocation in mobile edge computing enabled IoT networks / Li S., Zhai D., Du P., Han T. *Science China Information Sciences.* 2019. Vol. 62, Iss. 2. P. 1–3.

55. Bouet M., Conan V. Geo-partitioning of mec resources. *Proceedings of the Workshop on Mobile Edge Communications.* 2017, August. P. 43–48.

56. Optimizing clustering algorithm in mobile ad hoc networks using genetic algorithmic approach / Turgut D., Das S. K., Elmasri R., Turgut B. *2002 Global Telecommunications Conference (GLOBECOM'02).* IEEE. 2002, November. Vol. 1. P. 62–66.

57. Mahmudy W. F., Mariana R. M., Luong L. H. Hybrid genetic algorithms for multi-period part type selection and machine loading problems in flexible manufacturing

system. *2013 IEEE International Conference on Computational Intelligence and Cybernetics (CYBERNETICSCOM).* IEEE. .2013, December. P. 126–130.

58. Devaraj D., Banu R. N. Genetic algorithm-based optimisation of load-balanced routing for AMI with wireless mesh networks. *Applied Soft Computing*. 2019. Vol. 74. P. 122–132.

59. Genetic algorithms in wireless networking: Techniques, applications, and issues / Mehboob U., Qadir J., Ali S., Vasilakos A. *Soft Computing.* 2016. Vol. 20, No. 6. P. 2467–2501.

60. Datta S. K., Bonnet C. MEC and IoT based automatic agent reconfiguration in Industry 4.0. *2018 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS).* 2018, December. P. 1–5.

61. Miller M. J., Vaidya N. H. A MAC protocol to reduce sensor network energy consumption using a wakeup radio. *IEEE Transactions on Mobile Computing*. 2005. Vol. 4, Iss. 3. P. 228–242.

62. A novel MAC scheduler to minimize the energy consumption in a Wireless Sensor Network / Anchora L., Capone A., Mighali V. et al. *Ad Hoc Networks.* 2014. Vol. 16. P. 88–104.

63. Ye W., Heidemann J., Estrin D. An energy-efficient MAC protocol for wireless sensor networks. *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies.* 2002, June. Vol. 3. P. 1567–1576.

64. Ranadheera S., Maghsudi S., Hossain E. Mobile edge computation offloading using game theory and reinforcement learning. 2017. (arXiv preprint arXiv:1711.09012).

65. A game-based computation offloading method in vehicular multiaccess edge computing networks / Wang Y., Lang P., Tian D. et al. *IEEE Internet of Things Journal.* 2020. Vol. 7, Iss. 6. P. 4987–4996.

66. Poularakis, K., Llorca, J., Tulino, A. M., Taylor, I., & Tassiulas, L. (2019, April). Joint service placement and request routing in multi-cell mobile edge computing networks. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications* (pp. 10-18). IEEE.

67. Deng M., Tian H., Lyu X. Adaptive sequential offloading game for multi-cell mobile edge computing. *2016 23rd International Conference on Telecommunications (ICT)*. 2016, May. P. 1–5.

68. Shakarami A., Shahidinejad A., Ghobaei-Arani M. A review on the computation offloading approaches in mobile edge computing: A game-theoretic perspective. *Software: Practice and Experience.* 2020. Vol. 50, Iss. 9. P. 1719–1759.

69. A cloud–MEC collaborative task offloading scheme with service orchestration / Huang M., Liu W., Wang T. et al. *IEEE Internet of Things Journal.* 2019. Vol. 7, Iss. 7. P. 5792–5805.

70. Li N., Martinez-Ortega J. F., Rubio G. Distributed joint offloading decision and resource allocation for multi-user mobile edge computing: A game theory approach. 2018. (arXiv preprint arXiv:1805.02182).

РОЗДІЛ 2
МЕТОД ОПТИМІЗАЦІЇ РОЗМІЩЕННЯ МАСШТАБОВАНИХ ПОСЛУГ НА РОЗПОДІЛЕНИХ ОБЧИСЛЮВАЛЬНИХ РЕСУРСАХ МЕРЕЖІ СТІЛЬНИКОВОГО ОПЕРАТОРА

Як вже було показано в першому розділі, розрахункові потужності на межі стільників – це доволі перспективна концепція в контексті розвитку Інтернету речей, особливо для підтримки залежних від затримки додатків. Основною з основних проблем при цьому є задача по розміщенню релевантних сервісів, яка стосується рішення в яке ж місце помістити декілька додатків згідно їх вимогам до якості надання послуг QoS, це з одного боку, та обчислювальної доступності ресурсу, з іншого боку. Тому в даному розділі дисертаційного дослідження досліджується розподіл навантаження та розміщення масштабованих сервісів IoT, щоб мінімізувати потенційне порушення вимог до їх якості обслуговування QoS через обмеження обчислювальних ресурсів. Зокрема, було сформовано задачу, яку було вирішено із використанням цілочисельного нелінійного програмування. Було запропоновано два підходи, один через методи лінеаризації і інший на основі генетичного алгоритму.

Структура даного розділу наступна:

– Розробка системної моделі, в якій декілька моделей додатків можуть бути розміщені на різних вузлах, а запити на ці додатки будуть розподілятись між моделями.

– Формулювання проблеми оптимізації рішення послуг у вигляді проблеми нелінійного програмування зі зміщеним цільовим числом з врахуванням, як призначення вузла (тобто місця розгортання додатку), так і розподілення навантаження для мінімізації можливого виникнення порушень QoS.

– Розробка підходів до вирішення поставленої задачі оптимізації.

– Оцінка продуктивності мобільної мережі з обчислювальними можливостями edge.

2.1. Модель мережі у хмарі

Поки що концепція MEC має певні обмеження. В тому числі це стосується окремих EC вузлів, що в своїй більшості є різнорідними і мають різні ресурси та можливості (наприклад, швидкість обробки, пам'ять та ресурсі зберігання інформації) порівняно з хмарними центрами обробки даних [1]. Таким чином, в практичному застосуванні, зазвичай не є рентабельним запускати всі додатки на EC. Зіштовхнувшись з цими обмеженнями, актуальна проблема, яка повинна бути вирішена, заключається в прийнятті рішення, де необхідно розміщувати певні додатки (тобто, на вузлі edge або ж всередині хмари) у відповідності з обмеженнями ресурсів інфраструктури, та вимогам до якості обслуговування додатків (QoS), іншим цілям. Ця проблема прийняття рішення знайома як проблема розміщення додатків або сервісів, яка являється нетривіальною проблемою з врахуванням розподіленої, динамічної і гетерогенно-граничного обчислювального оточення [2].

Деякі дослідження стосовно проблеми розміщення послуг в хмарних обчислювальних сервісах можна знайти в працях [3, 4]. Але все-таки ці запропоновані рішення не можуть бути безпосередньо застосовані до EC, оскільки вони не враховують вищезгаданої характеристики граничного обчислювального середовища (наприклад, гетерогенних розподілених мереж) і вимоги по часу відклику додатків, чутливих до затримок. Окрім того, існуючі роботи по розміщенню сервісів в EC [5, 6] мають обмеження. По-перше, деякі з них припускають, що на рівні edge достатньо ресурсів для всіх додатків/сервісів, які потребують обслуговування. В практичному ж застосуванні деякі додатки розгортаються на віддаленому хмарному центрі обробки даних, через обмеженість ресурсів на обладнанні типу edge, що в деяких випадках може привести до порушення вимог до максимально допустимої затримки для певного додатку. По-друге, деякі рішення не враховують ці вимоги до часу відклику або не вказують про час обробки і затримки. По-третє, вони не враховують вертикальне та горизонтальне масштабування додатків. Горизонтальне масштабування означає додавання або

видалення моделей додатку в системі. З іншого боку, вертикальне масштабування означає можливість додавання (або видалення) ресурсів в модель одного додатку. Нарешті, в більшості робіт не розглядається вплив робочого навантаження вузлів на виконання вимог QoS для додатків.

Виходячи з вищезгаданих фактів, в цьому розділі пропонується вирішувати проблему розміщення сервісів в edge. Отже, формулюється проблема, приймаючи до уваги обмеження до пропускної здатності мережі, інфраструктурні обмеження та характеристики додатків (вимоги QoS, потреба в ресурсах, масштабованість і робоче навантаження), щоб мінімізувати порушення QoS додатків (тобто часу, що перевищує допустиму затримку).

2.2 Розробка математичної моделі

2.2.1 Обчислювальна модель граничних обчислень

Узагальнений випадок використання мобільної мережі з граничними обчислюваннями можливостями (5G мережа) показаний на рис. 2.1.

В запропонованій схемі обчислювальні вузли можуть розміщуватись в різних частинах мережі, а не тільки на базових станціях мобільної мережі. В даному випадку мережа ЕС складається з різних вузлів – граничних та хмарних вузлів, пристроїв кінцевого користувача і ліній зв'язку, які з'єднують ці вузли між собою. Крім того, пристрої (стаціонарні чи мобільні) можуть бути з'єднані безпосередньо з деякими вузлами за допомогою проводової чи безпроводової лінії зв'язку.

Рис. 2.1. Архітектура мережі 5G із використанням MEC

Ці пристрої можуть потребувати надання певних послуг від віддалено розташованих додатків. Декілька додатків, головним чином служби Інтернету речей, можуть бути розгорнуті і виконані в різних обчислювальних вузлах, де інфраструктура і постачальники послуг визначають місця розгортування цих служб. Потім, запит, який відправляється кінцевим пристроєм, направляється між вузлами прямо до того вузла, на якому розміщено потрібний додаток. Після цього, запит обробляється, і його відповідь відправляється в зворотньому напрямку до пристрою, який надіслав цей конкретний запит.

В цьому випадку (рис. 2.1) додатки можуть розміщуватись на обчислювальних вузлах, що розташовані в областях мережі радіодоступу (RAN), базової мережі і

хмарних обчислень. Якщо додаток виконується на базовій станції (BS, eNb або gNb), то запит може бути направлений тільки між сусідніми базовими станціями, щоб зменшити трафік в ядрі та зменшити затримку передачі. Але не всі додатки можуть бути розгорнуті в БС через обмеження обчислювальних ресурсів в цьому регіоні. Саме тому деякі додатки розміщаються в ядрі або в хмарі, не порушуючи при цьому певних критеріїв розміщення від постачальника.

2.2.2. Модель мережі та її ресурсів

Розглянута вище мережа може бути представлена у вигляді однонаправленого графу $G = (v, \varepsilon)$, де вершини v – вузли мережі, а ε – це мережеві канали між вузлами (рис. 2.2).

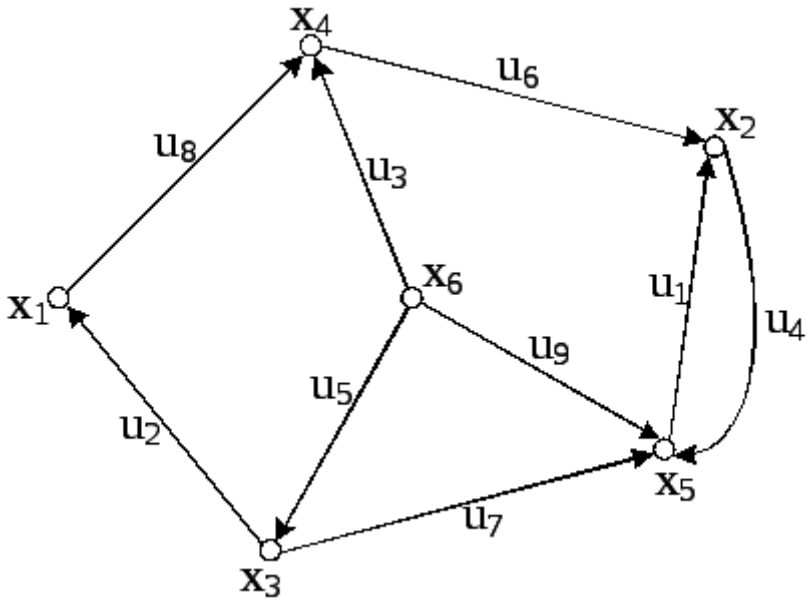


Рис. 2.2. Неорієнтований граф

Припустимо, що кожна вершина доступна будь-якій іншій вершині в графі через декілька послідовних передач (хопів). Окрім того, пристрої кінцевих користувачів і їх з'єднання не представлені в G. При цьому, кожне мережеве з'єднання $e \in \varepsilon$ мають наступні властивості:

– **Затримка передачі** $D_{net}^{a,e}$ **–** це час, який потрібен для виконання запиту, що надійшов від додатку *a*, який буде переданий в канал *e.*

Запропонована модель визначає різні типи ресурсів R. Наприклад, набір $R = \{CPU, RAM, Storage\}$ складається з обчислювальної потужності процесора, оперативної пам'яті (RAM) і ресурсів дискового простору. ОЗП та дисковий простір може вимірюватись в байтах, в той час як ЦП може вимірюватись в циклах на секунду (тактова частота в Гц) або виконання інструкцій в секунду (IPS).

Вузол або вершина на рис. 2.2 представляє собою сервер зі спеціальними можливостями і ресурсами для запуску додатків. Нехай декілька серверів можуть бути розташовані локально на одному мережевому вузлі, ми дивимось на ці вузли як на декілька серверів одного блоку. Саме тому є точно один сервер для кожного мережевого вузла. Ми використовуємо умови того, що вузол і сервер є взаємозамінними поняттями. Також з одним й тим самим сенсом використовуються терміни «додаток», «сервіс» або «послуга».

Кожен вузол $v \in V$ має наступні характеристики:

– **Ємність ресурсу** $C_{v,r}$ – число, що описує загальну ємність ресурсу $r \in R$ на вузлі $:v$; при цьому приймаємо, що хмара має потенційно необмежені ресурси, $C_{cloud,r} = \infty$, через різницю в ємності між хмарою і граничним вузлом.

Нехай A – це набір всіх різних додатків для розміщення по мережі. Тоді, додаток $a \in$ A має наступні параметри:

– **Максимальна допустима затримка** D_a – параметр, що вказує на максимальний час, який є допустимим на отримання відповіді для додатку a. Час відгуку включає в себе затримку мережі і затримку на обробку.

– **Максимальне число екземплярів додатку** N_a – значення, яке показує, як додаток масштабується по горизонталі.

– **Попит на ресурс** $f_a^r(\lambda)$ – функція, яка визначає середній об'єм ресурсів $r \in R$, які споживаються додатком з робочим навантаженням $\lambda \geq 0$. Ми визначаємо це робоче навантаження λ як середній коефіцієнт отримання запиту в екземплярі *a*. Таким чином, ця функція описує, як додаток масштабується по вертикалі.

– **Необхідна обчислювальна потужність** W_a – значення, що вказує на середній об’єм потужності, який необхідний для отримання відповіді на запит a. Вимірюється числом команд або тактів необхідних для повноцінного виконання запиту.

– **Частота запитів** λ_a – це середня швидкість генерації запитів a для кожного пристрою кінцевого користувача, який робить запит до цього додатку. В нашому випадку вона визначається розподіленням Пуассона [7, 8].

Звертаємо увагу на те, що пристрої кінцевих користувачів або самі користувачі не знають, де розгорнуті додатки допоки екземпляр додатку буде обробляти їх запити. Тому ми можемо розділити ці запити між декількома копіями, які розміщені в системі. Для простоти, припускаємо, що кожен користувач робить запит тільки до одного додатку. Тоді нехай U_a^v число користувачів підключених до вузла v зі зверненням до додатку a, U_a загальна кількість користувачів, які виконують звернення до додатку a в системі.

2.3 Розробка методу оптимізації розміщення масштабованих послуг на розподілених обчислювальних ресурсах мережі

В практичному сценарії неможливо розмістити всі додатки на граничних вузлах з урахуванням обмеженості ресурсів вузлів ЕС. З цього слідує, що деякі додатки розгортаються додатково (тобто в базовій мережі або хмарі) від кінцевих користувачів. Ця значна відстань між вузлом та користувачем може привести до збільшення затримки відгуку на запит. Більше того, перевантажений сервер також збільшує час відклику, а розподілення навантаження між копіями додатків можуть зменшити цю проблему. Тому, вивчаємо сумісну проблему розміщення і завантаження сервісів розподілення для мінімізації можливих порушень QoS (максимальна допустима затримка).

Надалі представлено послідовність кроків для оцінки часу відповіді на запит.

Визначаємо потік звернень $F_a^{u,v}$ як запити на копію додатку $a \in \mathrm{A}$, що розміщується на вузлі $v \in V$ (цільовий вузол) і генеровані користувачем, які підключені до вузла $u \in V$ (вихідний вузол). Таким чином , (2.1) визначає середній час відклику потоку запитів $F_a^{u,v}$, де d_{net} – середній час відправки запитів від користувачів a до вузлу u, a – середній час обробки запитів по v. При цьому, мережева затримка обробки оцінюється наступним чином.

$$\overline{d}(F_a^{u,v}) = \overline{d}_{net}(F_a^{u,v}) + \overline{d}_{proc}(a,v) \tag{2.1}$$

1) ***Мережева затримка:***

Мережева затримка відгуку на запити включає в себе:

– Затримку між кінцевим пристроєм, що робить запит, та вузлом, до якого він приєднаний;

– Затримка передачі з цього останнього вузла на сервер, де розміщено додаток, який слідує транзитним маршрутом.

Важливо відмітити, що вузол обробки може приймати звернення і обробляти їх, а тому відповідно і затримка передачі другої частини вищенаведеної рівності дорівнюють нулю. Оскільки ми перевіряли проблемні випадки в автономному режимі, затримка зв’язку між пристроями і його підключеним вузлом не впливає на рішення проблеми [12]. Тому ми не розглядаємо цю складову затримки в оцінці загальної затримки мережі. Тому можемо оцінити середню мережеву затримку потоку запитів $F_a^{u,v}$ як:

$$\overline{d}(F_a^{u,v}) = D_{net}^{a,u,v} = \begin{cases} 0 & \textit{якщо} \quad u = v, \\ \sum_{e \in P_{u \to v}} D_{net}^{a,e} & \textit{інакше.} \end{cases} \tag{2.2}$$

де $P_{u \to v}$ набір зв'язків на шляху в напрямку від u до ϑ. Цей набір може бути визначений деяким алгоритмом маршрутизації, таким як, наприклад, алгоритм Флойда [9].

2) ***Модель затримки:*** Обробка запиту моделюється з використанням моделі масового обслуговування M/M/1. В цій моделі користувачів безперервно виробляються запити щодо застосування відповідного гомогенно-пуасонівского процесу з відношенням A. Коефіцієнт надходження запитів для додатку, що

виконується на вузлі v визначається як сума всіх запитів, які надходять на цей вузол. Еквівалентно (2.3) виражає цю частоту надходження запиту, де $\delta_a^{u,v} \in \left[0, Q_a^u\right]$ – цілочисельна змінна, яка вказує розмір потоку запитів F_u (тобто кількість запитів в потоці) і $Q_a^u = \left[U_a^u, \lambda_a\right]$ є кількість запитів, що створені користувачами, які підключені до u.

$$\lambda_a^u = \sum_{u \in v} \delta_a^{u,v} \tag{2.3}$$

Час обслуговування має експоненціальне розподілення з параметром μ, де $1/\mu$= середній час обслуговування в черзі M/M/1. Таким чином, ми виражаємо $1/\mu_a^h$ = має час для виконання роботи W_a ЦП запиту з ресурсами, які виділені для копії додатку a у вузлі v, як:

$$\frac{1}{\mu_a^v} = \frac{W_a}{f_a^{CPU}(\lambda_a^v)} \tag{2.4}$$

У результаті, (2.5) середній час обробки запитів для виконування запиту на вузлі *v* у відповідності до закону Літтла.

$$\bar{d}_{proc}(a,v) = \frac{1}{\mu_a^v - \lambda_a^v} \tag{2.5}$$

Слід відзначити, що вирішення проблеми обмеження автономного розміщення служби можливо тільки при виконанні всіх наступних вимог:

1. Кількість екземплярів: вузол може розміщувати тільки одну репліку даного додатку. Окрім того, кількість екземплярів, які розгорнуті в системі, повинна відповідати обмеженням, які визначені додатками і всі вони мають бути розміщені.

$$1 \le \sum_{v \in V} p_a^v \le N_a \qquad \forall_a \in \mathrm{A} \tag{2.6}$$

2. Існування потоку запитів: потік запитів існує тільки в тому випадку, якщо копія додатку *a* розміщена в *v* і є користувачі, які підключені в *u* з запитом *a.* Нехай $\gamma_a^{u,v} \in \{0,1\}$ – двійкова змінна, яка виражає існування потоку $F_a^{u,v}$.

$$\gamma_a^{u,v} \le p_a^v Q_a^u \qquad \forall_a \in \mathrm{A}, \forall_{u,v} \in V \tag{2.7}$$

3. Розмір потоку запиту: якщо потік існує, його розмір повинен бути, принаймні, одним і дорівнювати кількості запитів

$$\gamma_a^{u,v} \le \delta_a^{u,v} \le \gamma_a^{u,v} Q_a^u \qquad \forall_a \in \mathrm{A}, \forall_{u,v} \in V \tag{2.8}$$

4. Збереження навантаження: сукупний розмір всіх потоків запитів для застосування a від того ж вихідного вузла u дорівнює загальній кількості запитів від користувачів, пов'язаних з цим вузлом.

$$\sum_{v \in V} \delta_a^{u,v} = Q_a^u \qquad \forall_a \in \mathrm{A}, \forall_u \in V \tag{2.9}$$

5. Продуктивність вузла: загальна сума ресурсів, що запитуються додатками, не повинна перевищувати продуктивності вузлів.

$$\forall_r \in R, \forall_v \in V \tag{2.10}$$

6. Стабільність черги: черга в моделі M/M/1 стабільна, тільки якщо середній темп обслуговування більше, ніж середній темп надходження запитів. Ця стабільність повинна бути гарантована для кожного додатку, поміщеного у вузол.

$$\lambda_a^v < \mu_a^v \qquad \forall_{a,v} (p_a^v = 1), \quad a \in \mathrm{A}, v \in V \tag{2.11}$$

7. Порушення QoS: час відклику існуючого потоку не повинен перевищувати крайній термін свого застосування плюс рівень порушення QoS системи

$$\gamma_a^{u,v} \overline{d}(F_a^{u,v}) \le D_a + \varepsilon \qquad \forall_a \in \mathrm{A}, \forall_{u,v} \in V \tag{2.12}$$

Ми визначаємо рівень порушення QoS потоку запиту як відмінність між його середнім часом відгуку і крайнім терміном виконання.

Рівень порушення QoS системи – найвищий рівень порушення серед усіх потоків в системі $(\overline{d}(F_a^{u,v}) - D_a)$. Тому основна мета, яка переслідується в даному розділі, полягає в тому, щоб мінімізувати рівень порушення QoS. Тому поставлена сервісна проблема розміщення може бути сформульована наступним чином, враховуючи обмеження (2.6)-(2.12):

$$\min(\varepsilon) \tag{2.13}$$

В таблиці 2.1 представлені основні параметри, які використовуються в даній запропонований моделі.

