

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Чорноморський національний університет імені Петра Могили

Факультет комп'ютерних наук

Кафедра комп'ютерної інженерії

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувачка кафедри,

д-р техн. наук, проф.

_____ Ірина ЖУРАВСЬКА

« __ » _____ 202__ р.

КВАЛІФІКАЦІЙНА РОБОТА

НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВРА

Розподілена система обміну DICOM-зображень

з безвтратним стисканням

з використанням одноплатних комп'ютерів

Спеціальність 123 Комп'ютерна інженерія

Освітня програма «Комп'ютерна інженерія»

Здобувач

_____ Віктор ШЕВЧЕНКО

підпис

« __ » _____ 20__ р.

Керівник PhD, доцент,
каф. комп'ютерної інженерії

_____ Євген ДАРНАПУК

підпис

« __ » _____ 20__ р.

Миколаїв – 2026

Факультет	Комп'ютерних наук
Кафедра	Комп'ютерної інженерії
Рівень вищої освіти	Перший (бакалаврський)
Освітній ступень	Бакалавр
Спеціальність	123 Комп'ютерна інженерія
Освітня програма	Комп'ютерна інженерія

ЗАТВЕРДЖУЮ
Завідувач кафедри комп'ютерної інженерії
_____ Ірина ЖУРАВСЬКА
« _____ » _____ 202__ р.

ЗАВДАННЯ
на кваліфікаційну роботу здобувача

Шевченка Віктора Вікторовича

(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи

Розподілена система обміну DICOM-зображень з безвтратним стисканням
з використанням одноплатних комп'ютерів

Затверджена наказом по ЧНУ ім. Петра Могили від 25.11.2025 № 294.

2. Строк представлення кваліфікаційної роботи « _____ » _____ 20__ р.

3. Очікуваний результат роботи та початкові дані, якщо такі потрібні

Очікуваним результатом роботи є прототип розподіленої системи обміну
DICOM-зображеннями з безвтратним стисканням на базі одноплатних
комп'ютерів. Початковими даними є тестові DICOM-зображення, Raspberry Pi 4,
PACS-сервер Orthanc, комп'ютер для розгортання серверної частини та програмні
засоби для приймання, стискання й передавання медичних даних.

4. Перелік питань, що підлягають розробці:

1) проаналізувати теоретичні основи організації обміну медичними
зображеннями у форматі DICOM та визначити роль стандартів PACS / HL7 у
медичній інформаційній інфраструктурі;

2) розглянути сучасні методи безвратного стискання медичних зображень, існуючі системи обміну медичними даними та апаратні платформи для побудови вузлів PACS-систем;

3) сформувані вимоги до апаратно-програмного забезпечення розподіленої системи обміну DICOM-зображеннями;

4) спроектувати архітектуру системи з урахуванням взаємодії периферійного вузла на базі Raspberry Pi 4 та центрального PACS-сервера Orthanc;

5) реалізувати основні програмні компоненти системи, зокрема модуль приймання DICOM-даних, модуль безвратного стискання та механізм передавання зображень до PACS-сервера;

6) провести експериментальну перевірку працездатності системи та оцінити ефективність запропонованого рішення за показниками стискання, часу оброблення і передавання DICOM-зображень.

5. Перелік графічних матеріалів

Презентація

6. Консультанти:

Консультант	Кафедра (організація)	Частина роботи

Керівник роботи

Особистий підпис

Євген ДАРНАПУК

Власне ім'я ПРИЗВИЩЕ

Здобувач

Особистий підпис

Віктор ШЕВЧЕНКО

Власне ім'я ПРИЗВИЩЕ

Дата видачі завдання « 03 » грудня 2025 р.