Таблиця 2.1 – Позначення, що використовуються в запропонованій системній моделі

Символ	Опис
Вхідні параметри	
$V, \varepsilon, R, \text{A}$	набір мережевих вузлів, мережевих посилань, типів ресурсів та програм відповідно
$C_{v,r}$	загальна ємність ресурсу r на вузлі v
D_a	максимально допустимий час відгуку програми a
N_a	максимальна кількість копій для програми a
$f_a^r(\lambda)$	попит ресурсу r на копію програми a з робочим навантаженням λ
$K_1^{a,r}, K_2^{a,r}$	константи лінійного попиту на ресурси для програми a та ресурсу r, $f_a^r(\lambda) = K_1^{a,r}\lambda + K_2^{a,r}$
W_a	розмір роботи CPU запиту про програму a
λ_a	швидкість генерації запитів для програми a
U_a^v	кількість користувачів, підключених до вузла v, що вимагає додаток a
$D_{net}^{a,u,v}$	затримка мережі для програми a між вузлами u і v
Змінні	
p_a^v	будь-який вузол v розгортає екземпляр програми a чи ні
$\gamma_a^{u,v}$	незалежно від того, чи існує потік $F_a^{u,v}$ запитів, чи ні
$\delta_a^{u,v}$	кількість запитів у потоці $F_a^{u,v}$
ε	рівень порушення якості системи
Інші	
$F_a^{u,v}$	потік запитів від користувачів, підключених до вузла u, до екземпляра програми a, розгорнутої на вузлі v

λ_a^v	запитує швидкість прибуття програми a на вузол u
μ_a^v	швидкість обслуговування програмиа a на вузлі u
Q_a^v	кількість запитів на генерацію програми a від користувачів, підключених до вузла u
Q_a	загальна кількість запитів на додаток a в системі

Проблемою оптимізації (2.13) є змішане ціле число (MINLP), при тому що обмеження (2.10) – (2.12) є нелінійними. MINLP зазвичай важко вирішити через його високу обчислювальну складність [10]. Один із можливих способів зменшити цю складність полягає в тому, щоб застосувати методи релаксації і лінеаризації. Тому ми перетворюючи (2.13) в проблему міксованого цілочисельного програмування Mixed INTEGER Linear Programming (MILP), можемо використати ці методи до наступних нелінійних обмежень:

Пропускна здатність вузла. Для деяких заявок функція попиту на ресурс може бути нелінійною. У цьому випадку це може бути замінене по лінійному оцінювачу $f^{*r}_a(.)$ в інтервалі області $[0, Q_a]$, як показано в (2.13) де $K_1^{a,r}, K_2^{a,r}$, є константами, і що рівне $\sum_{v \in V} Q_a^v$.

$$f^{*r}_a(\lambda) = K_1^{a,r}, K_2^{a,r} \tag{2.14}$$

Враховуючи, що запити тільки прибувають в сервери, запускають запитаний додаток згідно з (2.3), (2.7) і (2.8), ми маємо:

$$p_a^v \lambda_a^v = \lambda_a^v \tag{2.15}$$

Застосовуючи (2.14) і (2.15) до (2.10), повне обмеження вузла може бути переписано як:

$$\sum_{a \in \mathrm{A}} (\lambda_a^v K_1^{a,r} + p_a^v K_2^{a,r}) \le C_{v,r} \qquad \forall_r \in R, \forall_v \in V \tag{2.16}$$

Стабільність черги. Щоб отримати стандартну форму проблеми MILP, це має вилучити строгість нерівності в (2.11). Для цього ще додається константа $\Theta \approx 0$. Крім того, обидві сторони нерівності помножені на p_a^v, щоб гарантувати обмеження

існування черги. Звертаючись до (2.4), (2.14) і (2.15) до цього результату, ми далі маємо:

$$\lambda_a^v(K_1^{a,CPU} - W_a) + \begin{matrix} p_a^v K_2^{a,CPU} \geq p_a^v \Theta \\ \forall_a \in \mathrm{A}, \forall_v \in V \end{matrix} \tag{2.17}$$

Порушення QoS. Враховуючи рівняння (2.1), (2.2), (2.4), (2.5) і (2.14), ми можемо записати обмеження (2.12) як:

$$(\gamma_a^{u,v}\lambda_a^v D_{net}^{a,u,v} - \varepsilon\lambda_a^v - \lambda_a^v D_a)(K_1^{a,CPU} - W_a) + \gamma_a^{u,v}(K_2^{a,CPU} D_{net}^{a,u,v} + W_a) - K_2^{a,CPU}(D_a + \varepsilon) \leq 0 \qquad \forall_a \in \mathrm{A}, \forall_{u,v} \in V \tag{2.18}$$

Однак у (2.18) $\gamma_a^{u,v}\lambda_a^v$ та $\varepsilon\lambda_a^v$, обидві умови є білінеарні.

Ми можемо спростити ці умови, щоб отримати лінійні вирази, використовуючи конверти [11]. Таким чином, ми замінюємо ці білінеарні умови $\varphi_a^{u,v} = \gamma_a^{u,v}\lambda_a^v$ та $\psi_a^v = \varepsilon\lambda_a^v$ новими змінними і додаємо наступні нові обмеження в проблемі:

$$0 \leq \gamma_a^{u,v} \leq 1 \text{ і } 0 \leq \lambda_a^v \leq Q_a \text{ і } 0 \leq \varepsilon \leq \mathrm{E} \tag{2.19а}$$

$$0 \leq \varphi_a^{u,v} \leq \lambda_a^v \text{ і } Q_a(\gamma_a^{u,v} - 1) + \lambda_a^v \leq \varphi_a^{u,v} \leq \gamma_a^{u,v} \tag{2.19б}$$

$$0 \leq \psi_a^{u,v} \leq \lambda_a^v \mathrm{E} \text{ і } \varepsilon Q_a + \lambda_a^v \mathrm{E} - \mathrm{E} Q_a \leq \psi_a^v \leq \varepsilon Q_a \tag{2.19в}$$

де E – постійне визначення максимального рівня дозволеного порушення QoS. Потім ми можемо переписати (2.18) з двома новими змінними, щоб мати лінійне обмеження:

$$(\varphi_a^{u,v} D_{net}^{a,u,v} - \psi_a^u - \lambda_a^v D_a)(K_1^{a,CPU} - W_a) + \gamma_a^{u,v}(K_2^{a,CPU} D_{net}^{a,u,v} + W_a) - K_2^{a,CPU}(D_a + \varepsilon) \leq 0 \qquad \forall_a \in \mathrm{A}, \forall_{u,v} \in V \tag{2.20}$$

Нарешті, ми формулюємо проблему MILP наступним чином: (2.6) – (2.9), (2.16), (2.17), (2.19) та (2.20).

Важливо відзначити, що рішення (2.21) також виконується для (2.13), але це може уявити більш високу об'єктивну вартість, коли ставиться оригінальна проблема через білінеарну релаксацію.

Хоча відомі рішення, такі як CPLEX [12], можуть вирішити проблеми MILP, ці проблеми, як правило, мають NxP складність [13]. Більше того, (21) дуже трудомісткий через велику кількість цілих змінних. Отже, в даній роботі пропонується евристичне рішення, засноване на генетичних алгоритмах. Перевагою генетичного підходу є те, що він не обмежується опуклими або лінійними

проблемами [14]. Запропонований генетичний алгоритм використовує упереджені хромосоми з випадковими ключами [15], що є масивом випадково генерованих реальних чисел у інтервалі [0,1]. Це представлення хромосом використовується не для створення нездійсненних рішень, які можуть погіршити продуктивність генетичного алгоритму [16]. Дана пропозиція використовує елітарну стратегію, зберігаючи елітарних індивідів, тобто тих, хто має найкращі цінності, наступному поколінню. Він також додає нові випадкові покоління індивідів, амутантів, у наступному поколінні. Крім того, для поповнення популяції породження створюється параметризований рівномірний кросовер [17] між елітарними та неелітарними особами.

У алгоритмах з випадковим ключем випадковий алгоритм детермінованого декодера приймає хромосому індивіда та обчислює її значення. Таким чином, представлення хромосоми та алгоритму декодера відіграє важливі правила в розробленій моделі. Наша пропозиція щодо кодування хромосоми та опис її частин наведені нижче:

$$C = [\mathrm{O}_1^1, \mathrm{O}_1^2, ..., \mathrm{O}_1^{|V|}, ..., \mathrm{O}_{|\mathrm{A}|}^1, \mathrm{O}_{|\mathrm{A}|}^2, ..., \mathrm{O}_{|\mathrm{A}|}^{|V|} \quad \mathrm{M}_1, \mathrm{M}_2, ..., \mathrm{M}_{|\mathrm{A}|}, \quad V_1^1, V_1^2, ..., V_1^{|V|}, ..., V_{|\mathrm{A}|}^1, V_{|\mathrm{A}|}^2, ..., V_{|\mathrm{A}|}^{|V|}] \tag{2.21}$$

1. $\mathrm{O}_1^1, ..., \mathrm{O}_{|\mathrm{A}|}^{|V|}$ – це порядок створення запитів потоку.

2. $\mathrm{M}_1, ..., \mathrm{M}_{|\mathrm{A}|}$ – описує вагу, яка використовується при виборі сервера для розміщення програми.

3. $V_1^1, ..., V_{|\mathrm{A}|}^{|V|}$ – параметри для обчислення пріоритету вузла, який буде обраний як місце для розгортання програми. Цей пріоритет вузла v для програми a задається як:

$$M_a^v V_a^v + (1 - M_a^v) \frac{D_{net}^{a,u,cloud} - D_{net}^{a,u,v}}{D_{net}^{a,u,cloud}} \tag{2.21}$$

– Алгоритм роботи декодера представлений на рис. 2.3.

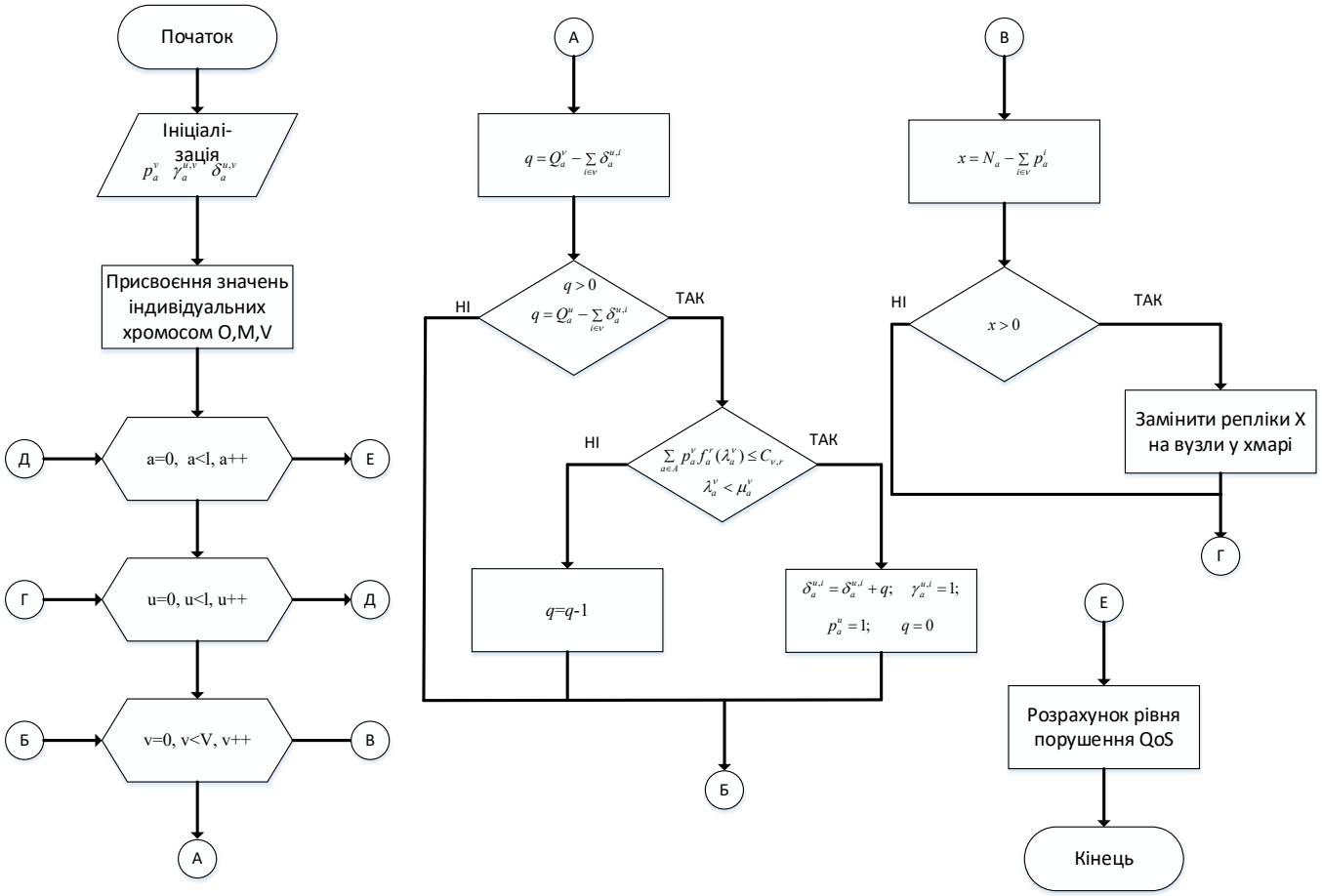


Рис. 2.3. Алгоритм роботи декодера

Просте рішення проблеми розміщення сервісу розгортає додаток на найближчих серверах до користувачів цього додатку, тобто серверів з меншою затримкою мережі для цих користувачів. Однак, обмеження ємності вузлів заважає цій схемі розгортання оптимально працювати для великої кількості програм або користувачів. Щоб покращити це рішення, пропонується включити ще один параметр в доповнення до затримки мережі в процедурі вибору вузла. Крім того, ваговий коефіцієнт (M) врівноважує ці два параметри в процесі прийняття рішення.

Ми розробили алгоритм 1 (рис. 2.3), ґрунтуючись на ідеях, які були представлені вище. У його зовнішньому циклі (рядок 6) відбувається ітерація над усіма можливими джерелами запиту, де перша частина (*i.e.*, $O_a^u part$) хромосоми визначає порядок циклу. Потім у рядку 8 він перевіряє всі можливі цілі, упорядковані специфікацією в (2.22). У самому внутрішньому циклі він намагається виділити

максимальну кількість запитів до обраного цільового вузла, при цьому, враховуючи обмеження (2.10) та (2.11). Важливо зазначити, що цей цикл є цілим завдяки припущенню необмежених ресурсів хмарного вузла. Якщо кількість реплік перевищує максимально дозволену, алгоритм здійснює локальну пошукову оптимізацію, замінюючи надлишки реплік хмарою. Нарешті, він обчислює рівень порушення QoS і повертає цей рівень як значення точності для вхідного індивіда.

Висновки до розділу 2

Дослідження, проведені у другому розділі, надали змогу отримати наступні результати.

1. Було досліджено проблему розміщення служб IoT, що підтримують горизонтальне та вертикальне масштабування в обчислювальному середовищі із можливістю забезпечення граничних обчислень.

2. Було сформульовано задачу цілочисельного лінійного програмування. Для її вирішення були запропоновані методи лінеаризації та генетики.

Список використаних джерел у другому розділі

1. Gavrilovska L., Rakovic V., Denkovski D. Aspects of resource scaling in 5G-MEC: Technologies and opportunities. *2018 IEEE Globecom Workshops (GC Wkshps).* 2018, December. P. 1–6.

2. A competitive approximation algorithm for data allocation problem in heterogenous mobile edge computing / Shao X., Liu Z., Dong M. et al. *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring).* 2019, April. P. 1–5.

3. Liu B., Liu C., Peng M. Resource allocation for energy-efficient MEC in NOMA-enabled massive IoT networks. *IEEE Journal on Selected Areas in Communications*. 2020.

4. Energy-aware mobile edge computation offloading for IoT over heterogenous networks / Li S., Tao Y., Qin X. et al. *IEEE Access.* 2019. Vol. 7. P. 13092–13105.

5. Arnold A. Mec: A system for constructing and analysing transition systems. *International Conference on Computer Aided Verification.* Berlin, Heidelberg: Springer, 1989, June. P. 117–132.

6. Power-delay trade-off for heterogenous cloud enabled multi-UAV systems / Duan R., Wang J., Du J. et al. *IEEE International Conference on Communications (ICC).* 2019, May. P. 1–6.

7. Філоненко Н. В. Розподіл пуассона в історичному вимірі. *Сімнадцята міжнародна наукова конференція імені академіка Михайла Кравчука.* Київ: НТУУ «КПІ», 2016. С. 331.

8. Пасічник В. В., Іванущак Н. М. Дослідження та моделювання складних мереж. *Eastern-European Journal of Enterprise Technologies.* 2010. Vol. 2, No. 3. P. 43–48.

9. Изотова Т. Ю. Обзор алгоритмов поиска кратчайшего пути в графе. *Новые информационные технологии в автоматизированных системах*: материалы девятнадцатого научно-практического семинара. Москва: ИПМ им. М. В. Келдыша, 2016. 352 с.

10. Sargent R. W. My contribution to broadening the base of chemical engineering. *Annual review of chemical and biomolecular engineering.* 2011. Vol. 2. P. 1–7.

11. Bussieck M. R., Vigerske S. MINLP solver software. *Wiley encyclopedia of operations research and management science*. 2010.

12. Path planning via CPLEX optimization / Ademoye T. A., Davari A., Castello C. C. et al. *2008 40th Southeastern Symposium on System Theory (SSST)*. 2008, March. P. 92–96.

13. Car relocation for carsharing service: Comparison of CPLEX and greedy search / Zakaria R., Dib M., Moalic L., Caminada A. *2014 IEEE Symposium on Computational Intelligence in Vehicles and Transportation Systems (CIVTS)*. 2014, December. P. 51–58.

14. Погорілий С. Д., Білоус Р. В. Генетичний алгоритм розв'язання задачі маршрутизації в мережах. *Проблеми програмування*. 2010. № 2–3. С. 171–177.

15. Войтюк І. Ф., Дивак М. П., Неміш В. М. Метод та генетичний алгоритм структурної ідентифікації інтервальних різницевих операторів в задачах екологічного моніторингу. *Наукові праці Донецького національного технічного університету. Серія: Інформатика, кібернетика та обчислювальна техніка: всеукр. наук. зб. Покровськ: ДонНТУ*, 2011. Вип. 14. С. 8–17.

16. Ладієва Л. Р., Зав'ялова Т. П. Оптимізація плівкового апарату роторного типу за максимальною продуктивністю. *Автоматика. Автоматизация. Электротехнические комплексы и системы.* 2007. № 2. С. 124–130.

17. Генетичні алгоритми вирішення задач управління ресурсами і навантаженням центрів оброблення даних / Теленик С. Ф., Ролік О. І., Букасов М. М., Андросов С. А. *Автоматика. Автоматизация. Электротехнические комплексы и системы.* 2010. № 1. С. 106–120.

РОЗДІЛ 3
МЕТОДИ ОПТИМІЗАЦІЇ ГРАНИЧНИХ ОБЧИСЛЕНЬ В СТІЛЬНИКОВИХ МЕРЕЖАХ

Очікування щодо високої якості досвіду (QoE) широко зростають з недавнім просуванням мереж 5G, що відкриває шлях до широкого нового набору послуг, таких як розширена / віртуальна реальність, безпека руху, розпізнавання зображень та облич тощо [1, 2]. QoE, орієнтований на користувачів, що включає, але не обмежується, ультранизькою затримкою (тобто 5 мс), надвисокою надійністю (тобто 99,999%) та підтримку в 1000 разів більших обсягів даних [1 – 3]. Потреба у підтримці більших обсягів даних виникає внаслідок надзвичайного передбачуваного збільшення кількості пристроїв Інтернету речей (IoT).

Обмежені ресурси для обчислення та зберігання користувальницького обладнання (UE) та IoT-пристроїв, окрім обмеженого ресурсу акумулятора, роблять їх непридатними для підтримування обробки оброблюваних ресурсами мобільних та IoT-додатків, що надають послуги 5G, такі як онлайн ігрові програми та програми розпізнавання обличчя / мови тощо [4, 5]. Одним з можливих рішень для подолання цих обмежень є використання мобільних хмарних обчислень, що дозволяє UE завантажувати оброблювані ними обчислювальні завдання для обробки додатків, розміщених на потужній централізованій віддаленій хмарі, доступній через Інтернет або в основній мережі мобільного оператора, таким чином, розширюючи можливості мобільних пристроїв [2, 6]. Однак, такий підхід спричиняє великі затримки зв'язку через віддалене місце розташування віддаленої хмари від користувача, що порушує вимоги QoE для деяких додатків, чутливих до затримки в реальному часі. Отже, для забезпечення швидкої доставки послуг, що відповідає контексту в режимі реального часу, було запропоновано використовувати граничні обчислення, що можуть розташовуватись значно ближче до кінцевого користувача [7].

Отже, ефективне використання ресурсів граничних мобільних обчислень необхідне для гарантування передбачених переваг, які тісно пов'язані з вирішенням наступних завдань:

1. Проблема розвантаження задач, яка полягає у визначенні серверів, на які слід розвантажувати задачі.

2. Проблема розподілу ресурсів додатків, яка визначає обчислювальні ресурси, які повинні бути розподілені для кожної програми, розгорнутої на граничному сервері, щоб обробити всі призначені їм завдання в межах їхніх вимог затримки.

3. Планування завдань, що визначає порядок, в якому кожне завдання має бути оброблене в спільній програмі, дотримуючись вимог до часу обробки.

Саме ці завдання вирішувались в даному розділі дисертаційної роботи. Крім того, після процесу розвантаження та планування завдань вирішувалось завдання оптимізації використання обчислювальних ресурсів та енергії в стільникових мережах під час проведення граничних обчислень.

Вже багато алгоритмів було запропоновано для підвищення продуктивності у подібних системах. Проте, дослідження контролю живлення у цих системах тільки починаються. Контроль живлення у цих системах, як одно-користувацьких (single-user) так і вільних від завад багатокористувацьких, був досліджений; проте у вільних від завад багатокористувацьких МПО системах ця проблема не була вивчена детально. Тому в даному розділі будуть проводитись дослідження, присвячені вирішенню саме цієї вищеокресленої проблеми.

3.1. Модель мережі 5G з з можливістю проведення граничних обчислень

3.1.1. Аналіз архітектури мережі

Методи, що досліджуються в існуючих роботах, є в основному рішеннями, які не досліджують переваги, які можуть бути принесені завдяки включенню динамічних модифікацій обчислювальних ресурсів, що виділяються спільним додаткам IoT, та їх впливу на планування завантажених завдань.

У цьому розділі ми дотримуємось більш цілісного підходу до розв’язання завдань із спільним розподілом ресурсів та плануванням, з особливим акцентом на IoT-послуги, що є чутливими до затримки [8].

Ефективне використання ресурсів тягне за собою розумне впорядкування обміну ресурсами, яке неможливо здійснити без кваліфікованого планування їх використання. Отже, щоб максимізувати дохід власника інфраструктури, було розроблено метод оптимізації Ляпунова для вирішення проблеми розміщення та планування віртуальних машин під час обліку Угоди про рівень сервісу (SLA) для критично важливих для часу послуг [9]. Цей підхід до планування враховує планування кількості та тип (малих, середніх та великих) обчислювальних ресурсів, які потрібно розгортати в кожному часовому слоті для кожного оператора мобільних послуг, виходячи з мінливості його завантаженості. Аналогічно, автори [10] використовували функцію Ляпунова, щоб вирішувати схеми розвантаження задач, при цьому стохастично максимізували корисність мережі під частково застарілою інформацією про стан мереж без будь-якого врахування вимог щодо затримки виконання/вирішення завдань. Автори роботи [11] спільно оптимізували завдання розвантаження та планування задачі разом із задачею розподілу потужності передачі. Вони були зацікавлені у мінімізації зваженої суми затримки виконання та споживання енергії на обслуговування. Вони розклали цю проблему на завдання розвантаження та планування завдання, які вони вирішують за допомогою алгоритму Джонсона [12] з метою мінімізації обсягу всіх завдань та проблеми розподілу потужності передачі, яку вони вирішують за допомогою опуклої оптимізації.

Проблема призначення та планування завдань була досліджена в [13, 14], де також було представлено алгоритмічне рішення для вирішення задачі, враховуючи терміни виконання завдань та планування їх передачі та обчислення. Автори роботи [15] вирішили проблему розвантаження задач, враховуючи можливість відправки завдань у віддалену хмару, а також у MEC. Крім того, вони представили попереджувальне планування для завантажених завдань з метою мінімізації їх зваженого часу відповіді за допомогою онлайн-алгоритму.

3.1.2. Модель запропонованої системи

В даному розділі розглядається інтелектуальна мережа типу WAN, як зображено на рис. 3.1, що складається з набору S базових станцій стільникового зв'язку, які можуть бути представлені або макростільниками (eNB), або маленькими стільниками (SCeNB).

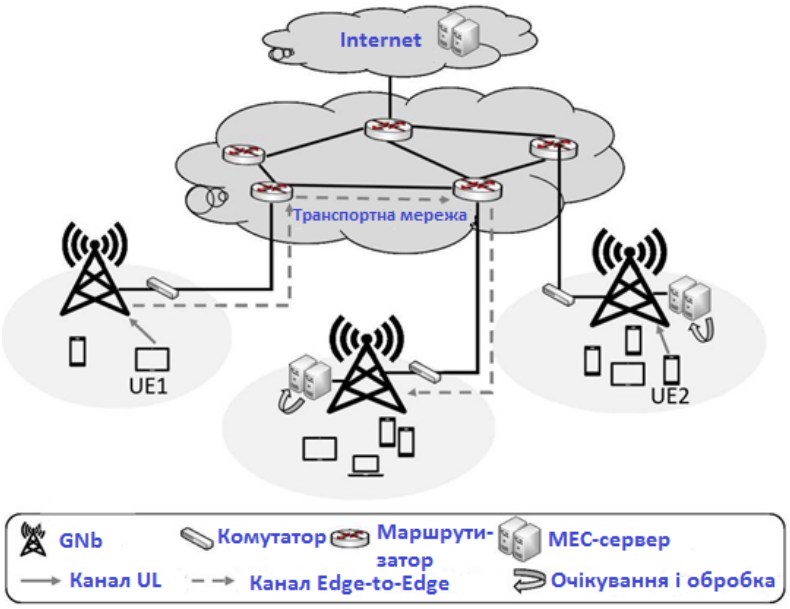


Рис. 3.1. Стільникова мережа із забезпеченням можливості граничних обчислень

Для спрощення представлятимемо, що всі базової станції однакові. Для того, щоб увімкнути гнучку маршрутизацію та зв’язок між БС, ми вважаємо, що основна стільникова мережа розгорнута із використанням технології SDN. SDN – нова парадигма, яка спрощує управління мережею шляхом логічної централізації логіки управління в одному централізованому об'єкті, який називається контролером SDN. Більш детально технологія SDN була розглянута в першому розділі даної дисертаційної роботи. Таким чином, ми вважаємо стільникову мережу на основі SDN, включену контролерами SDN та комутаторами OpenFlow [16]. Контролери SDN користуються глобальним уявленням про мережу і можуть використовуватися для надання деякої інформації, заснованої на моніторингу, наприклад, затримки,

переживаної потоком між двома БС. Підмножина eNB в S встановлена на серверах MEC для надання послуг вивантаження обчислень користувачам IoT (тобто обладнання користувачів (UE), такі як планшети та інші носимі пристрої) (рис. 3.1). Ми вважаємо, що система, що підтримує MEC, працює в часовій структурі, де ми позначаємо два часових проміжки.

Нехай M – це набір розгорнутих серверів MEC; кожен MEC-сервер складається з пулу фізичних серверів з агрегованою обчислювальною ємністю, зазначеної в циклах / секунді або МГц. Сервери MEC розміщують набір A IoT-програм декількох типів (наприклад, розпізнавання облич, кодування відео тощо), призначених для обробки завантажених завдань UE. Кожна програма A – це програмне забезпечення, яке може бути розгорнуте поверх віртуальних машин або контейнерів, розміщеного на сервері MEC. Як і будь-яке інше програмне забезпечення, програми із множини A вимагають деяких мінімальних специфікацій системи, щоб мати можливість ефективно працювати. Для спрощення представляємо мінімальні системні вимоги програми a мінімальною пам'яттю, яку вона потребує для роботи. Однак кожна програма може бути забезпечена деякими обчислювальними ресурсами, що перевищують її мінімальні вимоги пам'яті, щоб максимально збільшити навантаження, яке вона може обробити протягом строку дії. Далі ми припускаємо, що ці програми можуть бути спільними для багатьох UE, але можуть обробляти завдання одного UE за один раз.

Ми розглядаємо набір U користувальницьких пристроїв UE з проханням вивантажити свої функції, що залежать від затримки, щоб оброблятись програмою IoT $a \in A$ відповідного типу, розгорнутою на граничному сервері $m \in M$. У цьому розділі ми враховуємо квазістатичний сценарій де множина U UE залишається незмінною протягом періоду розвантаження, однак вона може змінюватися протягом різних періодів. Ми вважаємо, що кожний користувальницький пристрій UE $u \in U$ має декілька обчислювальних завдань. Кожне завдання можна представити кортежем $\langle t_u, \mu_u, \theta_u \rangle$, де $t_u \in T$ вказує на тип програми IoT, необхідний для обробки завдання UE $u \in U$. μ_u являє собою навантаження (цикли), необхідні для виконання

задачі UE u, і може бути отримане шляхом профілювання завдання виконання. θ_u позначає вимоги до затримки (тобто терміну) у часових інтервалах завдання UE u.

Обробка завантажених завдань з урахуванням їх затримок вимагає прийняття рішення про сервер MEC, на який слід завантажувати кожне із завдань, визначення ресурсів обчислення для розподілу в додатках IoT, які будуть обробляти завдання, окрім визначення порядку в якому вивантажені завдання повинні оброблятися кожним із додатків. Вирішення трьох вищезазначених завдань сильно впливають на прийняття завдань у мережу, оскільки вони безпосередньо впливають на деякі затримки, які вони зазнають. Далі ми підсумовуємо затримки, що виникають у завантаженій задачі. Вони можуть бути наступних типів.

1. Затримка завантаження d_{up}^{u}: затримка завантаження завдання відповідає часу, який є необхідним для передачі завдання з UE *u* службі базової станції. Ми припускаємо, що обслуговуюча базова станція $s \in S$ кожного UE *u* є базовою станцією з найвищою якістю прийнятого сигналу. Для спрощення припускаємо, що d_{up}^{u} визначений заздалегідь і може бути розрахований на основі співвідношення сигнал / перешкода плюс коефіцієнт шуму (SINR).

2. Час затримки з кінця – в кінець (end-to-end) d_{ee}^{u}: після завантаження завдання UE *u* його слід обробити додатком IoT від $a \in A$ відповідного типу, розгорнутий на MEC $m \in M$, на який було завантажено завдання. Найкращою ситуацією вважатимемо, коли завдання було оброблено MEC, приєднаним до його обслуговуючої БС, щоб уникнути будь-яких додаткових затримок у мережі. Однак обслуговуюча БС може бути не включена з можливостями MEC (UE1 на рис. 3.1), або приєднаний до нього MEC-сервер може не мати змоги обробити завдання *u* протягом часу *u*; тобто:

а) сервер MEC *m* не розміщує екземпляр додатку того ж типу, який вимагається *u* $(t_a != t_u)$, або

б) розміщений екземпляр додатку того ж типу, який запитується *u*, не має достатньої ємності для обробки, щоб досягти терміну виконання завдання або

в) *a* перевантажений, і, отже, завданню UE *u* доведеться як очікувати свого виконання впродовж затримки свого буфера перед тим, як його обробляти, так і виконання інших завдань, які були заплановані перед ним, оскільки *a* може обробляти тільки завдання одного UE за один раз. Таким чином, у будь-якій з цих ситуацій завдання UE *u* може бути завантажено на інший MEC-сервер m', який здатний обробити його відповідно до його вимог QoE. У цьому випадку обслуговуючій БС необхідно передати завдання іншій БС $s \in S$, де розміщено m'. Отже, ми позначаємо через затримку d_{ee}^{u}, що виникає для передачі завдання UE *u* від його обслуговуючої БС до БС, підключеної до сервера MEC, де завдання *u* буде оброблено. Оскільки ядро на основі SDN може використовуватися для встановлення маршруту маршрутизації між двома БС [17], то затримка d_{ee}^{u} від краю до краю може бути виміряна контролерами SDN, які включають інструменти моніторингу, такі, наприклад, як SLAM [18]. Таким чином, ми визначаємо матрицю *H* з елементами h_{mu} для відображення значення $d_e e^u$ для кожного UE $u \in U$ для кожного MEC-сервера $m \in M$. Зауважимо, що $d_e e^u = h_u^m = 0$, якщо MEC-сервер *m* приєднаний до обслуговуючої БС *u*.

3. **Очікування затримки** d_{wait}^{u}**:** Коли завдання UE доходить до сервера MEC *m*, що розміщує додаток IoT, який може обробити це завдання, може мати певні затримки очікування, які ми позначаємо d_{wait}^{u}, в буфері *a*. Така затримка залежить від порядку планування та розміру завдань, призначених *a*.

4. З**атримка на обробку** d_{proc}^{u}**:** Після того, як завдання *u* почне обробку на призначеному додатку *a*, воно відчує затримку обробки, яка оцінюється як $d_{proc}^{u} \times d_{proc}^{u}$, тобто час, який витрачається *a* на виконання завдання UE *u* і навпаки:

$$d_{proc}^{u} = \frac{\mu_u}{p_a} \tag{3.1}$$

5. **Затримка на завантаження** d_{down}^{u} .Після того, як виконання завдання UE *u* додатком IoT *a* буде завершено, виконуючий віддалений вузол повинен передавати результат виконання назад до *u*. Оскільки об’єм даних для виведення, як правило, набагато менший, ніж початковий розмір завдання, ми вважаємо, що затримка

завантаження, що виникає при перенесенні результатів обробки на *u*, є мізерною. Таким чином, вважатимемо, що $d^u_{down} = 0$.

Далі вирішимо спільну проблему розвантаження задач, розподілу ресурсів програми та планування завдань.

3.1.3. Модель для динамічного планування та розвантаження завдань

Користуючись вищенаведеною моделлю, формулюємо задачу динамічного розвантаження та планування задач як змішану цілочисельну програму.

Нехай *G (N; E)* – фізична мережа (рис. 3.1), що складається з набору вузлів $N = R \cup M \cup S$, де *R* – це набір фізичного обладнання (наприклад, комутатори, маршрутизатори тощо), а *M* – набір серверів MEC приєднаний до набору *S* БС; *E* – це сукупність зв'язків, що їх з'єднують. Нехай *A* – це набір програм IoT різних типів розгорнутий на серверах MEC $m \in M$, і нехай *U* – це набір UE, що вимагають вивантажити та обробити свої функції, залежні від затримок у цих додатках. Тоді проблему змішаного цілочисельного програмування (ЗЦП) можна формально визначити наступним чином.

З огляду на фізичну мережу G (N; E), набір U UE, кожен UE з проханням завантажити та обробити генеровану задачу в додатку IoT одного типу, розгорнутому на одному з серверів MEC $m \in M$ визначити оптимальне призначення завдань, породжених UE, набору додатків $a \in A$, забезпечення обчислювальних ресурсів для кожної програми a і графік обробки завдань, призначених для кожного з них, щоб максимально збільшити кількість допущених завдань у відповідності до їх вимог до затримки.

У таблиці 3.1 окреслені параметри, використані при формулюванні задачі ЗЦП, представленої нижче.

Таблиця 3.1 – Вхідна інформація при формулювання задачі ЗЦП

Мережева інформація	
G (N; E)	Фізична мережа із N вузлів та відповідно з'єднувальних ліній
S	Набір БС
R	Набір фізичного обладнання
M	Набір серверів MEC
A	Набір додатків, які мають бути розгорнуті на $m \in M$
T	Набір типів додатків
P	Набір обчислювальних потужностей, які можна призначити додатку $a \in A$
$c_m \in N^+$	Обчислювальна потужність серверу MEC $m \in M$
$x_m^a \in \{0,1\}$	Ситуація, коли MEC сервер виконує завдання додатку (1), або ні (0)
$t_a \in N^+$	Конкретний тип виконуваного додатку
$p_{\min}^a \in N^+$	Мінімальна затребування обчислювальна потужність для виконання додатку $a \in A$
Інформація щодо обладнання	
U	Набір користувальницьких пристроїв UE
$t_u \in N^+$	Тип запитуваного додатку для виконання завдань UE
$DL_u \in N^+$	Максимальний час виконання задачі, яка розвантажена UE
$m_u \in N^+$	Кількість циклів, необхідних для виконань завдань UE
$m_{ир}^u \in N^+$	Затримка на завантаження завдання
$h_u^m \in N^+$	Затримка на передачу даних end-to-end від UE до MEC сервера
Інша інформація	
Δ	Кількість досліджуваних часових інтервалів

Вводимо змінну $y_u^{a\delta}$, щоб визначити, що додаток IoT $a \in A$ розпочав обробку завдання UE $u \in U$ на часовому слоті $\delta \in DL$.

$$y_u^{a\delta} = \begin{cases} 1, & \text{якщо завдання UE } u \in U \text{ розпочало його обробку,} \\ & \text{використовуючи додаток } a \in A \text{ на часовому проміжку } \delta \in DL; \\ 0, & \text{у протилежному випадку.} \end{cases}$$

Таким чином, наша мета – максимізувати кількість допущених завдань (рівняння (3.2)). Завдання UE $u \in U$ приймається, якщо воно може бути оброблене програмою $a \in A$ протягом визначеного терміну *u*.

$$Maximize \sum_{u \in U} \sum_{a \in A} \sum_{\delta \in \Delta} y_u^{a\delta} \tag{3.2}$$

Для досягнення нашої мети необхідно дотримуватися декількох обмежень, які полягають в наступному.

Визначаємо $p_a \in R^+$ як змінну, яка визначає ємність обчислювальних потужностей, виділених для програми $a \in A$.

Введемо змінну $n_a \in \{0,1\}$, щоб зобразити, що використовується програма $a \in A$, тобто вона обробляє принаймні одне завдання (1), чи ні (0).

$$n_a = \begin{cases} 1, & \text{якщо додаток } a \in A \text{ виконується;} \\ 0, & \text{у протилежному випадку.} \end{cases}$$

Далі введемо $s_{uu'}^a \in \{0,1\}$ як змінну, що вказує на те, що завдання UE *u* почало оброблятись в додатку *a* на виконання завдання UE *u'*.

$$s_{uu'}^a = \begin{cases} 1, & \text{якщо додаток } a \in A \text{ виконується до завдання UE } u'; \\ 0, & \text{у протилежному випадку.} \end{cases}$$

Щоб максимально збільшити кількість допущених завдань, нам потрібно спершу визначитися з обчислювальними ресурсами, які потрібно виділити розгорнутим програмам. Отже, ми визначаємо рівняння (3.3) та рівняння (3.4), щоб вказати, що додаток *a* використовується, якщо в ньому планується обробляти принаймні одне завдання, і переконатися, що воно не використовується інакше.

$$n_a \le \sum_{u \in U} \sum_{\delta \in \Delta} y_u^{a\delta} \qquad \forall_a \in A \tag{3.3}$$

$$Hn_a \ge \sum_{u \in U} \sum_{\delta \in \Delta} y_u^{a\delta} \qquad \forall_a \in A \tag{3.4}$$

Рівність (3.5) гарантує, що для виконання програми $a \in A$ щонайменше виділено мінімальні обчислювальні ресурси, потрібні для її виконання:

$$p_a \geq n_a p^a_{\min} \qquad \forall_a \in A \tag{3.5}$$

Рівність (3.6) гарантує, що максимальна обчислювальна потужність, що може бути призначена для програми $a \in A$ не може перевищувати ємність сервера МЕС $m \in M$.

$$p_a \geq n_a \sum_{m \in M} x^a_m c_m \qquad \forall_a \in A \tag{3.6}$$

Зауважимо, що рівність (3.5) та рівність (3.6) гарантують, що додатку $a \in A$ не буде виділено жодних обчислювальних ресурсів, якщо він не використовуються.

При цьому рівність (3.7) гарантує, що ємність сервера МЕС $m \in M$ не перевищується.

$$\sum_{a \in A} x^a_m p_a \leq c_a \tag{3.7}$$

Поточне завантаження завдання говорить про те, що завдання UE u не планується для більш ніж однієї програми $a \in A$ (рівність (3.8)).

$$\sum_{a \in A} \sum_{\delta \in \Delta} y^{a\delta}_u \leq 1 \qquad \forall_u \in U \tag{3.8}$$

Завдання UE $u \in U$ повинно повністю обробитися перед початком нового завдання. Таким чином, додаток *a* не може почати обробляти завдання UE *u'* перед тим, як закінчити обробку завдання UE *u*. Це можливо лише в тому випадку, якщо u заплановано перед *u'* на *a* (вираз (3.11)). Аналогічно, програма *a* не може почати обробляти завдання UE *u*, перш ніж закінчити обробку завдання *u'*. Це можливо лише в тому випадку, якщо *u'* заплановано перед *u* на *a* (вираз (3.12)).

$$\sum_{\delta \in \Delta} y^{a\delta}_{u'} \delta \geq \sum_{\delta \in \Delta} y^{a\delta}_{u'} \delta + d^u_{proc} - H\left(1 - s^a_{uu'}\right) \qquad \begin{matrix} \forall_a \in A : \left(t_u = t_{u'} = t_a\right) \\ \forall_{u,u'} \in U : \left(u ! = u'\right) \end{matrix} \tag{3.11}$$

$$\sum_{\delta \in \Delta} y^{a\delta}_{u} \delta \geq \sum_{\delta \in \square} y^{a\delta}_{u'} \delta + d^{u'}_{proc} - H\left(1 - s^a_{u'u}\right) \qquad \begin{matrix} \forall_a \in A : \left(t_u = t_{u'} = t_a\right) \\ \forall_{u,u'} \in U : \left(u ! = u'\right) \end{matrix} \tag{3.12}$$

При цьому, d^u_{proc} у виразах (3.12) та (3.11) визначають затримку обробки UE *u* з обробки додатків:

$$d^u_{proc} = \sum_{a\in A}\sum_{\delta\in\Delta} y^{a\delta}_u \frac{\mu_u}{p_a} \qquad \forall_u \in U \tag{3.13}$$

Програма *a* не може почати обробляти завдання UE *u*, якщо завдання не завантажено і не передається додатку (умова (3.15)).

$$\sum_{a\in A}\sum_{\delta\in\Delta} y^{a\delta}_u d^u_{up} + d^u_{ee}\delta\sum_{a\in A}\sum_{\delta\in\Delta} y^{a\delta}_u \delta \qquad \forall_u \in U \tag{3.15}$$

де глибина d^u_{ee} фіксує затримку від краю до краю і визначається, як зазначено у рівності (3.16).

$$d^u_{ee} = \sum_{m\in M}\sum_{a\in A}\sum_{\delta\in\Delta} y^{a\delta}_u x^a_m h^m_u \qquad \forall_u \in U \tag{3.16}$$

Нарешті, оскільки ми вирішуємо завдання із суворими строками, нам потрібно забезпечити повну затримку, яку зазнає завдання UE $u\in U$, щоб вона не перевищувала гранично допустимого строку виконання, визначеного в (3.17), де d^u_{proc} є таким, як визначено у рівності (3.13).

$$\sum_{a\in A}\sum_{\delta\in\Delta} y^{a\delta}_u \delta + d^u_{proc} \le \theta_u \qquad \forall_u \in U \tag{3.17}$$

Рівності (3.11), (3.12) та (3.17) є нелінійними через нелінійність d^u_{proc} (рівність (3.13)). Така нелінійність пов'язана з поділом на змінну рішень p_a, яка множиться на іншу змінну $y^{a\delta}_u$. Отже, щоб лінеаризувати його, ми зменшуємо простір пошуку, дозволяючи p_a, взяти максимум одну конкретне значення із заздалегідь заданого набору *P* замість усіх значень R^+. Це визначається рівнянням (3.18).

$$\sum_{p\in P} z^p_a \le 1 \qquad \forall_a \in A \tag{3.18}$$

де z^p_a - нова змінна рішення, яка визначається наступним чином:

$$z^p_a = \begin{cases} 1, & \text{якщо додатку } a\in A \text{ виділено обчилювальну потужність } p\in P; \\ 0, & \text{у протилежному випадку.} \end{cases}$$

Таким чином, рівність (3.13) можна переписати наступним чином:

$$d^{u}_{proc} = \sum_{a\in A}\sum_{\delta\in\Delta} y^{a\delta}_{u} \sum_{p\in P} z^{p}_{a} \frac{\mu_{u}}{p_{a}} \qquad \forall_{u} \in U \quad (3.19)$$

Аналогічно p_a може бути замінено $\sum_{p\in P} z^{p}_{a}$ на обмеження (3.5), (3.6) та (3.7). Нарешті, рівності (3.11), (3.12), (3.14), (3.17) і (3.19) нелінійні і можуть бути легко лінеаризовані, але ми опускаємо деталі лінеаризації через обмеження простору.

3.1.4. Використання декомпозиції Бендерса для динамічного планування та розвантаження завдань

Як видно із попереднього підрозділу, динамічне планування та розвантаження завдань – це складна задача, для вирішення якої потрібні високопродуктивні обчислення. Фактично, дана задача розвантаження може бути розглянута як NP-складна за рахунок скорочення від узагальненої проблеми призначення [19], тоді як проблема розподілу ресурсів програми може бути розглянута як NP-складна шляхом зменшення з двовимірної проблеми пакування, де MEC-сервери – це пакувальники, а програми – об'єкти для пакування. Аналогічно, проблема планування завдань може бути представлена як NP-складна через зменшення проблеми планування робочих місць [20].