КАЛЕНДАРНИЙ ПЛАН
виконання кваліфікаційної бакалаврської роботи

Тема: Розподілена система обміну DICOM-зображень з безвтратним стисканням з використанням одноплатних комп'ютерів

№	Найменування роботи	Початок	Закінчення	Примітки
1	Розробка та затвердження завдання на виконання КБР	28.11.2025	03.12.2025	Виконано
2	Складання календарного плану КБР	04.12.2025	15.12.2025	Виконано
3	Огляд літератури та сучасних рішень	16.12.2025	28.12.2025	Виконано
4	Аналіз методів безвтратного стискання DICOM-зображень та вибір підходу для реалізації в системі	10.01.2026	29.01.2026	Виконано
5	Проектування архітектури та вибір апаратно-програмних компонентів	29.01.2026	10.03.2026	Виконано
6	Розробка та налаштування прототипу системи з використанням Raspberry Pi 4	10.03.2026	25.04.2026	Виконано
7	Збір та аналіз матеріалів до КБР	25.04.2026	05.05.2026	Виконано
8	Оформлення КБР та презентації	27.04.2026	19.05.2026	Виконано
9	Перший передній захист КБР	22.05.2026	22.05.2026	Виконано
10	Другий передній захист КБР	05.06.2026	05.06.2026	Виконано
11	Завершення оформлення КБР та презентації	06.06.2026	08.06.2026	Виконано
12	Перевірка на академічний плагіат, фальсифікацію та списування	09.06.2026	10.06.2026	Виконано
13	Відгук керівника КБР	09.06.2026	10.06.2026	Виконано
14	Рецензування КБР	10.06.2026	13.06.2026	Виконано
15	Подання КБР, її електронної копії та інших документів (відгуку, рецензії)	15.06.2026	17.06.2026	Виконано
16	Захист кваліфікаційної бакалаврської роботи	24.06.2026	24.06.2026	Виконано

Керівник роботи

Особистий підпис

Сьген ДАРНАПУК

Власне ім'я ПРИЗВИЩЕ

Здобувач

Особистий підпис

Віктор ШЕВЧЕНКО

Власне ім'я ПРИЗВИЩЕ

« 15 » _____ грудня _____ 2025 р.

АНОТАЦІЯ

до кваліфікаційної бакалаврської роботи
«Розподілена система обміну DICOM-зображень з безвтратним стискуванням
з використанням одноплатних комп'ютерів»

Здобувач: Шевченко Віктор Вікторович

Керівник: PhD, доцент Дарнапук Євген Сергійович

Кваліфікаційна бакалаврська робота присвячена дослідженню теоретичних і практичних засад побудови розподіленої системи обміну DICOM-зображеннями з безвтратним стискуванням на базі доступних одноплатних комп'ютерів. Актуальність роботи зумовлена зростанням обсягів медичних візуальних даних, необхідністю їх оперативного передавання між вузлами медичної інформаційної інфраструктури, а також потребою у використанні компактних і економічно доцільних апаратних засобів для організації периферійного рівня обробки. У роботі розглянуто підхід, за якого частина операцій з приймання, підготовки та передавання DICOM-даних виконується на периферійному вузлі, побудованому на Raspberry Pi 4, а центральне архівування та серверна взаємодія забезпечуються PACS-сервером Orthanc.

Об'єктом дослідження є процес обміну медичними зображеннями у форматі DICOM у розподілених інформаційних системах. Предметом дослідження є методи безвтратного стискування медичних зображень, а також архітектурні й програмно-апаратні засоби побудови системи розподіленого обміну на основі одноплатних комп'ютерів. Метою роботи є розробка прототипу розподіленої системи обміну DICOM-зображеннями з безвтратним стискуванням для підвищення ефективності передавання та оброблення медичних даних в умовах обмежених апаратних ресурсів. Практичне значення одержаних результатів полягає у можливості використання запропонованого рішення для створення доступного стенду або прототипу системи обміну медичними зображеннями з підтримкою PACS-сумісної взаємодії, локального безвтратного стискування та мережевого передавання даних.