Враховуючи складність вирішення вище окресленої задачі, розглядаємо наступну методику розкладання Бендера [21] на основі логіки (LBBD).

LBBD [22] – це техніка генерації рядків, яка дотримується стратегії "недобре навчання". Вона заключається в розкладанні всієї задачі на головну задачу, що представляє собою спрощену вихідну модель та одну або кілька підзадач. Таким чином, буде проводитись оптимізація над вторинними змінними при одночасній фіксації первинних змінних, обчислених на основі рішення основної задачі. Присвоюючи первинним змінним деякі пробні значення та вирішуючи вторинні задачі, LBBD буде визначати якість інших пробних рішень, які потім використовуються для зменшення кількості рішень, які необхідно перерахувати, щоб знайти оптимальне. Точніше, рішення вторинної задачі використовується для

отримання надрізів Бендерса, які додаються до результатів вирішення основної задачі, щоб вирізати нездійсненні розв’язки.

LBBD складається з ітеративного вирішення основної та вторинних задач, обчислення їх результатів та додавання скорочень Бендерса, поки рішення основної та вторинних задач не зблизяться.

Ефективність LBBD спирається на підхід до розкладання та міцність визначених скорочень Бендерса. На відміну від класичного підходу Бендерса, коли зрізи Бендерса можна легко визначити на основі множників Лагранжа, отриманих з рішення вторинної задачі, не існує стандартної схеми для генерації Бендерського розрізу для LBBD [21]. Оскільки класичний Бендерс є недоцільним для проблеми вирішення вище окресленої проблеми, наша пропозиція складається з поділу задачі динамічного планування та розвантаження завдань на основну підзадачу, що вирішує завдання розподілу ресурсів додатків, та декілька вторинних підзадач для вирішення проблеми планування завдань. Спільне вирішення завдань розподілу ресурсів програми позитивно впливає на ефективність нашого рішення:

1) Оцінка верхньої межі кількості завдань, які можна запланувати, оскільки це спрощує вихідну проблему.

2) Можливість скористатися схемою розподіленого планування, розробивши проблему планування для кожного використовуваного додатку. При цьому, розподілене планування дозволяє паралельно виконувати декілька вторинних підзадач, що очевидно скорочує загальний час вирішення загальної задачі.

Як зображено на рис. 3.2, виконання розробленого алгоритму починається з вирішення первинної задачі.

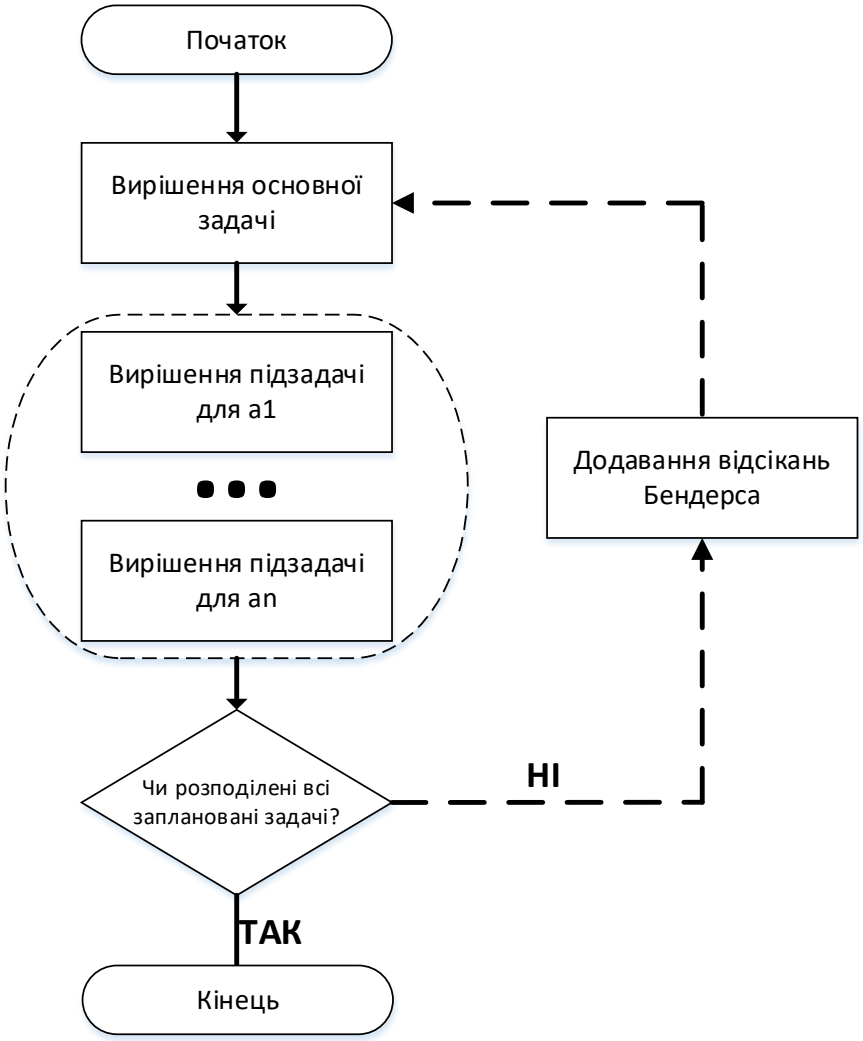


Рис. 3.2. Узагальнений алгоритм вирішення задачі динамічного планування та розвантаження завдань

Її рішення забезпечує призначення підмножини завдань UE для розміщених додатків та ємність обробки, що виділяється для кожного з цих додатків. Для кожної використовуваної програми $a \in A$, вторинна задача визначається і подається набором завдань UE, призначених первинною задачею за допомогою та обробкою ємності p_a, виділеної на *a*. Нехай $\psi_{ОЗа}$ - загальна кількість завдань UE, призначених основній задачі. Крім того, нехай вторинні задачі *a* позначають максимальну кількість завдань UE, які можуть бути заплановані у відповідності зі своїми вимогами до затримки в $\psi_{ВЗ}$. Для кожної використовуваної програми $a \in A$, якщо $(\psi_{ОЗа} > \psi_{ВЗа})$, отриманий відрізок Бендерса і доданий до основної задачі, щоб направити його на

визначення кращого значення для p_a, а значить, і виконання завдання, яке, ймовірно, буде заплановано вторинній задачі. Проблема з основною задачею вирішується знову після додавання зрізів Бендерса, що випливають із всіх вторинних задач. Цей процес повторюється до тих пір, поки $\sum_{a\in A}\psi_{O3a} = \sum_{a\in A}\psi_{B3a}$, коли оптимальне рішення буде досягнуто.

Опис основної задачі. Вхідні параметри, визначені для основної задачі, детально описані в таблиці 3.2.

Таблиця 3.2 – Вхідні параметри до вирішення основної задачі

P_u^a	Набір процесорних потужностей, виділених додаткам $a \in A$, що надають змогу виконати завдання до гранично допустимого строку
$\sigma_u^a \in N^+$	Час прибуття задачі від UE *u* до додатку $a \in A$
$\sigma_{\min}^a \in N^+$	Мінімальний час прибуття для всіх задач від UE *u* до додатку $a \in A$
$DL_{\max}^a \in N^+$	Максимально допустимий строк прибуття всіх задач, які можуть бути виконані додатком $a \in A$

Визначимо змінну $q_u^a \in \{0,1\}$, щоб визначити, що завдання UE $u \in U$ відображається на додаток $a \in A$.

$$q_u^a = \begin{cases} 1, & \text{якщо завдання UE відображено на додаток } a \in A; \\ 0, & \text{у протилежному випадку.} \end{cases}$$

Мета вирішення основної задачі – максимальне збільшення кількості допущених завдань (рівність (3.20)).

$$\psi MP = Maximize \sum_{u\in U}\sum_{a\in A} q_u^a \tag{3.20}$$

Основна задача містить кілька обмежень. Почнемо з визначення змінної $n_a \in \{0,1\}$, щоб вказати, що використовується програма $a \in A$, тобто якщо їй призначено завдання принаймні одного UE.

$$n_a = \begin{cases} 1, & \text{якщо використовується додаток } a \in A; \\ 0, & \text{у протилежному випадку.} \end{cases}$$

Ми визначаємо $p_a \in R^+$ як змінну, яка визначає ємність обробки, що виділена програмі, $a \in A$.

Таким чином, основні обмеження, яких потрібно дотримуватись під час вирішення основної задачі, наступні.

Набір обмежень включає рівність (3.21), де показано, що завдання UE $u \in U$ може бути оброблено щонайменше одним додатком $a \in A$.

$$\sum_{a \in A} q_u^a \leq 1 \qquad \forall_u \in U \tag{3.21}$$

Крім того, формула (3.22) сформульована для запобігання обробці завдань UE для додатків різного типу.

$$\sum_{u \in U} \sum_{a \in A: t_a != t_u} q_u^a = 0 \tag{3.22}$$

У рівностях (3.23) та (3.24) вказується, що програма $a \in A$ використовується, якщо щонайменше одне завдання завантажено для обробки на ній.

$$n_a \leq \sum_{u \in U} q_u^a \qquad \forall_a \in A \tag{3.23}$$

$$Hn_a \sum_{u \in U} q_u^a \qquad \forall_a \in A \tag{3.24}$$

Рівність (3.25) гарантує, що якщо використовуються додатки $a \in A$, то повинна бути виокремлена принаймні мінімальна обчислювальна потужність p^a_{min}.

$$p_a \geq n_a p_{min}^a \qquad \forall_a \in A \tag{3.25}$$

$$p_a \leq n_a \sum_{m \in M} x_m^a c_m \qquad \forall_a \in A \tag{3.26}$$

Рівності (3.25) та (3.26) гарантують, що жоден обчислювальний ресурс не може бути призначений невикористаному додатку $a \in A$. Ми вводимо рівність (3.27) для того, щоб забезпечити дотримання потужності кожного сервера MEC $m \in M$, які можна виділити для програми $a \in A$.

$$\sum_{a \in A} x_m^a p_a \le c_m \quad \forall_a \in A \tag{3.27}$$

Хоча вищеназвана формула для опису основної задачі забезпечує верхню межу оптимального рішення задачі, наші ресурси, необхідні для обробки всіх покладених завдань на додаток *a*, можна визначити, виходячи з рівняння (3.1), враховуючи максимальний час обробки, доступний їм, і який можна обчислити, відрахувавши мігімальний час прибуття до *a* (тобто, $min\sigma_u^a = min\forall_u \in U : (t_u = t_a)$ де σ_u^a відповідає затримці завантаження і граничним обчисленням UE до *a* від максимальної граничної тривалості всіх завдань UE, які вимагають одного типу *a*.

$$p_a \ge \frac{\sum_{u \in U:(t_u = t_a)} \mu_u q_u^a}{\theta_{max}^a - \theta_{min}^a} \quad \forall_a \in A \tag{3.28}$$

Визначаємо обчислювальні ресурси, встановлені P_u^a за допомогою схеми попередньої обробки, попередньо застосувавши рівняння (3.29) для обчислення мінімальних ресурсів обробки, необхідних для *a* для виконання терміну виконання завдання UE *u*. Потім можна визначити значення ємності $p \in P$, що перевищує p_{min}^{ua} і при цьому не перевищує ємність MEC-сервера $a\left(P_u^a = \left\{ p \in P : p_{min}^{ua} \le p \le \sum_{m \in M} x_m^a c_m \right\}\right)$:

$$p_{min}^{ua} = \frac{\mu_u}{\theta_u - \sigma_u^a} \quad \begin{matrix} \forall_u \in U \\ \forall_a \in A : (t_u = t_a) \end{matrix} \tag{3.29}$$

Змінна, яка може бути використання для прийняття рішення про те, яка процесорна потужність виділена на обробку завдання UE $u \in U$ додатком $a \in A$.

$$\beta_{ua}^j = \begin{cases} 1, & \text{якщо відбувається обробка завдання UE } u \in U \text{ додатком } a \in A; \\ 0, & \text{у протилежному випадку.} \end{cases}$$

Додамо нерівність, зображену у рівнянні (3.30), як обмеження в модель для опису основної задачі, щоб вказати, що p_a має бути більшим або рівним максимальним ресурсам обробки, необхідним для обробки будь-якої задачі, призначеної для UE $u \in U$.

$$p_a \geq \sum_{j \in P_u^a} j \beta_{ua}^j \quad \begin{matrix} \forall_u \in U \\ \forall_a \in A \end{matrix} \tag{3.30}$$

Рівність (3.31) вводиться для гарантування того, що для завдання UE $u \in U$ для програми $a \in A$ вибрано одну ємність обробки, якщо і лише тоді, вона відображена на a.

$$\sum_{j \in P_u^a} \beta_{ua}^j = q_u^a \quad \begin{matrix} \forall_u \in U \\ \forall_a \in A \end{matrix} \tag{3.31}$$

Таким чином, спочатку додаємо рівність (3.28) для визначення нижньої межі ресурсів обробки, які повинні бути розподілені для кожної програми по a на основі поставлених завдань.

Щоб гарантувати, що одна програма обробляє додаток a вводимо наступну рівність:

$$\sum_{u \in U : \left(P_u^a = \varnothing\right)} \sum_{a \in A : t_u = t_a} q_u^a = 0 \tag{3.32}$$

Опис вторинних задач. У таблиці 3.3 представлені вихідні дані для даної моделі.

Таблиця 3.3 – Вхідні параметри до вторинних задач

p_a	Необхідна процесорна потужність для додатку $a \in A$
U_a	Підмножина UE, які були відведені до виконання додатком a
$\sigma_u^a \in N^+$	Час прибуття для всіх задач від UE u до додатку $a \in A$

Вволимо змінну $y_u \in N^+$ для визначення рівня часового проміжку, під час якого розпочинається обробка задачі UE $u \in U_a$ додатку a. Далі представляємо $s_{uu'} \in \{0,1\}$ як змінну, щоб вказати, чи розпочато обробку завдання UE $u \in U_a$ в додатку a перед завданням UE $u' \in U_a$.

$$s_{uu'} = \begin{cases} 1, & \text{якщо розпочато обробку завдання UE } u \in U_a \text{ в додатку } a \text{ перед} \\ & \text{завданням UE } u' \in U_a; \\ 0, & \text{у протилежному випадку.} \end{cases}$$

$$\psi SP_a = Maximiz \sum_{u \in U_a} \alpha_u \tag{3.33}$$

Для даної підзадачі також вводимо декілька обмежень. Рівність (3.34) гарантує, що завдання UE $u \in U_a$ не можуть почати обробку до моменту прибуття задачі:

$$y_u \geq \sigma_u^a \alpha_u \quad \forall_u \in U_a \tag{3.34}$$

Крім того, заявка *a* повинна гарантувати послідовну обробка завдання UE $u \in U_a$ протягом усіх його необхідних значень часу обробки. Таким чином, рівності (3.35) та (3.36) гарантують, що жлдні дві задачі не можуть бути заплановані одночасно.

$$y_u \geq y_{u'} + d_{proc}^{u'} \alpha_{u'} - H\left(1 - s_{uu'}\right) \quad \forall_{u,u'} \in U_a : \left(u != u'\right) \tag{3.35}$$

$$y_{u'} \geq y_u + d_{proc}^{u} \alpha_u - H\left(1 - s_{uu'}\right) \quad \forall_{u,u'} \in U_a : \left(u != u'\right) \tag{3.36}$$

де d_{proc}^{u} і $d_{proc}^{u'}$ надають змогу визначити затримку обробки *u* і *u*’ (рівність (3.1)). Рівність (3.37) являє собою обмеження пріоритету графіку завдань UE *u* і *u’* на *a*.

$$s_{uu'} + s_{u'u} = \alpha_u \alpha_{u'} \quad \forall_{u,u'} \in U_a : \left(u != u'\right) \tag{3.37}$$

Нарешті, рівняння (3.38) гарантує, що загальна затримка, яку зазнає завдання UE $u \in U_a$ не перевищує максимально допустимої затримки.

$$y_u + d_{proc}^{u} \alpha_u \leq \theta_u \quad \forall_u \in U_a \tag{3.38}$$

Відсікання Бендерса. Якщо вторинна підзадача не в змозі запланувати всі завдання призначені UE щоб застосувати за допомогою основної задачі, тоді повинно відбутися відсічення Бендерса і має бути додане до рішення основної задачі для відкидання нездійсненних рішень.

Для того, щоб визначити відсікання Бендерса, будемо намагатись визначити рішення, які, ймовірно, будуть отримані в результаті вирішення основної задачі, але будуть нездійсненним для вторинної підзадачі. Таким чином, можна графічно зобразити додаток *a* як бункер висоти $h = p_a$ і ширини $\omega = \theta_{max}^a \sigma_{min}^a$, що вказує на часовий горизонт, протягом якого завдання UEs $u \in U_a$. Ua мають бути заплановані та оброблені на *a* (табл. 3.4).

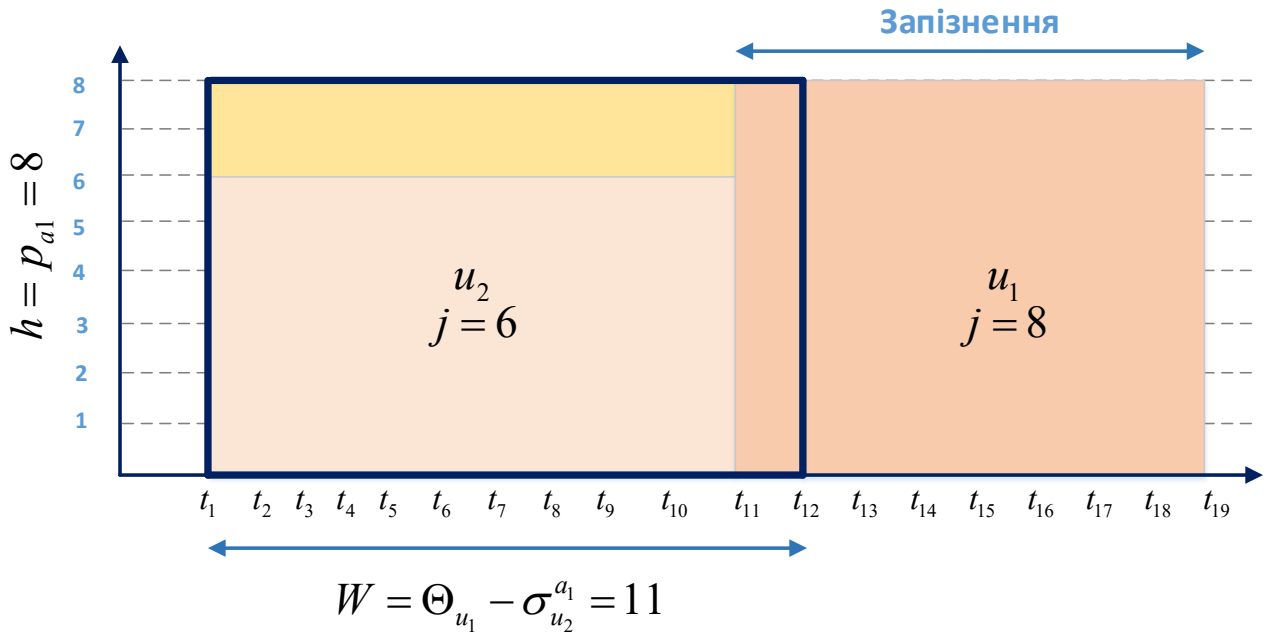
Таблиця 3.4 – Вхідні дані до вирішення вторинної підзадачі (додаток а, p_{a_1})

Користувальніцькі пристрої UE	Час прибуття завдання	Цикли	Граничний час виконання	P_u^a
u_1	t_4	61	t_{12}	{6,7,8,9,10}
u_2	t_1	60	t_{11}	{6,7,8,9,10}

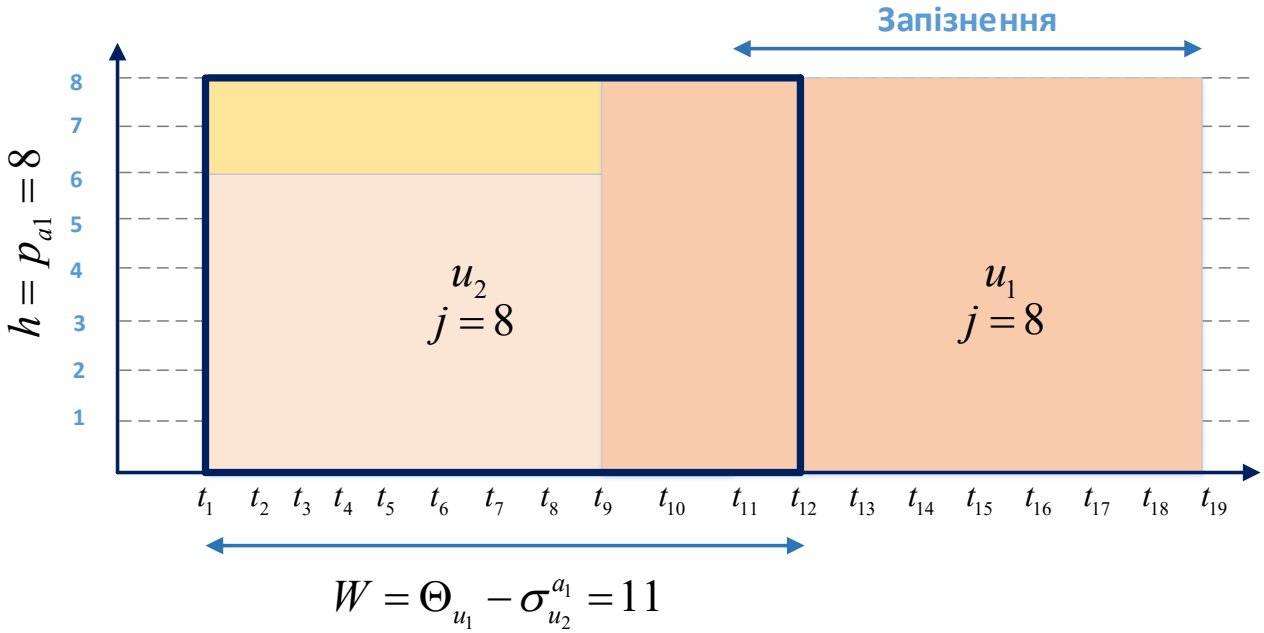
Кожне завдання $u \in U_a$ можна розглядати як прямокутник висоти $h_u = j$, де j – призначена йому ємність обробки основної задачі $(j \in P_u^a : \beta_u a^j = 1)$ і шириною $\omega_u = d_{proc}^u$, що представляє час обробки u на a при призначенні ємності обробки j. Проблема з плануванням може бути абстрагована від проблеми упаковки у бін, де A – це бін, а завдання a – це об'єкти, які потрібно розмістити. Геометричний розмір (тобто більша ємність обробки) задачі u може бути збільшена за допомогою збільшення її висоти, якщо збільшуємо j (тобто продовжити час його завершення (3.1)).