У першому розділі роботи здійснено аналіз теоретичних основ організації обміну медичними зображеннями, розглянуто особливості стандарту DICOM, роль PACS і HL7 у медичній IT-інфраструктурі, досліджено методи втратного та безвтратного стискування. У другому розділі розроблено архітектуру системи, обґрунтовано вибір Raspberry Pi 4 як периферійного вузла, описано використання алгоритму безвтратного стискування СВРС / СВРС2, подано логіку інтеграції зі серверною частиною на базі Orthanc, а також розглянуто програмну реалізацію клієнтського та серверного модулів обміну даними. У третьому розділі наведено результати експериментальної перевірки працездатності системи, описано стенд, використані тестові DICOM-набори, результати передавання даних, порівняльний аналіз алгоритмів СВРС, СВРС2 та ZIP, а також оцінено продуктивність Raspberry Pi 4 у ролі периферійного вузла в розподіленому середовищі. У висновках наведено загальне узагальнення результатів виконаної роботи, оцінку реалізованого прототипу та ступінь виконання поставлених завдань.

У цілому, кваліфікаційна бакалаврська робота, містить 64 с. (без додатків), 18 рис., 10 табл., 20 джерел посилання та 3 додатки.

Ключові слова: DICOM, PACS, Orthanc, Raspberry Pi 4, безвтратне стискування, СВРС, СВРС2, REST API, медичні зображення, розподілена система.

ABSTRACT

of the Bachelor's Thesis

“Distributed DICOM image exchange system with lossless compression
using single-board computers”

Applicant: Viktor Shevchenko

Supervisor: PhD, Associate Professor Yevhen Darnapuk

The bachelor's thesis is devoted to the study of theoretical and practical principles of building a distributed DICOM image exchange system with lossless compression based on affordable single-board computers. The relevance of the work is determined by the growing volume of medical imaging data, the need for their prompt transmission between nodes of medical information infrastructure, as well as the demand for compact and cost-effective hardware solutions for organizing the peripheral level of data processing. The thesis considers an approach in which part of the operations related to receiving, preparing, and transmitting DICOM data is performed on a peripheral node built on Raspberry Pi 4, while central archiving and server interaction are provided by the Orthanc PACS server.

The object of the study is the process of exchanging medical images in DICOM format within distributed information systems. The subject of the study is the methods of lossless compression of medical images, as well as architectural, software, and hardware means of building a distributed exchange system based on single-board computers. The purpose of the thesis is to develop a prototype of a distributed DICOM image exchange system with lossless compression in order to improve the efficiency of transmitting and processing medical data under limited hardware resources. The practical significance of the obtained results lies in the possibility of using the proposed solution to create an affordable stand or prototype of a medical image exchange system with PACS-compatible interaction, local lossless compression, and network-based data transmission.

The first chapter analyzes the theoretical foundations of medical image exchange, considers the features of the DICOM standard, the role of PACS and HL7 in medical IT infrastructure, and investigates lossy and lossless compression methods. The second chapter presents the system architecture, substantiates the choice of Raspberry Pi 4 as a peripheral node, describes the use of the CBPC / CBPC2 lossless compression algorithm, explains the logic of integration with the server side based on Orthanc, and discusses the software implementation of the client and server data exchange modules. The third chapter contains the results of the experimental verification of the system operability, describes the test stand, the DICOM datasets used, the results of data transmission, the comparative analysis of CBPC, CBPC2 and ZIP algorithms, as well as the performance evaluation of Raspberry Pi 4 as a peripheral node in a distributed environment. The conclusions provide a general summary of the completed work, an assessment of the implemented prototype, and the degree of fulfillment of the assigned tasks.

The paper consists of 64 pages (excluding appendices), 18 figures, 10 tables, 20 references, and 3 appendices.