Розглянемо приклад, показаний на рис. 3.3, де ми беремо дві задачі u_1 та u_2, заплановані на додаток a_1 (табл. 3.4). a_1 може бути представлений у вигляді бункера, а u_1 і u_2 – об'єкти, що розміщуються в ньому рис. 3.3.

На рис. 3.3 (а) призначаємо завданню u_2 обчислювальні ресурси j = 6 циклів/ часовий проміжок, який дає затримку на обробку в 10 часових інтервалів, щоб закінчити на t_{11}. Збільшення обчислювальних ресурсів до j = 8 циклів /часовий проміжок для опрацювання завдання u_2 зменшує час його обробки до 8 часових інтервалів, щоб закінчитись при t_9 (рис. 3.3 (б)). Однак в обох випадках не вдалося визнати u_1 на a_1 та точний термін виконання.



(а)



(б)

Рис. 3.3. Планування виконання задач (а – *j*=6; б – j=8)

Отже, з ємністю обробки p_{a_1} = 8 циклів / часовий проміжок, призначений для застосування a_1, максимум одна з обох задач u_1 або u_2 може бути допущена до a_1 при призначенні будь-яких обчислювальних ресурсів $j \le p_{a_1}$. Точніше, варіювання ємності обробки *j*, присвоєної u_1 або u_2, призведе до того ж нездійсненного рішення. Таким чином, робимо висновок, що якщо набір задач $u \in U_a$ не зміг би бути запланований вторинній задачі з $p_a \in P$, то точно вони не будуть допущені з будь-

яким іншим значенням $(p'_a < p_a) \in P$. Отже, ми визначаємо скорочення (3.39), щоб обрізати такі нездійсненні рішення, де p_a – обчислювальна потужність, призначена для застосування *a* по основній задачі на попередній ітерації, $\hat{U}_a$ - це набір завдань UE, які були допущені вторинній задачі на *a*, а $\hat{U}_a$ – це набір кінцевих користувачів UE, завдання яких було відхилено вторинною задачею. Зауважимо, що рівність (3.39) скеровує основну задачу в призначенні додатку *a* щонайменше однакової кількості завдань $\left|\hat{U}_a\right|$, які були допущені вторинною задачею з набору завдань $\left(\hat{U}_a \cup u'_{\in \hat{U}_a}\right)$.

$$\underbrace{\sum_{u \in \hat{U}_a} \sum_{j \in P_u^a : (j \le p_a)} \beta_{ua}^j + \sum_{j \in P_{u'}^a : (j \le p_a)} \beta_{u'a}^j}_{\text{Admitted tasks by SP}} \quad \underbrace{}_{\text{One rejected task by SP}} \quad \begin{matrix} \forall_{u'} \in \hat{U} \\ \forall_a \in A \end{matrix} \qquad (3.39)$$

$$\le \underbrace{\left|\hat{U}_a\right|}_{\text{number of admitted tasks by SP}}$$

Такі вказівки застосовуються лише у випадку, коли завданню з $\left(\hat{U}_a \cup u'_{\in \hat{U}_a}\right)$ були призначені обчислювальні ресурси $j \le p_{a_1}$. Відсікання (3.39) додається для кожного завдання, відхиленого в процесі рішення вторинної задачі.

Нижче представлено відсікання, яке необхідно додати як додаткову умову.

$$\beta_{u_1 a_1}^6 + \beta_{u_1 a_1}^7 + \beta_{u_1 a_1}^8 + \beta_{u_2 a_1}^6 + \beta_{u_2 a_1}^7 + \beta_{u_2 a_1}^8 \le 1$$

Щоб гарантувати, що алгоритм сходиться до оптимального рішення, потрібно довести, що рівняння (3.39) є дійсним відсіканням Бендерса [23]. Розріз Бендерса є дійсним, якщо він відповідає наступним двом умовам [23].

3.2. Метод керування випромінюваною потужністю мобільних пристроїв під час розвантаження завдань

В даному підрозділі досліджується управління потужністю випромінюваною потужністю радіопередавачів у стільниковій багатокористувацькій системі в умовах впливу інтерференції. В даному випадку, відмінність даної ситеми від вільної від

завад у тому, що контроль живлення у багатокористувацькій системі в умовах впливу інтерференції головним чином розподілений через взаємодію між різними користувачами мобільних пристроїв, що дуже ускладнює задачу керування потужністю.

Даний підрозділ побудований наступним чином: спочатку було представлено системну модель та проаналізовані проблеми, які вирішувались в даному підрозділі; потім відбувається доведення існування та унікальності рівноваги Неша; потмі представляється, а також досліджується конвергенція (збіжність) та складність обчислення розробленого алгоритму; потмі відбувається оцінка властивостей цього алгоритму через теоретичний аналіз та чисельне моделювання.

3.2.1. Модель системи

Існує N користувачів мобільних пристроїв у багатокористувацькій системі із можливістю проведення граничних обчислень, позначимо їх як $N = \{1,2,...,N\}$. Кожен користувач має обчислювальну задачу, яка може бути розрахована локально або передана на граничний сервер через базову станцію (БС) з маленькими секторами, близькими до користувача. *OFDMA (Orthogonal Frequency Division Multiple Access)*/(Множинний доступ до ортогонального відділення частоти) використовується для перенесення задач з користувачів мобільних пристроїв на граничний сервер сервер [24]. У такій багатокористувацькій системі, є L базових станцій з маленькими секціями, позначимо їх як $BS = \{BS_1, BS_2, ..., BS_l\}$,де $BS_l = l$ та $1 \le l \le L$. Кожна дрібна секція обслуговується БС. n-й користувач мобільного пристрою, який обслуговується BS_l позначається як n_{BS_l}, а BS_l, який обслуговує n-того користувача позначається як BS_l^n. Кожен користувач може підключитися тільки до однієї базової станції для довгострокового вимірювання якості каналу [25]. У цій роботі використовується універсальне частотне повторення [24]. Наявний спектр розділяється на K підканалів, індексація цих підканалів позначається як $sc = \{sc_1, sc_2, ..., sc_k\}$,де $sc_k = k$ та $1 \le k \le K$. Таким чином, користувач, який використовує

канал sc_k для перенесення задачі на граничний сервер, позначається як n^{sc_k}. У системі, заснованій на *OFDMA*, оскільки внутрішньо-секційних багатокористувацький доступ є ортогональним, а між-секційні користувачі повторно використовують спектр [24, 25], то це призводить до сильних між-секційних втручань, що обмежується ємністю системи. У моделі, яка використовується у даному підрозділі, оскільки базові станції мають більше глобальної інформації ніж користувачі мобільних пристроїв, це призводить до того, що базова станція використовує частотний діапазон та самостійно розподіляє користувачів на різні підканали обраного діапазону [25].

Схема задачі розвантаження є бінарною. Рішення про розвантаження користувача *n* позначається як: $a_n = \{0, sc_1, sc_2, .., sc_k\}$; рішення про розвантаження означає, що якщо користувач *n* вивантажує задачу на граничний сервер через канал sc_k, тоді $a_n = sc_k$; якщо ні, то $a_n = 0$. Оскільки головним призначенням даного підрозділу є дослідження контролю живлення у представленій системі, то ми припустимо, що рішення про розвантаження користувача мобільного пристрою приймається заздалегідь, що $a = \{a_1, a_2, ...a_N\}$. Занотуємо, що не всі рішення про розвантаження більше за 0; це означає що деякі користувачі у мережі вирішують свої задачі на граничному сервері, у той час як інші обчислюють свої задачі локально. У даному підрозділі, ми припустимо, що користувач *n* відправляє задачу на граничний сервер, тобто $a = sc_k$.

Позначення, які будуть використовуватися в цьому підрозділі, наведені в таблиці нижче (табл. 3.5).

Таблиця 3.5 – Позначення, які використовуються у даному підрозділі

Позначення	**Його значення**
N	Кількість користувачів мобільних пристроїв у мережі
n	Один з користувачів мобільних пристроїв
L	Кількість базових станцій у мережі
BS_l	Одна з базових станцій
K	Кількість під-каналів
sc_k	Один з під-каналів
n_{BS_l}	Користувач, що обслуговується БС $\boldsymbol{BS_l}$
BS_l^n	БС, яка обслуговую користувача n
n^{sc_k}	Користувач n, який використовує канал $\boldsymbol{sc_k}$
a_n	Рішення про розвантаження користувача n
$r_n(p)$	Швидкість передачі даних користувача n під час вивантаження задачі
w_{sc_k}	Пропускна здатність каналу $\boldsymbol{sc_k}$
l_n	Засада(interference) користувача n
p_n	Потужність передачі користувача n
T_n	Обчислювальна задача користувача n
s_n	Розмір $\boldsymbol{T_n}$
c_n	Цикл процесора, необхідний для обробки $\boldsymbol{T_n}$
t_n^{off}	Час, що потрібен для передачі даних від користувача до МПО сервера
$t_{c,n}$	Час, що потрібен для вирішення задачі на МПО сервері
f_c	Можливості процесора на МПО сервері
e_n^{off}	Енергетичні затрати, що витрачаються на передачу даних від користувача до МПО сервера

f_n	Можливості процесора користувача
$U_{c,n}$	Накладні витрати користувача *n*, що потрібні для перенесення обчислювальної задачі на МПО сервер
U_n	Накладні витрати користувача *n*, що потрібні для розв'язання задачі локально
φ_n	Набір користувачів, які створюють перешкоди для користувача мобільного пристрою *n*

3.2.2. Модель комунікаційного каналу

Потужність передачі користувача *n* рівна p_n. Вона може встановлюватися на рівні від мінімуму до максимуму. У розглянутій системі з наявністю завад, мінімальна потужність передачі повинна зробити *SINR (Signal to Interference plus Noise Ratio)*/(Відношення сигналу до шуму та інтерференції) більше, ніж поріг [18] (поріг *SINR* стосується апаратної архітектури користувача), позначається як p_t. Тому, для надання профайлу з рішенням про розвантаження *a*, швидкість передачі користувача *n* (що враховує інтерференцію) для вивантаження обчислювальної задачі на граничний сервер розраховується як:

$$r_n(\mathrm{p}) = \mathrm{w}_{sc_k} \log_2 \left(1 + \frac{p_n G_n}{\left(\eta_0 + \sum_{i \in N\{n\}, a_i = a_n} p_i G_i \right)} \right) \tag{3.40}$$

де w_{sc_k} – це пропускна здатність каналу sc_k, G_i – посилений канал між користувачем мобільного пристрою *i* та БС BS_l (G_i пов'язана з середовищем та дистанцією між двома точками), η_0 – білий Гаусівський шум та $p = \{p_1, p_2, ..., p_N\}$. У (3.40), припустимо, що $I_n = \sum_{i \in N\{n \ \}, a_i = a_n} p_i G_i$ – сума інтерференцій від інших користувачів мобільних пристроїв, які використовують спільний канал з користувача *n*. Як показано у (3.40), через інтерференцію, потужність передавачів може впливати не тільки на свою швидкість передачі, але також і інших

користувачів, які використовують спільний канал з користувачем *n*. Це робить контроль живлення у таких багатокористувацьких системах повністю розподіленим. Тому, було вирішено обрати елементи теорії ігор для забезпечення контролю живлення у багатокористувацькій системі із граничними обчисленнями в умовах впливу завад.

3.2.3. Модель забезпечення обробки даних граничними серверами

Якщо припустимо, що користувач має обчислювальну задачу, яку необхідно завантажити на граничний сервер, позначимо її як $T_n = \{s_n, c_n\}$, де s_n це розмір вхідних даних завдання, а c_n – цикли процесора, необхідні для обробки вхідних даних. Базуючись на профайлі з рішенням про розвантаження *a*, затримка обчислень та енергетичні витрати можуть бути розраховані на основі математичного апарату приведеного в [26].

Коли $a_n = sc_k$, затримка обчислення стосується двох аспектів:

1) користувач мобільного пристрою передає вхідні дані завдання на граничний сервер;

2) граничний сервер розраховує обчислювальну задачу. Затримка, що відбувається через передачу даних, може бути розрахована як:

$$t_n^{off} = \frac{s_n}{r_n(\text{р})} \tag{3.41}$$

Час, що потрібен для виконання задачі на МПО сервері, розраховується як:

$$t_{c,n} = \frac{c_n}{f_c} \tag{3.42}$$

де f_c це можливості процесора на граничному сервері.

В даній роботі ігноруємо час для передачі результатів обчислення з граничного сервера до користувача мобільного пристрою. Тому, енергетичні витрати, що виникають через передачу даних від користувача мобільного пристрою до граничного сервера, розраховуються наступним чином:

$$e_n^{off} = p_n t_n^{off} = \frac{p_n s_n}{r_n(\text{p})} \tag{3.43}$$

Енергія, що потрібна для обчислення завдання на сервері, може бути обчислена наступним чином:

$$e_{c,n} = k_c c_n f_c^2 \tag{3.44}$$

де k_c – константа, що стосується апаратної архітектури граничного серверу.

Коли $a_n = 0$, обчислювальна задача буле розрахована локально. Тоді затримка на виконання завдання рівна:

$$t_n = \frac{c_n}{f_n} \tag{3.45}$$

де f_n – константа, яка стосується апаратної архітектури користувача мобільного пристрою.

Подібно до понаднормового навантаження, визначеного у [26] та [27], у даній роботі, понаднормове навантаження користувача мобільного пристрою *n* для завантаження обчислювальної задачі на сервер та виконання локально може бути розраховано за наступними формулами:

$$U_{c,n} = a_t \left(t_n^{off} + t_{c,n} \right) + a_в \left(e_n^{off} + e_{c,n} \right) \tag{3.46}$$

$$U_n = a_t t_n + a_в e_n \tag{3.47}$$

де a_t та $a_в$ – вагові коефіцієнти енергетичних витрат та затримки та $a_t, a_в \in [0,1]$. У (3.46) та (3.47), енергетичні витрати та затримка враховуються. Як показано у [26], значення a_t та $a_в$ можуть бути визначені на основі багатозначного корисного підходу теорії прийняття рушення з декількома критеріями [28].

У розглянутій системі, коли $U_{c,n} > U_n$, користувачі мобільних пристроїв обчислюють задачі локально; в іншому випадку, якщо $U_{c,n} > U_n$ обчислювальна задача завантажується на граничний сервер. У цьому підрозділі, оскільки ми припускаємо, що $a_n = sc_k$, то користувач мобільного пристрою *n* завантажить задачу на граничний сервер. Зауважимо, що ми просто припустили, що $a_n = sc_k$, рішення про розвантаження користувача мобільного пристрою може бути рівне або більше за

0. У даній роботі, оскільки ми припускаємо, що рішення про розвантаження вузлів відомі, тому з цього слідує, що ми лише розмовляємо про ефективність контролю живлення для представленої багатокористувацької системи.

Існування та унікальність рівноваги Неша. Відповідно до вищеокресленої моделі, головна проблема щодо надання профайлу з рішенням про розвантаження *a*, яка буде вирішена у цьому підрозділі, полягає у тому, як визначити потужність передачі для кожного користувача мобільного пристрою, щоб мережа понаднормових витрат, представлена у (3.48), отримала мінімальне значення.

$$\min_{p} \sum\nolimits_{n\in N} U_{c,n} \quad , \tag{3.48}$$

де $p_n \in = [p_t, p_{\max}], n \in N$

Гра пошуку оптимального рішення продемонстрована у (3.48), може бути визначена як: $G = \left\{ N, \{p_n\}_{n\in N, a_i = a_n}, U_{c,n}\{p_n\}_{n\in N, a_i = a_n} \right\}$, де p_n – стратегія живлення для користувачі мобільних пристроїв $n, U_{c,n}\{p_n\}$ – понаднормові витрати користувача мобільного пристрою n. Ми спочатку визначили рівновагу Неша гри G наступним чином.

Якщо стратегічний профайл $p_n = \{p_1, p_2, ..., p_N\}$ – це рівновага Неша гри G, тоді жоден з користувачів мобільних пристроїв не може додатково зменшити їх понаднормові витрати, односторонньо корегуючи свої стратегії, тобто $U_{c,n}(p_n, p_{-n}) \le U_{c,n} = (p`_n, p_{-n}), \forall p_n, p`_n \in \forall n \in N$.

У даному твердженні, $p_{-n} = (p_{1,} p_2, ..., p_{n-1}, p_{n+1}, ..., p_N)$ – це набір потужностей передачі усіх інших користувачів, крім користувача *n.* Як відомо, у *OFDMA* системі, тільки користувачі мобільних пристроїв, які знаходяться у різних маленьких стільниках та використовують спільний канал зв'язку з користувачем *n*, можуть впливати на задачу розвантаження користувача *n*, тому таким чином може визначатися набір користувачів, що створюють перешкоди.

Набір користувачів, що створюють перешкоди, для користувача мобільного пристрою *n* визначаємо як набір користувачів мобільних пристроїв, які

використовують спільний канал зв'язку з користувачем мобільного пристрою *n*, позначимо як φ_n.

Наприклад, якщо користувач мобільного пристрою $i \in \varphi_n$, тоді $BS_l^i \neq BS_l^n$ та $a_1 = a_n$.

Тоді для внутрішніх $[p_t, p_{\max}]$, рівновага Неша для контролю живлення гри *G* існує [29]. Доведення цього твердження наведено нижче.

Відповідно до Теореми Гліксберга [30], якщо підходящій регіон p_n це компактна опукла множина та $U_{c,n} = (p_n, p_{-n})$ безперервна на p_n, тоді рівновага Неша гри *G* існує.

По-перше, ми доводимо, що внутрішніх $[p_t, p_{\max}]$ є компактною опуклою множиною. Оскільки $[p_t, p_{\max}]$ це обмежена замкнута множина, отже це компактна множина. Відповідно до визначеної опуклої множини, якщо внутрішні $[p_t, p_{\max}]$, тому для $\forall \lambda \in [0,1]$ та $\forall p_{1,} p_2 \in [p_t, p_{\max}], \lambda p_1 + (1-\lambda) p_2 \in [p_t, p_{\max}]$ утримання. Якщо внутрішні $[p_t, p_{\max}]$ не є опуклою множиною, вони повинні існувати $\lambda \in [0,1]$, яка робить $\lambda p_1 + (1-\lambda) > p_{\max}$ або $\lambda p_1 + (1-\lambda) p_2 < p_1$ утримуються. Якщо $\lambda p_1 + (1-\lambda) p_2 > p_{\max}$, це дорівнює $\lambda p_1 + (1-\lambda) > p_{\max} - p_2$. Отже якщо $p_1 > p_2$, то $\lambda > \frac{p_{\max} - p_2}{p_1 - p_2} > 1$; якщо $p_1 < p_2$, то $\lambda < \frac{p_{\max} - p_2}{p_1 - p_2} \leftarrow 1$; це неможливо оскільки $\lambda \in [0,1]$. Коли $\lambda p_1 + (1-\lambda) p_2 < p_t$, це дорівнює $\lambda(p_1 + p_2) < p_1 - p_2$; отже якщо $p_1 > p_2$, то $\lambda < \frac{p_t - p_2}{p_1 - p_2} < 0$; якщо $p_1 < p_2$, то $\lambda > \frac{p_t - p_2}{p_1 - p_2} > 1$. Отже внутрішні $[p_t, p_{\max}]$ це компактна опукла множина.

По-другу, ми доведемо що $U_{c,n} = (p_n, p_{-n})$ безперервна на p_n. Розглядаючи дві множини $p_n = \{p_1, p_2, ..., p_N\}$ та $p_\varepsilon = \{p_1 + \varepsilon_1, p_2 + \varepsilon_2, ..., p_N + \varepsilon_N\}$, де $\varepsilon \in \left[0, \min\{p_{\max} - p_j, j \in N\}\right]$. Відповідно до (3.46), $U_{c,n} = (p_n, p_{-n})$ та $U_{c,n}(p_n + \varepsilon_n p_{-n} + \varepsilon_{-n})$ можуть бути розраховані як:

$$U_{c,n}(p_n, p_{-n}) = a\left(\frac{s_n}{r_n(p_n, p_{-n})} + \frac{c_n}{f_c}\right) + a_в\left(\frac{p_n s_n}{r_n(p_n, p_{-n})} + k_c c_n f_c^2\right)$$

$$U_{c,n}\left(p_n + \varepsilon_n p_{-n} + \varepsilon_{-n}\right) = a_t$$

Відповідно до визначення неперервності функцій, оскільки $\lim_{\varepsilon_n \to 0}\left(p\right| \left|n + \varepsilon_n\right) = p_n$ та $\lim_{\varepsilon_n \to 0}$, отже робимо висновок, що $\lim_{\varepsilon_n \to 0} U_{c,n}\left(p_n + \varepsilon_n p_{-n} + \varepsilon_{-n}\right) = U_{c,n}\left(p_n, p_{-n}\right)$. Це значить, що функція яку представлено у (3.46) є безперервною на p_n. Таким чином рівновага Неша існує.

Крім того, для задачі контролю живлення гри *G* багатокористувацької системи, в якій стратегічний простір ***p*** рівновага Неша є унікальною. Доведення цього твердження також приведено нижче.

Згідно з концепцією та властивостями стиснутого відображення в безперервних іграх, для відображення $F : X \to X$, де X – це закритий набір, якщо $\left|F(\mathrm{x}) - \mathrm{F(y)}\right| v \le \beta v \left|x - y\right| \nu$ утримується для $\forall x, y \in X$ та $\beta \in$ тоді відображення F є скороченням та покриттям; більше того, F має унікальну нерухому точку.

Для формули представленої у [29], якщо $\exists \beta \in$, робить $\mathrm{v} U_{c,n}(p_{n,1}) - U_{c,n}(p_{n,2})\,\mathrm{v} \le \beta\,\mathrm{v}\left|p_{n,1} - p_{n,2}\right|\mathrm{v}$ утриманим, тоді гра контролю живлення G має унікальну рівновагу Неша [29].

3.2.4. Алгоритм контролю випромінюваної потужності на основі теорії ігор

Згідно з [28, 29], для гри, що існує у рівновазі Неша та є унікальною, найкращою стратегією реагування, яка використовується для отримання рівноваги Неша, є конвергенція. Оскільки існування та унікальність гри G було доведено, то можемо використати найкращу стратегію реагування для отримання рівноваги Неша у цьому алгоритмі. Для цього, ***a*** та p_{-n}, найкраща відповідь користувача *n`s.*

Найкраща відповідь користувача *n*, поміченого, як p_n може бути обчислена вирішенням проблеми оптимізації, яка представлена в (3.48), основуючись на заданому ***a*** та p_{-n}.

Для заданих a та , найкраща стратегія реагування p_n користувача мобільного пристрою *n* існує.

Для заданих а та p_n, функція, що представлена в (3.46) є продовженням p_n, де $p_n \in [p_t, p_{\max}]$. Більше того, диференціал першого порядку (3.46) на p_n є:

$$U_{c,n}^{`}(p_n) = \frac{a \cdot a_в \ln\left(1 + \frac{p_n G_n}{\Delta n}\right) - \frac{G_n (a_t + a_в p_n)}{\Delta n + p_n G_n}}{\ln^2\left(1 + \frac{p_n G_n}{\Delta n}\right)} \tag{3.49}$$

де $a = s_n \ln 2 / \mathrm{w}_{sc_k}$, та $\Delta_n = \eta_0 + I_n$ – це сума інтерференції та шуму. Коли $U_{c,n}^{`}(p_n) = 0$, (3.46) може приймати значення екстремуму. Припустімо, що $U_{c,n}^{`}(p_n) = 0$, (3.49) дорівнює:

$$\left(\frac{\Delta_n + p_n G_n}{\Delta n_в}\right)^{a_в(\Delta n + p_n G_n)} = e^{a_t G_n - \Delta_n a_n} \tag{3.50}$$

Оскільки (3.50) є трансцендентним рівнянням, аналітичне рішення (3.50) не існує; при цьому числове рішення (3.50) може бути отримано методом Ньютона. Далі ми перевіримо існування чисельних розв'язків:

$$f(\mathrm{p}_n) = \left(\frac{\Delta_n + p_n G_n}{\Delta n^{в}}\right)^{a_в(\Delta n + p_n G_n)} \tag{3.51}$$

Далі $f'(p_n)$ може бути розрахована наступним чином:

$$f`(\mathrm{p}_n) = \left(\frac{\Delta_n + p_n G_n}{\Delta n^{в}}\right)^{a_в(\Delta n + p_n G_n)} * \tag{3.52}$$

Якщо (3.52) дорівнює нулю, тоді $\ln\left(1 + \frac{p_n G_n}{\Delta n^{в}}\right) = -1$, що означає $p_n G_n = 0$. Оскільки $p_n G_n > 0$ утримується для $\forall p_n \in [p_t, p_{\max}]$ отже $f`(\mathrm{p}_n) > 0$ ; це значить що $f`(\mathrm{p}_n)$ – це зростаюча функція для $\forall p_n \in [p_t, p_{\max}]$.

Припускаючи, що рішення (3.50) це $p_n^{`}$ тоді $p_t > p_n^{`}$, що дорівнює $(\frac{\Delta n + p_n G_n}{\Delta n^{в}})^{a_в(\Delta n + p_n G_n)} > e^{a_t G_n - \Delta_n a_n}$, тоді для $\forall p_n \in [p_t, p_{\max}]$, $U_{c,n}^{`}(p_n) > 0$ утримується і $U_{c,n}(p_n)$ – це зростаюча функція. Таким чином, найкраща відповідь буде отримана коли $p_n = p_t$. Якщо $p_t < p`_n$, що дорівнює $(\frac{\Delta n + p_n G_n}{\Delta n^{в}})^{a_в(\Delta n + p_n G_n)} > e^{a_t G_n - \Delta_n a_n}$, тоді для $\forall p_n \in [p_t, p_{\max}], U_{c,n}^{`}(p_n) < 0$ утримується і $U_{c,n}(p_n)$ це спадаюча функція. Отже, найкраща відповідь буде отримана $p_n < p_{\max}$. Коли $p_t < p`_n < p_{\max}$, що дорівнює

$(\frac{\Delta n + p_n G_n}{\Delta n^{\varepsilon}})^{a_в(\Delta n + p_n G_n)} < e^{a_t G_n - \Delta_n a_n} < (\frac{\Delta n + p_n G_n}{\Delta n^{\varepsilon}})^{a_в(\Delta n + p_n G_n)}$, тоді якщо $p_n \in \left[p_t, p_n^{`}\right]$, $U_{c,n}^{`}(p_n) < 0$ утримується і $U_{c,n}(p_n)$ – це зростаюча функція; якщо $p_n \in [p_n', p_{max}], U_{c,n}'(p_n) > 0$ утримується і $U_{c,n}(p_n)$ – це зростаюча функція; найкраща відповідь буде отримана коли $p_n < p^{`}_n$. Більше того:

$$p_n \in \arg\min\left\{U_{c,n}\left(p_t, p_{-n}\right)U_{c,n}\left(p_n^{`}, p_{-n}\right)U_{c,n}\left(p_{\max}, p_{-n}\right)\right\}$$

На основі найкращої стратегії реагування, ми представляємо слотову структуру часу в передачі контролю потужності протягом розвантаження обчислювальної задачі. Стратегія потужності користувачів мобільними пристроями оновлюється на початку кожного слоту. Для мобільних користувачів, на початку кожного часового слоту, маленька секція БС вимірює перешкоди каналу і розподіляє канали зв'язку мобільним користувачам, У цьому алгоритмі, підхід бездротового інтерфиренного вимірювання такий самий, як і в (3.43):

$$\varphi_n\left(sc_k, p_{-n}(t)\right) = \begin{cases} \eta_m(p_n(t)) - p_n G_n, \text{якщо } sc_k = m \\ \eta_m(p_n(t)), \text{інакше} \end{cases} \tag{3.53}$$

Де $m \in sc$ – це канал, що розрахований BS, $\eta_m(p_n(t)) = \sum_{i \in N, a_i = a_n} p_i G_i$ – це розрахована інтерференція каналу m та $\varphi_n\left(sc_k, p_{-n}(t)\right)$ – це інтерференція користувача мобільного пристрою n. Коли користувач мобільним пристроєм отримує зворотню відповідь інтерференсного каналу с малого стільника BS, користувач мобільного пристрою буде розраховувати найкращу відповідь потужності передачі. Тоді кожен користувач вирішує, чи варто оновлювати цю стратегію потужності чи ні, опираючись на:

$$\begin{aligned} D_n(t) \in \{p_n : p_n = \arg\min_{p_n \in P, n \in N} U_{c,n}\left(p_n, p_{-n}(\mathrm{t})\right), \\ U_{c,n}\left(p_n^*, p_{-n}(\mathrm{t})\right) < U_{c,n}\left(p_n(t), p_{-n}(\mathrm{t})\right)\} \end{aligned} \tag{3.54}$$

Якщо $D_n(t) = \varnothing$, тоді у слоті часу $t+1$, потужність передачі користувача n дорівнює цьому у слоті t; інакше якщо $D_n(t) = \varnothing$ користувач n буде обновляти свою стратегію передачі у слоті $t+1$ для p_n. Цей процес буде повторений допоки гра

контролю потужності G дійде до рівноваги Неша. Цей процес можна представити за допомогою наступного алгоритму (рис. 3.4).

Алгоритм контролю потужності (рис .3.4) працює наступним чином.

1. Спочатку обчислюється початковий профіль потужності передачі $p_0 = (p_{10}, p_{20}, ..., p_{N0})$ та профайл рішення про розвантаження $a_0 = (a_{10}, a_{20}, ..., a_{N0})$.

2. Кожен користувач *n* розраховує інтерференцію у кожному часовому слоті *t;* цей процес буде повторюватися всіма користувачами паралельно.

3. Користувач *n* розраховує найкращу відповідь p_n основуючись заданому профілю рішення про розвантаження a та розрахованій інтерференції.

4. Якщо $D_n(t) = \varnothing$, тоді користувач *n* обновлює його рівень потужності як $p_n(t+1) = p_n(t)$.

5. В іншому випадку, користувач *n* обновлює його передачу потужності як $p_n(t+1) = p_n^*$.

Цей алгоритм повторюється допоки не отримається рівновага Неша.

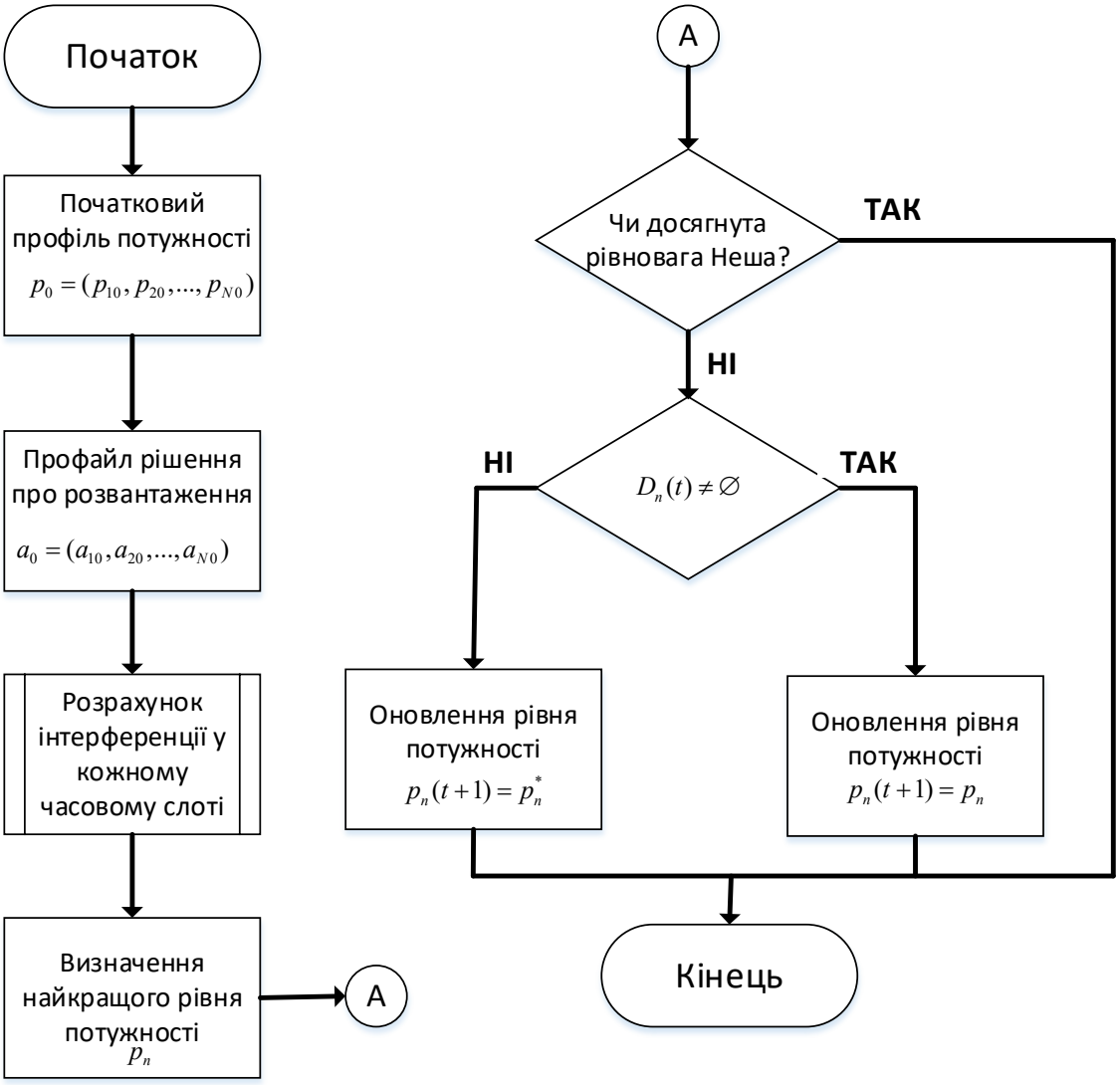


Рис. 3.4. Алгоритм контролю потужності для розвантаження завдань в системі з граничними обчисленнями

Надалі дослідимо конвергенцію та складність обчислення на основі запропонованого алгоритму, який заснований на теорії ігор.

Як показано в [29], оскільки рівновага Неша гри G існує і є унікальною, тому алгоритм є конвергентним.

Як вже було зазначено, для розрахунку оптимальної потужності передачі p_n, застосовується метод Ньютона. Доведено, що для функції $f(x)$, складність Ньютоновського Методу 0 , де $F(\mathrm{n})$ - це ціна розрахунку $f(x)/f`(x)$. Для гри G, оскільки $f(n)$ представлено в (3.46), отже $f(x)/f`(x)$ може бути розраховано як:

$$\frac{U_{c,n}(p_n)}{U_{c,n}^{`}(p_n)} = \left(\frac{a_в}{(a_t + a_в p_n)} - \frac{G_n (a_t + a_в p_n)^2 \ln\left(1 + \frac{p_n G_n}{\Delta n}\right)}{\Delta n + p_n G_n} \right)^{-1} \quad (3.55)$$

Як показано в (3.55), складність розрахунку в більшості походить з другого члена функції; тому складність обчислення $F(\mathrm{n})$. Тому обчислювальна складність цього алгоритму є $o\left[n^2 \log(n)\right]$, де $F(\mathrm{n})$ є $o\left[n^2 \log(n)\right]$. Це демонструє, що алгоритм, який було запропоновано у роботі, може бути виконаний впродовж поліноміального часу.

Слід звернути увагу на те, що алгоритм надасть змогу досягти рівновагу Неша принаймні після C круглої ітерації, де $C = \frac{\left(a_t + a_в p^{'}\right)}{\Delta p_{\max} \ln\left(1 + \frac{p^{'} G}{\Delta_{\max}}\right)}$ та

$p^{'} \square arg\max\left\{U_{c,n}\left(p_n^{`}, p_{-n}\right), \left(p_{\max}, p_{-n}\right)\right\}$

Нижня межа числа ітерації може бути розрахована як:

$$C = \frac{\left(a_t + a_в p^{'}\right)}{\Delta p_{\max} \ln\left(1 + \frac{p^{'} G}{\Delta_{\max}}\right)} \quad (3.56)$$

Оскільки в разовій ітерації складність обчислень є $O\left[\log(n)\mathrm{F}(n)\right]$, отже загальна складність обчислень для отримання рівноваги Неша гри G буде не меншою $O\left[\mathrm{C}\log(n)\mathrm{F}(n)\right]$.

Висновки до розділу 3

Дослідження, проведені в другому розділі, надали змогу отримати наступні результати.

1. В даному розділі були досліджені завдання вивантаження завдання, розподілу ресурсів додатків та планування завдань в мережі МЕС. Враховуючи складність вирішення проблеми динамічного планування та розвантаження завдань, було представлено нову стратегію декомпозиції, що реалізує методику розкладу

Бендерса. Розроблений метод розбиває задачу на основну, яка вирішує завдання з завантаженням та розподілом ресурсів програми; і декілька вторинних підзадач, кожна з яких звертається до планування завдань в одному використовуваному додатку.

2. Розроблений метод є доволі точним і надає можливість бути припиненим при будь-якій ітерації, таким чином, реалізуючи компроміс між якістю та часом проведення обчислень.

3. Крім цього, в даному розділі було запропоновано алгоритм контролю потужності, що базується на теорії ігор, для багатокористувацьких систем із забезпеченням граничної обробки даних, та під впливом дії завад, що враховує як інтерференцію, так і багатокористувацький сценарій.

4. Для алгоритму заснованого на теорії ігор, найголовніші властивості це існування та унікальність рівноваги Неша; ці дві властивості відповідають ефективності та дієвості алгоритму. Тому в даному розділі було доведено існування рівноваги Неша в такого типу грі, крім того вона є унікальною.

Список використаних джерел у третьому розділі

1. End-to-end QoE optimization through overlay network deployment / De Vleeschauwer B., De Turck F., Dhoedt B. et al. *2008 International Conference on Information Networking.* 2008, January. P. 1–5.

2. DeepNetQoE: Self-adaptive QoE optimization framework of deep networks / Wang R., Chen M., Guizani N. et al. 2020. (arXiv preprint arXiv:2007.10878).

3. Affective content-aware adaptation scheme on QoE optimization of adaptive streaming over HTTP / Hu S., Xu M., Zhang H. et al. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM).* 2019. Vol. 15, No. 3s. P. 1–18.

4. Learning from experience: A dynamic closed-loop QoE optimization for video adaptation and delivery / Triki I., El-Azouzi R., Haddad M. et al. *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC).* 2017, October. P. 1–5.

5. Study of user QoE improvement for dynamic adaptive streaming over HTTP (MPEG-DASH) / Zhao S., Li Z., Medhi D. et al. *2017 International Conference on Computing, Networking and Communications (ICNC).* 2017, January. P. 566–570.

6. Choi Y., Lim Y. Optimization approach for resource allocation on cloud computing for IoT. *International Journal of Distributed Sensor Networks.* 2016. Vol. 12. No. 3.

7. Shah-Mansouri H., Wong V. W. Hierarchical fog-cloud computing for IoT systems: A computation offloading game. *IEEE Internet of Things Journal.* 2018. Vol. 5, No. 4. P. 3246–3257.

8. Deng, R., Lu, R., Lai, C., Luan, T. H., & Liang, H. (2016). Optimal workload allocation in fog-cloud computing toward balanced delay and power consumption. *IEEE internet of things journal*, *3*(6), 1171-1181.

9. Edge computing in IoT-based manufacturing / Chen B., Wan J., Celesti A. et al. *IEEE Communications Magazine.* 2018. Vol. 56, Iss. 9. P. 103–109.

10. Ананьевский И. М., Анохин Н. В., Овсеевич А. И. Синтез ограниченного управления линейными динамическими системами с помощью общей функции Ляпунова. *Доклады Академии наук.* 2010. Vol. 434, No. 3, P. 319–323.

11. Deep reinforcement learning based computation offloading and resource allocation for MEC / Li J., Gao H., Lv T., Lu Y. *2018 IEEE Wireless Communications and Networking Conference (WCNC).* 2018, April. P. 1–6.

12. Partial offloading scheduling and power allocation for mobile edge computing systems / Kuang Z., Li L., Gao J. et al. *IEEE Internet of Things Journal.* 2019. Vol. 6, No. 4. P. 6774–6785.

13. Mao Y., Zhang J., Letaief K. B. Joint task offloading scheduling and transmit power allocation for mobile-edge computing systems. *2017 IEEE wireless communications and networking conference (WCNC).* 2017, March. P. 1–6.

14. Cheon H. R., Lee S. Q., Kim J. H. Traffic offloading algorithm using social context in MEC environment. *The Journal of Korean Institute of Communications and Information Sciences.* 2017. Vol. 42, No. 2. P. 514–522.

15. Dynamic task offloading and scheduling for low-latency IoT services in multi-access edge computing / Alameddine H. A., Sharafeddine S., Sebbah S. et al. *IEEE Journal on Selected Areas in Communications.* 2019. Vol. 37, Iss. 3. P. 668–682.

16. OpenFlow: enabling innovation in campus networks / McKeown N., Anderson T., Balakrishnan H. et al. *ACM SIGCOMM computer communication review.* 2008. Vol. 38, No. 2. P. 69–74.

17. Alghamdi K., Braun R. Software defined network (SDN) and OpenFlow protocol in 5G network. *Communications and Network.* 2020. Vol. 12, No. 01. P. 28.

18. Enabling communication technologies for automated unmanned vehicles in industry 4.0 / Fellan A., Schellenberger C., Zimmermann M., Schotten H. D. *2018 International Conference on Information and Communication Technology Convergence (ICTC).* 2018, October. P. 171–176.

19. Назначение заданий на процессоры в системах реального времени / Колесов Н. В., Скородумов Ю. М., Толмачева М. В., Юхта П. В. *Материалы XXVIII конференции памяти выдающегося конструктора гироскопических приборов Н. Н. Острякова.* 2012. С. 57–58.

20. Тарнавский А. Г., Чесноков С. С., Жибинов С. Б. Современное состояние компьютерного моделирования в интернете: краткий обзор сайтов. *Проблемы інформатики.* 2009. № 2.

21. Geoffrion A. M. Generalized benders decomposition. *Journal of optimization theory and applications.* 1972. Vol. 10, Iss. 4. P. 237–260.

22. Hooker J. N., Ottosson G. Logic-based Benders decomposition. *Mathematical Programming.* 2003. Vol. 96, Iss. 1. P. 33–60.

23. Shahidehopour M., Fu Y. Benders decomposition: applying Benders decomposition to power systems. *IEEE Power and Energy Magazine.* 2005. Vol. 3, Iss. 2. P. 20–21.

24. Resource allocation for intelligent reflecting surface aided wireless powered mobile edge computing in OFDM systems / Bai T., Pan C., Ren H. et al. 2020. (arXiv preprint arXiv:2003.05511).

25. Energy efficiency based joint computation offloading and resource allocation in multi-access MEC systems / Yang X., Yu X., Huang H., Zhu H. *IEEE Access.* 2019. Vol. 7. P. 117054–117062.

26. Yang H. C., Alouini M. S. Order statistics in wireless communications: diversity, adaptation, and scheduling in MIMO and OFDM systems. Cambridge University Press, 2011.

27. Resource allocation for OFDM-based maritime edge computing networks / Wang H., Wang Y., Ma Y., Lin B. *2020 13th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI).* 2020, October. P. 983–988.

28. Energy-efficient joint offloading and wireless resource allocation strategy in multi-MEC server systems / Cheng K., Teng Y., Sun W. et al. *2018 IEEE international conference on communications (ICC).* 2018, May. P. 1–6.

29. Власов Д. А. Визуализация равновесия Нэша в биматричных играх средствами Wolfram // *Успехи современной науки*. – 2016. – Т. 1. – № 10. – С. 156–158.

30. Кравець П. Ігрова задача взаємодії елементів мультиагентних систем. *Современные информационные технологии и ИТ-образование.* 2006. Т. 12, № 4.

РОЗДІЛ 4
ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ РОЗРОБЛЕНИХ МЕТОДІВ ТА МОДЕЛЕЙ

В даній лисертаційній роботі (розділи 1-3) було проведено розробку та удосконалення наступних методів: методу оптимізації розміщення масштабованих послуг на розподілених обчислювальних ресурсах мережі стільникового оператора; методу керування випромінюваною потужністю мобільних пристроїв під час розвантаження завдань та методу динамічного планування та розвантаження завдань в системах з розподіленими та граничиними обчисленнями.

З метою оцінки ефективності запропонованих рішень в даному розділі дисертаційної роботи наведені результати проведеного комп'ютерного моделювання та відповідно проведено аналіз основних результатів.

4.1. Оцінка ефективності розробленого методу оптимізації розміщення масштабованих послуг на розподілених обчислювальних ресурсах мережі стільникового оператора

Першим розробленим методом був метод оптимізації розміщення масштабованих послуг на розподілених обчислювальних ресурсах мережі стільникового оператора.

В даному підрозділі проводилась оцінка розробленого метода та його порівняння із реалізаціями інших методів оптимізації, застосованих до вирішення задачі розміщення масштабованих послуг.

Моделювання роботи різних алгоритмів проводились у програмному середовищі MathCad. Для моделювання вищерозглянутих алгоритмів використовувались наступні параметри (табл. 4.1).