Keywords: *DICOM, PACS, Orthanc, Raspberry Pi 4, lossless compression, CBPC, CBPC2, REST API, medical images, distributed system.*

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	4
ВСТУП.....	5
1 АНАЛІЗ ТЕОРЕТИЧНИХ ОСНОВ ТА ІСНУЮЧИХ РІШЕНЬ.....	8
1.1 Актуальність розробки системи розподіленого обміну DICOM-зображеннями	8
1.2 Особливості стандарту DICOM та протоколів PACS / HL7.....	13
1.3 Методи стискання медичних зображень: втратні та безвтратні алгоритми.....	16
1.4 Огляд сучасних систем і сервісів обміну медичними даними (Orthanc, MIDOM, DCMTK, OsiriX).....	19
1.5 Аналіз апаратних платформ для побудови вузлів PACS-систем (Raspberry Pi, Jetson Nano, NUC).....	22
1.6 Формування вимог до апаратно-програмного забезпечення	26
Висновки до розділу 1	27
2 ПРОЄКТУВАННЯ РОЗПОДІЛЕНОЇ СИСТЕМИ ОБМІНУ DICOM-ДАНИМИ	29
2.1 Загальна архітектура системи та принципи функціонування	29
2.2 Обґрунтування вибору апаратного забезпечення.....	32
2.3 Використання алгоритму безвтратного стискання СВРС / СВРС2.....	35
2.4 Інтеграція алгоритму стискання у серверну частину PACS-системи	39
2.5 Програмна реалізація серверного та клієнтського модулів.....	42
2.6 Схема мережевої взаємодії та безпечного передавання даних між вузлами	46
Висновки до розділу 2	49
3 ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ.....	51
3.1 Методика тестування та опис експериментального стенду	51
3.2 Набори даних та параметри випробувань (СТ, MRI, X-ray).....	53

3.3 Оцінка ефективності стискання: коефіцієнт, якість, час передавання	54
3.4 Порівняльний аналіз алгоритмів СВРС, СВРС2 та ZIP	56
3.5 Аналіз продуктивності Raspberry Pi 4 у розподіленому середовищі.....	58
Висновки до розділу 3	60
ВИСНОВКИ.....	61
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	63
ДОДАТОК А Блок-схема функціонування системи розподіленого обміну DICOM- даними	65
ДОДАТОК Б Код програми передавання DICOM-файлу на PACS-сервер	66
ДОДАТОК В Матеріали апробації роботи.....	67

ПЕРЕЛІК СКОРОЧЕНЬ

АПЗ	– апаратно-програмна забезпечення
КБР	– кваліфікаційна бакалаврська робота
API	– Application Programming Interface
CBPC	– Classification and Blending Predictor Coder
CT	– Computed Tomography
CV	– Computer Vision
DCMTK	– DICOM Toolkit
DICOM	– Digital Imaging and Communications in Medicine
DOI	– Digital Object Identifier
HL7	– Health Level Seven
HTTP	– HyperText Transfer Protocol
JPEG	– Joint Photographic Experts Group
JPEG2000	– Image Compression Standard
JPEG-LS	– Lossless and Near-Lossless Image Compression Standard
MIDOM	– Medical Image Communication System
MRI	– Magnetic Resonance Imaging
NUC	– Next Unit of Computing
PACS	– Picture Archiving and Communication System
PNG	– Portable Network Graphics
REST	– Representational State Transfer
TLS	– Transport Layer Security

ВСТУП

Сучасний розвиток медичних інформаційних технологій пов'язаний із широким використанням цифрових засобів діагностики, зберігання та оброблення візуальних даних. Одним із базових стандартів у цій сфері є DICOM, який забезпечує уніфіковане подання медичних зображень, метаданих дослідження та їх передавання між різними апаратно-програмними системами. На його основі побудовані PACS-системи, що використовуються для архівування, пошуку, передавання та перегляду медичних зображень у клінічній практиці.

У сучасних лікувально-діагностичних установах формується значний обсяг цифрових медичних даних, особливо під час комп'ютерної томографії, магнітно-резонансної томографії та цифрової рентгенографії. Зростання обсягів DICOM-даних підвищує навантаження на канали зв'язку, сервери архівування та засоби оброблення інформації. Через це задача ефективного обміну медичними зображеннями має не лише технічне, а й практичне значення.