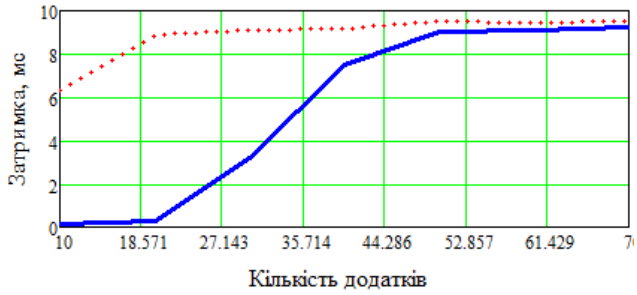
Таблиця 4.1 – Параметри для моделювання досліджуваних алгоритмів

Параметр	**Значення параметра**
Кількість базових станцій	10, 20
Середня затримка мережевого з'єднання	БС-БС: 1 мс; БС-ядро: 1 мс; БС-хмара: 10 мс.
Процесорна потужність (MIPS)	Хмара: необмежена; Ядро мережі: 200000; БС: 50000.
Кількість користувачів	1000, 4000, 7000, 10000
Кількість запущених додатків	10, 20, 30, 40, 50

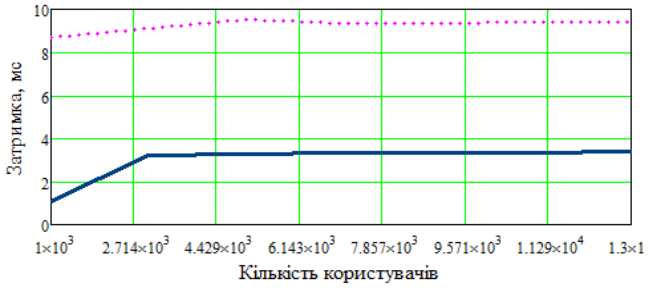
В таблиці 4.1 приведені значення ключових показників продуктивності (KPI) мереж 5G [1]. Для досліджень використовувались KPI для трьох основних типів додатків, які плануються до розгортання в мережах 5G: eMBB [2], URLLC [2] та mMTC [3]. Для спрощення моделювань, були зроблені припущення, що користувачі рівномірно розподілені між базовими станціями, розташованими в шестикутній сітці. Обчислювальні вузли розміщуються в різних мережевих частинах (базові станції, опорна мережа та в хмарі), і їх потужність зменшується, коли вони спускаються в нижчи рівні топології мережі (тобто від хмари до базових станцій).

Резуьтати проведеного поделювання наведені нижче. На рис. 4.1 представлений вплив збільшення кількості додатків та користувачів на рівень порушення QoS в сценарії маломасштабної мережі.

На рис. 4.1 (a1) ми можемо бачити, що розроблений генетичний алгоритм демонструє підвищення рівня порушень QoS із збільшенням кількості застосувань. Поясненням такої поведінки є те, що розподіл максимальної кількості запитів на сервері не є задовільною стратегією в середовищі з високими потребами в ресурсах. На рис. 4.1 (a2) можна побачити, що збільшення кількості користувачів не має великого впливу на продуктивність.

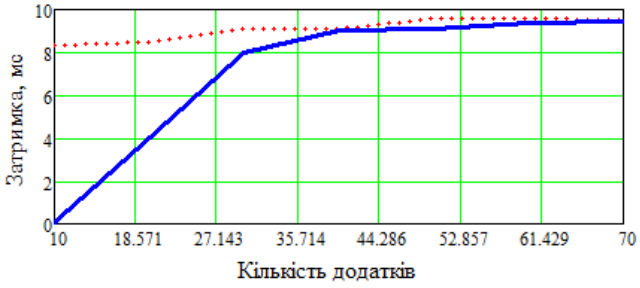


(а.1) Кількість користувачів – 4000

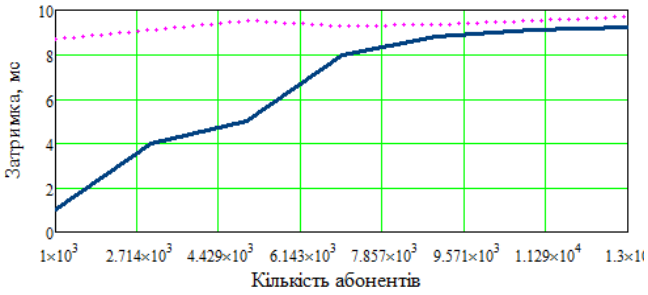


(а.2) Кількість додатків – 30

(а) для мережі із 10 базових станціій gNb



(б.1) Кількість користувачів – 4000



(б.2) Кількість додатків – 30

(б) для мережі із 20 базових станціій gNb

Рис. 4.1. Результати моделювання досліджуваних алгоритмів

У другому сценарії були проаналізовані методи розміщення в мережі з 20 базовими станціями (1 центральна базова станція, кільце з 6 базових станцій навколо центрального та 12 базових станцій навколо першого кільця). У цьому сценарії Хмара, Жадібний та Генетичний алгоритми мають таку поведінку, як у дрібномасштабній мережі, коли спостерігається збільшення додатків, як показано на рис. 4.1 (б).

Рис. 4.1 (а2) та 4.1 (б2) представляють вплив на ефективність збільшення кількості користувачів. Ми можемо спостерігати збільшення рівня порушень від 1000 до 4000 користувачів, і тоді воно залишається майже до 10000 користувачів, головним чином для обох досліджуваних рішень. Таку поведінку можна пояснити розміщенням програм у хмарі, що збільшує затримку в мережі. З іншого боку, хмарний вузол має необмежену кількість ресурсів для прийому користувачів, не сильно впливаючи на продуктивність обробки.

4.2. Дослідження ефективності методу керування випромінюваною потужністю мобільних пристроїв під час розвантаження завдань

У цьому підрозділі дослідимо ефективність методу керування випромінюваною потужністю мобільних пристроїв під час розвантаження завдань шляхом теоретичного аналізу та подальшргр чисельного моделювання.

4.2.1 Теоретичний аналіз

У цьому підрозділі проведемо оцінку ступеня анархії (PoA) загального обсягу розрахунків накладних витрат з точки зору всіх користувачів мобільних пристроїв, $\sum_{n\in N} U_{c,n}$. Виходячи з теоретичних положень, викладених в [4], можемо визначити PoA як:

$$P_0 A = \frac{\sum_{n\in N} U_{c,n}(p')}{\sum_{n\in N} U_{c,n}(p)} \tag{4.1}$$

де p є рівновагою Неша гри G та p – це централізоване оптимальне рішення яке робить значення $\sum_{n\in N} U_{c,n}$ мінімальним.

Як було представлено в [4], для мережевих обчислювальних накладних витрат, найменший PoA, означає кращу ефективність системи.

Так, для багатокористувацької гри контролю потужності в системі із забезпеченням граничних обчислень, PoA мережевих обчислюваних накладних витрат задовольняє наступну умову:

$$1 \le P_0 A \le \frac{\sum_{n=1}^{N} U_{c,n}^{\max}}{\sum_{n=1}^{N} U_{c,n}^{\min}} \tag{4.2}$$

де $U_{c,n}^{\min} = \frac{\left(a_t + a_в p'\right) s_n}{w_{sc_k} \log_2\left(1 + \frac{p_n G_n}{\eta_0}\right)}, U_{c,n}^{\max} = \frac{\left(a_t + a_в p_n\right) s_n}{w_{sc_k} \log_2\left(1 + \frac{p_n G_n}{\eta_0 + \sum_{i\in N\setminus\{n\}, a_i = a_n} p'_{\max} G_i}\right)} + a_t t_{c,n} + a_в e_{c,n}$ та

$p'_{\max} \in arg\max\left\{p_i, i \in N maa_i = a_n\right\}$. $p'_{\max}$ означає максимальну потужність передачі користувача мобільного пристрою у наборі користувачів інтерференції φ_n.

Зауважимо, що для вищенаведеного висновку, коли інтерференція від перешкоджаючих користувачів зменшується, PoA зменшується, а це в свою чергу демонструє, що ефективність рівноваги Неша може бути покращена коли інтерференція зменшується.

4.2.2. Чисельне моделювання

У цьому підрозділі можемо оцінити ефективність запропонованого методу за допомогою чисельного моделювання. У цьому моделюванні діапазон покриття малих стільників BS складає 50 м; 20 мобільних користувачів розміщені випадковим чином у зоні покриття BS. Пропускна здатність бездротового каналу каналу становить 5 МГц. Передача потужності користувача може бути врегульована з p_i до $p_{\max}$; p_t та може бути розраховано до порогу SINR та вимірюється інтерференцією; $p_{\max}$ встановлюється на 150мВат. Шум – 100ДБ. Посилання каналу – $G_n = d_{n,s}^{-a}$, де $d_{n,s}^{-a}$ – це відстань між користувачем та BS; α – коефіцієнт втрати шляху, який встановлений на рівні 4 у цьому моделюванні. Аналогічно [5], у цьому моделюванні, $b_n = 50000$ кБ та $d_n = 1000$ мегациклів. Можливість обчислення процесора f_c – 10ГГц. Рішення важить $a_t, a_в \in$ та $a_t + a_в = 1$, отже ми $a_t \in \{1, 0.5, 0\}$.

У традиційному локалізованому оптимальному підході, кожен користувач розраховує оптимальну потужність передачі на основі вимірюваного каналу інтерференції та регулює потужність передачі до оптимального значення. Різницею підходу основаного на теорії ігор є те, що під час регулювання потужності передачі умови, які представлені в методі не враховуються у традиційному підході, тому він не може гарантувати отримання рівноваги Неша.

Числові результати представлені на рис 4.2 та 4.3. Отримані дані надають змогу виявити, що гра G є ковергентною, та рівновага Неша існує і є унікальною. Перш ніж отримати рівновагу Неша (тобто кількість часових слотів менше ніж 40), накладні витрати кожного користувача змінюються зі збільшенням часових слотів, і ці зміни є нерегулярними; після отримання рівноваги Неша (тобто кількість часових

інтервалів перевищує 40), накладні витрати зберігають константу. Однак для різних мобільних користувачів кінцеві надмірні витрати (тобто значення рівноваги Неша) є різними. Накладні витрати мобільних користувачів перед отриманням рівноваги Неша (тобто кількість часових інтервалів менше 40) може бути більше або менше ніж після отримання рівноваги Неша (тобто коли кількість часових інтервалів більше 40). З рис. 4.3 можна зробити висновок, що накладні витрати мобільних користувачів під рівновагою Неша зменшуються порівняно з початковими значеннями, що демонструє, що підхід до контролю потужності заснований на теорії ігор може зменшувати накладні витрати успішно.

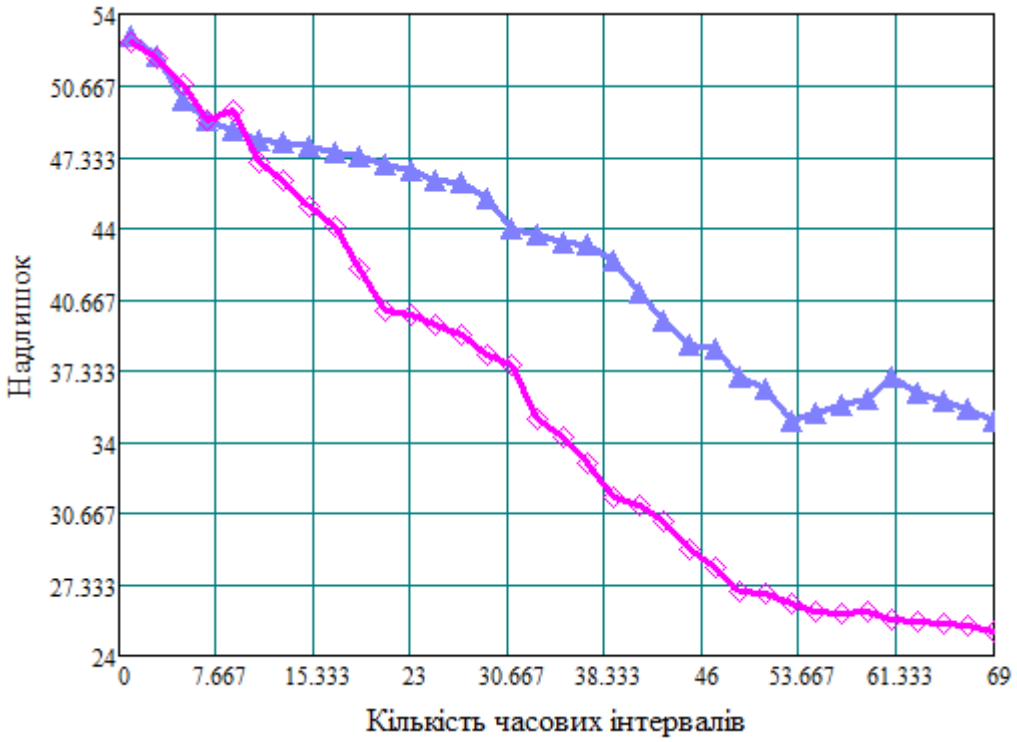


Рис.4.2. Величина накладних витрат за різної кількості часових інтервалів

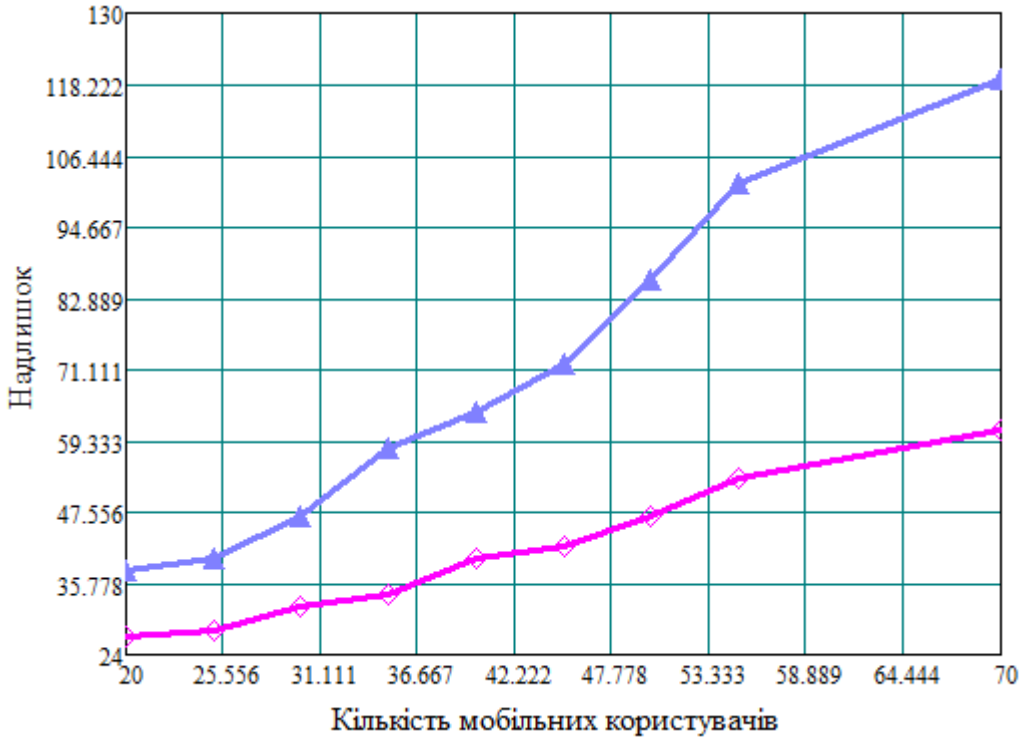


Рис. 4.3. Величина накладних витрат за різною кількістю мобільних користувачів

Рис. 4.2 ілюструє накладні витрати всієї мережі. Є дві криві мережевих накладних витрат, показані на рис. 4.2, для однієї – підхід заснований на теорії ігор та інший традиційний локалізований оптимальний підхід. Рис. 4.2 демонструє велику перевагу алгоритму контролю потужності, що основується на теорії ігор , в плані зменшення мережевих накладних витрат. Зі збільшенням часу моделювання тобто кількість часових слотів), мережеві надмірні витрати цих двох алгоритмів зменшуються. Однак, зменшення в підході , який базується на теорії ігор набагато гостріші, ніж в традиційному локалізованому оптимальному підході. Причиною цьому є те, що завдяки грі між різними мобільними користувачами, усі мобільні користувачі у мережі мають мінімальні накладні витрати згідно з іншими користувацькими стратегіями потужності; однак, оскільки в локалізованому оптимальному алгоритмі кожен мобільний користувач вирішує потужність тільки базуючись на його власних накладних витратах, тому контроль потужності навряд чи може відповідати статусу рівноваги. Отже, зменшення накладних витрат у

оптимальному локалізованому алгоритмі повільніше, ніж у алгоритмі, який базується на теорії ігор.

Більше того, після досягнення рівноваги Неша, мережеві накладні витрати в алгоритмі, створеному на основі теорії ігор, низькі та стабільні; однак ці накладні витрати все ще змінюються в локалізованому оптимальному алгоритмі.

На рис. 4.3 показані мережеві накладні витрати при різній кількості мобільних користувачів у мережі. Зі збільшенням кількості користувачів, мережеві накладні витрати в обох алгоритмах збільшуються; однак, збільшення в локалізованому алгоритмі контролю потужності є швидшим та довшим, ніж у тому, що базується на теорії ігор, особливо, коли кількість користувачів велика. Причина в тому, що зі збільшенням кількості користувачів в мережі, складність розрахунку та гри G збільшується; однак, як показано на рис. 4.2, оскільки мережеві накладні витрати в підході, що базується на теорії ігор, зберігають стабільність в кінці й значно менші, ніж у локалізованому оптимальному підході, тому збільшення мережевих накладних витрат у підході теорії ігор значно менші, ніж у оптимальному локалізованому підході. Більше того, чим більше мобільних користувачів у мережі, тим очевиднішою є перевага підходу теорії ігор. Це демонструє, що такий підхід є більш ситуативно-підходящим для широкомасштабних мереж, ніж локалізований оптимальний підхід.

4.3. Дослідження ефективності методу динамічного планування та розвантаження завдань в системах з розподіленими та граничиними обчисленнями

В даному підрозділі проведемо широке емпіричне дослідження для оцінки ефективності нашого підходу в порівнянні з іншими відомими. Далі будуть досліджуватись інженерний вплив проблеми за різних системних параметрів та вимог QoE. Підкреслимо вплив різних вирішених проблем (тобто розвантаження завдань, розподілу ресурсів додатків та планування завдань) на обслуговування

декількох вертикальних галузей, аналізуючи ефективність запропонованого методу для різних типів додатків (табл. 4.2).

Таблиця 4.2 – Вимоги до затримки для різних індустрій

Вертикальна індустрія	**Допустиме значення затримки, мс**	**Поточне значення затримки, мс**
Тактильний Інтернет	1-10	7
Автоматизація виробництва	0.25 – 10	10
Інтелектуальні траспортні системи	10 – 100	50
Віддалене оперування	250	110

У нашому чисельному дослідженні ми розглядаємо мережі різного розміру з різною кількістю серверів MEC, кожен має ємність $c_m = 20$ ГГц. Ми враховуємо $|T| = 5$ різних типів різної кількості застосувань, які належать до однієї галузі вертикалі (якщо не вказано інше). Кожна програма вимагає мінімальних обчислювальних ресурсів (p_{amin}) генерований випадковим чином між [2 – 5] ГГц. Програми випадковим чином розміщуються на серверах MEC. Припускаємо, що декілька завдань з розвантаження UE, що належать до різних галузей, і, отже, є різними вимогами QoE. Вважаємо, що кількість циклів (u), що вимагаються UE, випадковим чином генерується між [20 100] циклами. Затримки завантаження та від краю до кінця завантажених завдань генеруються випадковим чином між 1- 2 мс та 1- 3 мс відповідно. Усі наші числові оцінки у середньому оцінюються за 5 наборів. Вони проводяться за допомогою версії 12: 4 Cplex для вирішення MIP на процесорі Intel i7-4790 на частоті 3:60 ГГц з 16 ГБ оперативної пам’яті.

Дослідження починаємо з оцінки ефективності запропонованого рішення по відношенню до DTOS-MIP з точки зору часу виконання, оскільки ми змінюємо кількість завантажених UE завдань. Збільшення кількості завантажених завдань ускладнює вирішення проблеми з огляду на обмежені обчислювальні ресурси. Отже,

також дивимось на вплив такого збільшення на рівень вступу. Таким чином, розглянемо мережу, що складається з $|M| = 3$ серверів MEC та $|A| = 15$ IoT-додатків $|T| = 5$ різних типів, що представляють кілька вертикальних галузей. Кінцеві терміни завантажених завдань генеруються випадковим чином між [5 – 20 мс. Отримані результати представлені в таблиці 4.3.

Таблиця 4.3 – Порівння результатів моделювання

	Час виконання		**Частка відхилених завдань (%)**	
Кількість користувачів	**DTOS-MIP**	**Розроблений метод**	**DTOS-MIP**	**Розроблений метод**
5	922	56	92	92
10	2359	116.4	84	84
15	15512.6	1214	76	76
20	251218.4	10077.4	67	67
25	3014109.8	21602.6	62.4	62.4

На рис. 4.4 інтерпретовані отримані значення.

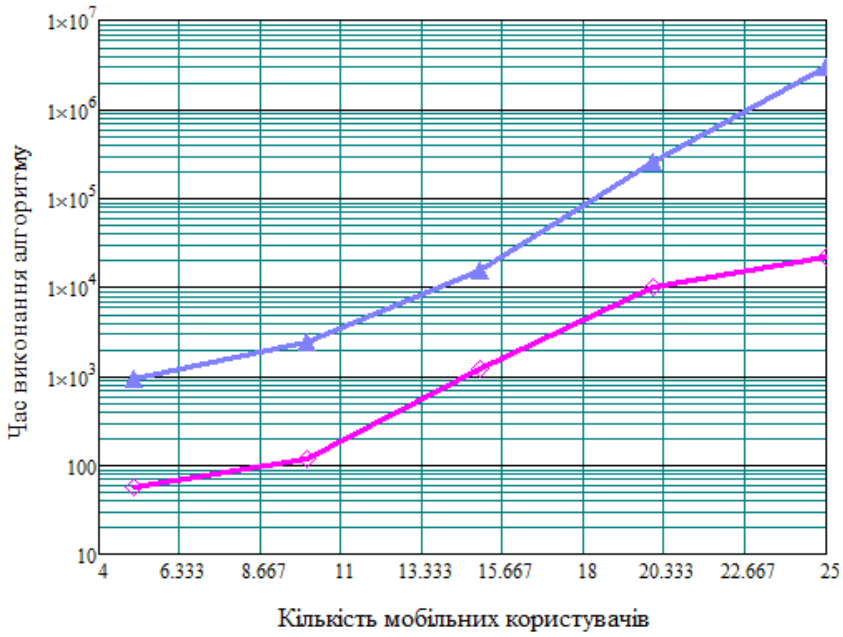


Рис. 4.4. Результати моделювання розробленого методу

З рис. 4.4 видно, що зі збільшенням кількості UE збільшується швидкість прийому. Таке зменшення очікується, оскільки більше завдань претендує на однакову кількість обчислювальних ресурсів, отже, деякі з них будуть страждати від великої затримки очікування деяких програм, очікуючи їх звільнення. Це негативно позначиться на їхніх вимогах до затримки, що призведе до пропуску строків та відхилення від мережі.

Дані щодо часу виконання також наведені в таблиці 4.3. Як видно, розроблений метод здатний забезпечити оптимальне рішення в середньому на 95% швидше, ніж DTOS-MIP. Це пояснюється тим, що він вчиться з якості розчину, що генерується при кожній ітерації, щоб вирізати подібні нездатні рішення з простору рішення. Це обмежує простір пошуку, оскільки кількість ітерацій збільшується, а отже, допомагає досягти оптимального рішення швидше, ніж DTOS-MIP. Крім того, розкладання задачі на кілька підзадач допомагає скоротити час виконання, особливо, що підзадачі для такого типу планування виконуються паралельно за допомогою потоків.