Централізований підхід до зберігання та передавання DICOM-зображень не завжди забезпечує достатню гнучкість і продуктивність, особливо в умовах розподіленої інфраструктури або обмежених апаратних ресурсів. Тому доцільним є використання розподілених систем, у яких частина функцій оброблення даних переноситься на периферійні вузли мережі. Це дає змогу зменшити навантаження на центральний сервер, оптимізувати мережеві ресурси та підвищити масштабованість системи. Як базову платформу периферійного вузла в роботі розглянуто Raspberry Pi 4.

Однією з ключових проблем під час передавання медичних зображень є великий обсяг файлів, що зумовлює потребу у стисканні. Водночас для медичних даних будь-яка втрата діагностично важливої інформації є небажаною, тому особливого значення набувають безвтратні алгоритми. У роботі розглядається підхід СВРС2, який базується на класифікації локального контексту пікселів, прогнозуванні їх значень і подальшому кодуванні похибки, що робить його придатним для стискання медичних зображень без втрати якості.

Актуальність теми зумовлена необхідністю створення технічно ефективних засобів обміну DICOM-зображеннями, придатних для використання в умовах обмежених обчислювальних ресурсів. Ця задача є важливою для невеликих медичних установ, віддалених діагностичних підрозділів, навчально-дослідних лабораторій та інших об'єктів, де використання повноцінної серверної інфраструктури може бути надмірно витратним. Запропонований підхід, який передбачає поєднання розподіленої архітектури, одноплатних комп'ютерів, PACS-сумісного обміну та безвратного стискання, має практичну цінність, яка повинна мати завершений характер отриманих результатів.

Метою роботи є розробка прототипу розподіленої системи обміну DICOM-зображеннями з безвратним стисканням з використанням одноплатних комп'ютерів для підвищення ефективності передавання та оброблення медичних даних.

Для досягнення поставленої мети в роботі необхідно розв'язати такі **завдання**:

- проаналізувати теоретичні основи організації обміну медичними зображеннями у форматі DICOM та визначити роль стандартів PACS / HL7 у медичній інформаційній інфраструктурі;

- розглянути сучасні методи безвратного стискання медичних зображень, існуючі системи обміну медичними даними та апаратні платформи для побудови вузлів PACS-систем;

- спроектувати архітектуру розподіленої системи обміну DICOM-зображеннями з урахуванням взаємодії периферійних вузлів і центрального PACS-сервера;

- визначити програмні та апаратні компоненти системи, необхідні для реалізації приймання, стискання, передавання та зберігання DICOM-даних;

- реалізувати основні функціональні модулі системи, зокрема серверну частину, взаємодію з PACS-сервером та механізм безвратного стискання медичних зображень;

– провести експериментальне дослідження ефективності запропонованого рішення та оцінити результати за показниками стискання, часу оброблення і передавання DICOM-зображень.

Об'єктом дослідження є процес обміну медичними зображеннями у форматі DICOM у розподілених інформаційних системах.

Предметом дослідження є методи безвтратного стискання медичних зображень та архітектурні засоби побудови розподіленої системи обміну даними на основі одноплатних комп'ютерів.

Практичне значення одержаних результатів полягає у можливості створення прототипу системи, придатної для організації безпечного, масштабованого та економічно доцільного обміну DICOM-зображеннями з використанням доступної апаратної платформи та безвтратного стискання даних.

У першому розділі роботи здійснюється аналіз теоретичних основ і наявних рішень у сфері обміну DICOM-зображеннями, у другому розділі розглядаються питання проєктування розподіленої системи та реалізації її основних компонентів, а в третьому розділі подано результати та їх порівняльний аналіз.