Для того, щоб оцінити ефективність розробленого методу, враховуємо один тестовий екземпляр і побудуємо на рис. 4.5 кількість допущених завдань на кожній ітерації, визначену основною та вторинними підзадачами. Розглядаємо мережу $|M| = 10$ MEC-серверів, що розміщують $|A| = 15$ додатків. Враховуємо завдання $|U| = 30$ UE, що належать вертикалі фабрики автоматизації ($\theta_u = 10$ мс).

На рис. 4.5 зображено, що мета вирішення основної задачі являє верхню межу оптимального цільового значення, тоді як кількість завдань, допущених підзадач, являє собою нижню межу. Зі збільшенням кількості ітерацій об'єктивне значення основної задачі зменшується, враховуючи, що до нього додається більше скорочень Бендерса. Навпаки, кількість завдань, допущених вторинними підзадачами, змінюється між ітераціями залежно від вимог завдань (тобто, кількості циклів, часу прибуття), що надсилаються основній задачі на кожне з них. Однак важливо зауважити, що оптимальне цільове значення завжди лежить між максимальною нижньою межею та мінімальною верхньою межею, досягнутою на даний момент. Крім того, дисперсія зазору, що існує між верхньою та нижньою межею, надає

можливість припинити виконання розробленого методу у будь-який час на основі бажаної якості рішення та часу виконання. Наприклад, можна припинити виконання розробленого методу за ітерацією 14 із зазором у 9% між верхньою та нижньою межею, що мало сканує якість розчину, отримуючи приблизно 75: 4% за час виконання. Якщо бажана краща якість рішення, можна зупинити в при ітерації 26, де зазор досягає 4: 5%; однак виграш у розрахунку на час обчислення становить близько 53%.

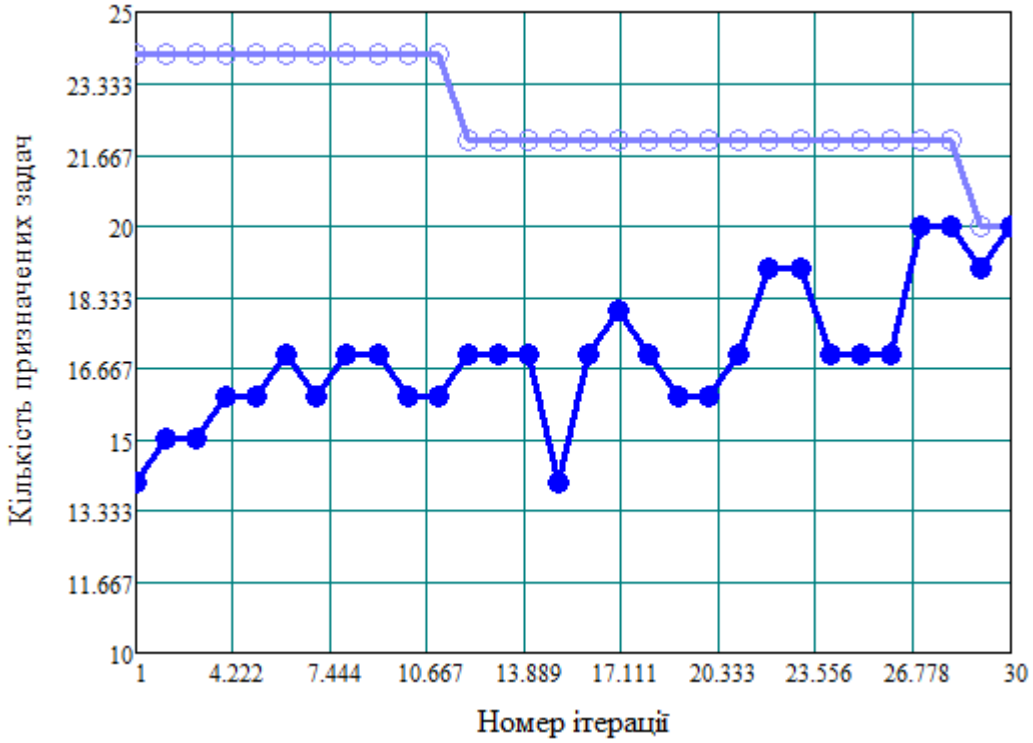


Рис. 4.5. Дослідження збіжності розробленого алгоритму

Щоб ще більше підкреслити той факт, що запропонований підхід являє собою алгоритм у будь-який час, який можна зупинити на будь-якій ітерації, забезпечуючи можливе рішення, ми показуємо в таблиці 4.4 усереднений час виконання методу, використовуючи ті самі мережеві налаштування, які були зазначені в попередньому пункті.

Таблиця 4.4 – Порівняння часу виконання досліджуваного алгоритму для оптимального рішення

Кількість користувачів	Час виконання алгоритму, мс
20	15365.9
30	284768.1
40	1521233.2
50	1659131.8

Результати, наведені в таблиці 4.4, зображають час виконання при оптимальному рішенні та при першому виникненні розриву оптимальності, який становить менше 10% або між 10% та 20%. Зрозуміло, що для визначеної кількості UE час виконання збільшується зі зменшенням розриву оптимальності. Насправді тривалість виконання методу збільшується зі збільшенням кількості ітерацій. У цьому випадку до основної задачі додається більше розрізів Бендерса, що затягує простір його розчину, отже, краще розміщувати оптимальне рішення, яке, ймовірно, зменшить зазор між верхньою межею, наданою основною задачею, і нижньою межею, заданою вторинними задачами. Таким чином, зупинка виконання розробленого методу на певному допустимому розриві може призвести до високих вигод у плані часу обчислень. Наприклад, коли$|N| = 30$, 97: 26% приросту в режимі виконання зображується, коли розрив становить від 10% до 20%, тоді як 89: 65% отримується з розривом менше 10%. Нарешті, варто відзначити, що зі збільшенням кількості UE збільшується час виконання розробленого методу із збільшенням розміру проблеми, а отже, проблему стає важче вирішити.

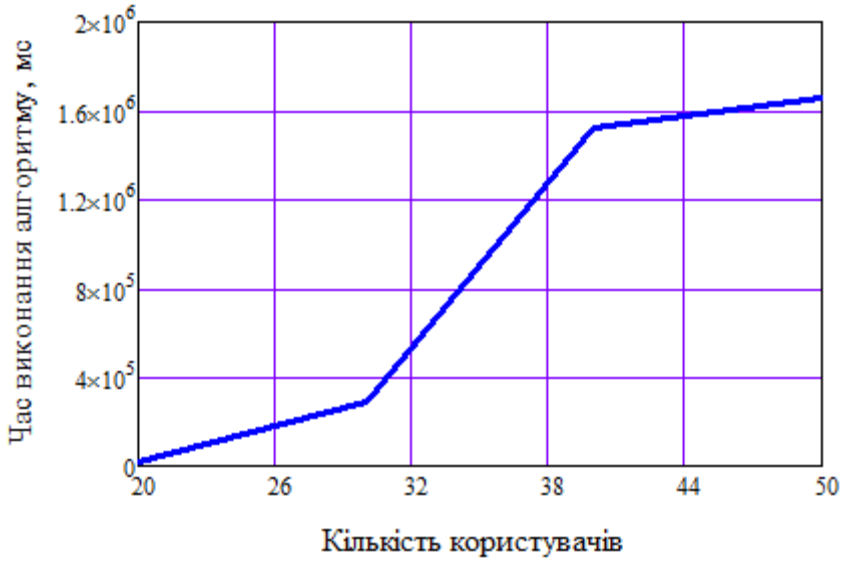


Рис. 4.6. Часові витрати на виконання алгоритму

Таким чином, вважаємо мережу$|N| = 10$ MEC-серверів, що розміщують $|A| = 15$ програм. Наші результати, представлені на рис. 4.5, показують, що при збільшенні кількості UE скорочення прийому зменшується для кожної вертикальної галузі, оскільки все більше завдань суперечить тим самим обчислювальним ресурсам (додатки), які перевантажуються і, отже, не відповідають вимогам до затримки всіх UE, які вимагають їх обслуговування. Насправді деякі завдання зазнають додаткових затримок очікування, що призведе до того, що вони пропустять свої терміни, і, таким чином, будуть відхилені від мережі. Однак такі затримки очікування можна допустити, якщо вимоги до затримки збільшаться.

Висновки до розділу 4

Дослідження, проведені в четвертому розділі, надали змогу отримати наступні результати.

1. Завдяки експериментам ми показали, що лінеаризація має низький рівень порушення терміну виконання програми у невеликому сценарії стільникової мережі. У сценарії з мережею, більшою за мережу малих масштабів, наші два підходи

(лінеаризація та генетична база) мають близькі результати, і обидва в цілому мали кращі результати, ніж просте жадібне рішення.

2. Шляхом моделювання було показано, що розроблений метод може досягти більш ніж 140 порядків покращення масштабів в умовах виконання в порівнянні з DTOS-MIP.

3. Було проаналызовано ефективність запропонованого алгоритму, заснованого на теорії ігор, шляхом теоретичного аналізу та чисельного моделювання. По-перше, було доведено, що запропонований алгоритм є збіжним (конвергентним), використовуючи найкращу відповідну стратегію; по-друге, було розраховано складність обчислення цього алгоритму, яка рівна O.

4. Результати моделювання показують, що алгоритм контролю потужності, що базується на теорії ігор, має великі переваги, щодо підвищення продуктивності багатокористувацької системи в умовах впливу завад.

Список використаних джерел у четвертому розділі

1. 5G new radio evaluation against IMT-2020 key performance indicators / Fuentes M., Carcel J. L., Dietrich C. et al. *IEEE Access.* 2020. Vol. 8. P. 110880–110896.
2. Yifei Y., Longming Z. Application scenarios and enabling technologies of 5G. *China Communications.* 2014. Vol. 11, Iss. 11. P. 69–79.
3. Development of models and methods for using heterogeneous gateways in 5G/IMT-2020 network infrastructure / Vlasenko L., Kulik V., Kirichek R., Koucheryavy A. *International Conference on Distributed Computer and Communication Networks.* 2019, September. Springer, Cham. P. 636–645.
4. A computation offloading algorithm based on game theory for vehicular edge networks / Liu Y., Wang S., Huang J., Yang F. *2018 IEEE International Conference on Communications (ICC).* 2018, May. P. 1–6.
5. A game-theoretic approach to computation offloading in satellite edge computing / Wang Y., Yang J., Guo X., Qu Z. *IEEE Access.* 2019. Vol. 8. P. 12510–12520.

ВИСНОВКИ

Сукупність наукових положень, сформульованих та обґрунтованих в дисертаційній роботі, складає вирішення науково-технічної задачі, яка полягала в необхідності підвищення ефективності обробки інформації підсистеми базових станцій оператора стільникового зв'язку.

У дисертаційній роботі отримані наступні теоретичні та практичні результати:

1. За аналізом якості обслуговування абонентів у реалізованих проектах мереж 5G в світі встановлено, що заявлені вимоги до мереж п’ятого покоління не досягаються в жодній із реалізованих мереж, а фактична якість обслуговування абонентів перебуває на не досить високому рівні, що свідчить про низьку ефективність існуючих методів керування мережами стільникового оператора та розподіленої обробки даних у них. Тому було проаналізовано ефективність методів, моделей та технологій розподіленої обробки даних у комп’ютерних системах операторів стільникового зв’язку та відповідно встановлені їх недоліки.

2. Вперше розроблено метод оптимізації розміщення масштабованих послуг на розподілених обчислювальних ресурсах мережі стільникового оператора, що полягає у послідовному використанні моделі граничних обчислень, узагальненої моделі мережі стільникового оператора та евристичного рішення, заснованого на використанні генетичних алгоритмів.

Даний метод дозволяє зменшити рівень деградація якості обслуговування кінцевих абонентів мережі стільникового оператора, зокрема, затримки на величину до 8 мс для великої кількості абонентів та відповідно великої кількості завдань.

3. Удосконалено метод динамічного розвантаження та планування задач для граничних комп’ютерних систем оператора стільникового зв’язку за рахунок формування задачі змішаного цілочисельного програмування та її вирішення за допомогою декомозиції Бендера.

Розроблений метод надає змогу максимізувати кількість допущених та відповідно виконаних завдань на розподілених граничних ресурсах мережі

стільникового оператора. При цьому швидкість виконання даного планування та розвантаження підвищилась до 10 разів.

4. Удосконалено метод керування випромінюваною потужністю мобільних пристроїв під час розвантаження завдань в розподіленій комп'ютерній системі граничних обчислень оператора стільникового зв'язку за рахунок послідовного використання моделі для оцінки умови необхідності розвантаження завдань в мобільній мережі та керування випромінюваною потужністю радіопередавальних пристроїв в каналах з інтерференцією на основі теорії ігор.

Даний метод дозволяє зменшити використання енергії під час використання граничних обчислень в комп'ютерних системах операторів стільникового зв'язку на величину від 5% до 40%.

5. Проведено імітаційне моделювання для оцінки ефективності розроблених у роботі моделей і методів, результати якого підтвердили їх адекватність та ефективність.

ДОДАТОК А
АКТИ ВПРОВАДЖЕННЯ РЕЗУЛЬТАТІВ ДИСЕРТАЦІЙНОГО ДОСЛІДЖЕННЯ

ЗАТВЕРДЖУЮ:
Проректор з наукової роботи
Центральноукраїнського національного
технічного університету
_____ О.М. Левченко
«21» _____вересня_____2020 р.

АКТ
реалізації результатів наукових досліджень
Усіка Павла Сергійовича

Комісія у складі голови – заступника завідуючого кафедрою «Кібербезпеки та програмного забезпечення» Центральноукраїнського національного технічного університету кандидата фізико-математичних наук, доцента Якименко Н.М., членів комісії – доцента кафедри «Кібербезпеки та програмного забезпечення» кандидата технічних наук, Мелешко Є.В., доцента кафедри «Кібербезпеки та програмного забезпечення» кандидата технічних наук, доцента Коваленко О.В. склала цей акт про те, що при розробці лекційних, практичних та лабораторних занять з навчальних дисциплін «Комп'ютерні мережі» та «Проектування й дослідження комп'ютерних мереж» у навчальному процесі Центральноукраїнського національного технічного університету були використані наступні результати наукових досліджень Усіка Павла Сергійовича:

1. Метод оптимізації розміщення масштабованих послуг на розподілених обчислювальних ресурсах мережі стільникового оператора.

Застосування результатів дисертаційних досліджень Усіка Павла Сергійовича дозволило підвищити рівень засвоєння навчального матеріалу з дисциплін «Комп'ютерні мережі» та «Проектування й дослідження комп'ютерних мереж» за рахунок більш поглибленого вивчення сучасних та перспективних методів передачі та перетворення інформації у телекомунікаційних мережах.

Голова комісії
Заступник завідуючого кафедри «Кібербезпеки та програмного забезпечення»
Центральноукраїнського національного
технічного університету
кандидат фізико-математичних наук, доцент Н.М. Якименко

Члени комісії:
доцент кафедри «Кібербезпеки та програмного забезпечення»
кандидат технічних наук, доцент Є.В. Мелешко

доцент кафедри «Кібербезпеки та програмного забезпечення»
кандидат технічних наук, доцент О.В. Коваленко

ТОВ «ІМПЕРІАЛ-НЕТ»
тел: +38 (0522) 27-60-06
факс : +38 (0522) 27-60-91
e-mail : office@imperial.net.ua
http://www.imperial.net.ua

ЗАТВЕРДЖУЮ:
Директор ТОВ «Імперіал-Нет».
_____
«14» _____ 202_ р.

АКТ

про впровадження результатів дисертаційного дослідження **Усіка Павла Сергійовича** на тему: *«Методи підвищення ефективності розподіленої обробки даних в комп'ютерних системах операторів стільникового зв'язку»*

Результати дисертаційного дослідження Усіка П.С. були впроваджені у виробничий процес ТОВ «Імперіал-Нет».

Для підвищення ефективності обробки даних в мережі Інтернет-провайдера «ІСП Імперіал» (ТОВ «Імперіал-Нет») було використано розроблену методику динамічного планування та розвантаження завдань в граничних комп'ютерних системах операторів зв'язку.

Запропоновані автором результати, надають можливість запропонувати принципову нову архітектуру розподілених обчислень для Інтернет-провайдера із використанням можливостей граничних обчислень (Edge Computing).

Експериментально досліджувалась ефективність використання розробленої методики на практиці. В результаті проведених досліджень було встановлено, що розроблена методика надала змогу:

– зменшити рівень затримки в мережі Інтернет-провайдера;

– проводити більш ефективне керування виконанням завдань в мережі Інтернет-провайдера із використанням граничних обчислень.

Результати проведеного дослідження дисертанта мають наукове та практичне значення для сучасних підприємств зв'язку, а тому можуть дістати широкого використання на практиці.

Голова комісії
Директор ТОВ «Імперіал-Нет» _____ Бур'янов К.В.

Члени комісії:
провідний розробник _____ Король К.В.

провідний розробник _____ Дуб О.М.

ТОВ «ІМПЕРІАЛ-НЕТ»
вул. Єгорова 8
м. Кіровоград,
Україна, 25006

ПАТ КБ «ПриватБанк»
Р/Р : 26000052913562
МФО : 323583
ЄДРПОУ : 39758019

тел : +38 (0522) 27-60-06
факс : +38 (0522) 27-60-91
office@imperial.net.ua
isp@imperial.net.ua
http://www.imperial.net.ua

ДОДАТОК Б
СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА ЗА ТЕМОЮ ДИСЕРТАЦІЇ ТА ВІДОМОСТІ ПРО АПРОБАЦІЮ РЕЗУЛЬТАТІВ ДИСЕРТАЦІЇ

Список публікацій здобувача за темою дисертації

1. Usik P., Smirnov O., Odarchenko R., Abakumova A., Kundyz M. QoE assesment technique for media delivery in 5g networks. *Problems of Infocommunications, Science and Technology (PIC S&T)*: 2019 IEEE Int. Sci.-Pract. Conf., (Kyiv, Oct. 8–11, 2019). P. 597–601 (**Scopus**).

2. Usik P., Odarchenko R., Volkov O., Simakhin V., Gospodarchuk O., Burmak Yu. 5G networks cyberincidents monitoring system for drone communications. *Actual Problems of Unmanned Aerial Vehicles Developments (APUAVD*): 2019 IEEE 5th Int. Conf., (Oct. 22–24, 2019). P. 165–169 (**Scopus**).

3. Ponomarenko O., Bulakovskaya A., Skripnichenko A., Usik P., Olenyuk A. Tomographic application-specific integrated circuits for fast radon transformation. *CEUR Workshop Proceedings.* 2020. No. 2654. P. 339–351 (**Scopus**).

4. Усік П.С., Смірнов О.А. Дослідження перспектив використання технологічних рішень в мережах 5g. *Кібербезпека та інформаційні технології*: монографія. Харків: ДІСА ПЛЮС, 2020. С. 122–135.

5. Котелянець В.В., Усік П.С., Кищенко В.В., Гнатюк В.О. Інтелектуалізована система моніторингу параметрів навколишнього середовища на базі технології інтернету речей. *Вісник інженерної академії України.* 2018. № 4. С. 133–140,

6. Усік П.С., Полігенько О.О., Одарченко Р.С., Терещенко Л.Ю., Смірнов О.А. «Інформаційна технологія та програмне забезпечення для підвищення ефективності планування підсистеми базових станцій стільникового зв'язку». *Проблеми телекомунікацій.* 2020 № 1(26). С. 83-96.

7. Усік П.С., Смірнов О.А., Миронець І.В., Буравченко К.О., Якименко Н.М. Метод підвищення ефективності розподіленої обробки даних у комп'ютерних

системах операторів стільникового зв'язку. *Вісник Черкаського державного технологічного університету.* 2020. № 4. С. 103–110.

8. Усік П.С., Полігенько О.О., Смірнов О.А. Напрямки підвищення ефективності управління підсистемою базових станцій стільникових операторів. *Проблеми розвитку глобальної системи зв'язку, навігації, спостереження та організації повітряного руху CNS/ATM*: тези доп. наук.-техн. конф., (м. Київ, 21–23 листоп. 2018 р.). Київ: НАУ, 2019. С. 32.

9. Одарченко Р.С., Мараткызы К., Усик П.С. Анализ перспектив использования сетей 5g для автоматизации производственных процессов. *Өндірістегі цифрлық технологиялар конференциясы*: Республикалық ғылыми және практикалық конференциясының жинағы=*Цифровые технологии в промышленности*: материалы респ. науч.-практ. конф.=*Digital technologies in industry*: Materials of sci. and pract. conf. Казахстан, Актау: КГУТИ им. Ш. Есенова, 2019. Каз., рус., англ. С. 42–44.

10. Усік П.С., Смірнов О.А., Якименко Н.М. Перспективи використання мережевих технологічних рішень в 5g. *Інформаційна безпека та інформаційні технології (Information Security and Information Technologies)*: ІІ Міжнар. наук.-практ. конф., (м. Кропивницький, 2–3 квіт. 2020 р.). С. 56.

11. Усік П.С., Смірнов О.А. Підвищення ефективності функціонування підсистеми базових станцій на основі Multi-Access Edge Computing. *Інформаційні технології – 2020 (IT-2020)*: VII Всеукр. наук.-практ. конф. молодих науковців, (м. Київ, 21 трав. 2020 р.). С. 135–136.

12. Chumachenko B.S., Zaitseva N.O., Grigorenko D.K., Usik P.S. Research of the advantages and disadvantages of the network virtualization of network resources of a consistent architecture of 5g networks. *POLIT. Challenges of science today*, (Apr. 1–3, 2020). P. 99–100.

Відомості про апробацію результатів дисертації

1. Основні теоретичні та практичні результати дисертаційної роботи доповідались і обговорювались на таких конференціях і семінарах:

2. Міжнародна науково-практична конференція «Безпека інформації в інформаційно-телекомунікаційних системах» (Київ, 2018-2020 рр.);

3. VIII Міжнародна науково-технічна конференція "Комп’ютерні системи і мережні технології" (Київ, НАУ, 2019 р.);

4. Міжнародна науково-технічна конференція "ITSEC" (Київ, НАУ, 2019 р.);

5. Міжнародна науково-практична конференція молодих учених і студентів "Політ. Сучасні проблеми науки" (Київ, НАУ, 2019 р., 2020 р.);

6. Всеукраїнська науково-практична конференція «Перспективні напрями захисту інформації» (Одеса, 2019, 2020 р.р.);

7. Автоматика та комп’ютерно-інтегровані технології у промисловості, телекомунікаціях, енергетиці та транспорті: всеукраїнська науково-практична інтернет-конференція (Кропивницький, 2019, 2020 рр.),

8. Перший Міжнародний семінар з кібергігієни і управління конфліктами в глобальних інформаційних мережах (Київ, НАУ, 2019 р.),

9. IEEE International Scientific-Practical Conference «Problems of Infocommunications Science and Technology (PIC S&T)» (Харків, ХНУРЕ, 2019 р.).