Апробація результатів кваліфікаційної роботи відбулася під час XXVIII Всеукраїнської щорічної науково-практичної конференції «Могилянські читання – 2025» (м. Миколаїв, 10–14 листопада 2025 р.) [1]; XXIII Міжнародна наукова конференція «Ольвійський форум – 2026: стратегії країн Причорноморського регіону в геополітичному просторі» (м. Миколаїв, 29 червня – 4 липня 2026 р.) [2]

1 АНАЛІЗ ТЕОРЕТИЧНИХ ОСНОВ ТА ІСНУЮЧИХ РІШЕНЬ

1.1 Актуальність розробки системи розподіленого обміну DICOM-зображеннями

Медичні зображення давно перестали бути лише допоміжним матеріалом до висновку лікаря. У багатьох випадках саме вони є основою для аналізу стану пацієнта, уточнення характеру патологічних змін і подальшого спостереження за перебігом хвороби. Комп'ютерна томографія, магнітно-резонансна томографія, рентгенографія, мамографія, ультразвукові та інші методи візуалізації формують значний обсяг клінічної інформації, яка використовується неодноразово. Один і той самий набір зображень може переглядатися під час первинного опису, повторного аналізу, консультації іншого спеціаліста, порівняння з попередніми дослідженнями або підготовки до втручання. Через це важливим стає не тільки отримання знімка як такого, а й увесь цикл його подальшого існування в медичному середовищі: зберігання, пошук, передавання, відкриття на іншому робочому місці, архівування та відновлення.

У цифровій медицині цей цикл майже неможливо розглядати окремо від стандарту DICOM [3]. Саме він забезпечує ту форму організації даних, у якій зображення зберігається разом із супровідною інформацією [4]. Для клінічної практики це означає, що файл містить не лише сам візуальний результат, а й описовий контекст: серію, параметри дослідження, службові атрибути, відомості про пацієнта та обладнання. У результаті знімок існує не ізольовано, а як частина структурованого інформаційного середовища. Така побудова є логічною і зручною для медицини, однак саме вона створює додаткове навантаження в ситуаціях, коли досліджень стає багато, а їх потрібно постійно переміщати між різними вузлами системи.

У межах одного апарата або одного дослідження ця проблема не завжди виглядає критичною. Вона стає відчутною тоді, коли накопичення даних відбувається щоденно і безперервно. Особливо це стосується томографічних серій, де один сеанс може включати десятки й сотні зрізів. Якщо ж мова йде про роботу

цілого відділення, діагностичного центру або лікарні, потік DICOM-файлів швидко перетворюється на великий масив інформації, який потрібно оперативно приймати, систематизувати, передавати та зберігати. Труднощі в такій ситуації пов'язані вже не з окремим файлом і не з окремим дослідженням. Вони пов'язані з тим, як вся система справляється з постійним рухом великих обсягів медичних даних.

Централізована модель архівування і доступу до зображень залишається найбільш зрозумілою з організаційної точки зору. Центральний сервер приймає дослідження, формує архів, обробляє запити, видає серії зображень користувачам і забезпечує загальну логіку зберігання. Подібний підхід добре узгоджується з класичним PACS-середовищем, але в умовах зростання кількості досліджень він починає дедалі сильніше залежати від навантаження. Один вузол фактично стає місцем концентрації майже всіх основних операцій. Це впливає на швидкість пошуку, на час відкриття серій, на обсяг трафіку в мережі та на саму стійкість системи до розширення. Навіть за наявності стабільного серверного обладнання така архітектура не завжди однаково добре працює в різних умовах експлуатації.

Ситуація ще більше ускладнюється тоді, коли обмін медичними зображеннями відбувається між кількома фізично віддаленими точками. Це можуть бути різні корпуси лікарні, окремі діагностичні кабінети, дистанційні робочі місця лікарів, консультативні підрозділи або телемедичні вузли. У подібних сценаріях питання обсягу даних стає особливо помітним. Передавання повних серій DICOM-зображень у первинному вигляді потребує значних мережевих ресурсів. Якщо така передача виконується регулярно, мережа починає працювати в режимі постійного навантаження. У результаті швидкість доступу до архіву, час пересилання дослідження та загальна чутливість системи до перевантаження залежать не тільки від серверної частини, а і від самої схеми організації обміну. Її зміст полягає не в тому, щоб повністю відмовитися від центрального архіву, а в тому, щоб інакше розподілити оброблення даних між вузлами системи. Частина функцій переноситься ближче до місця формування зображень. Локальний вузол може приймати серію DICOM-файлів, виконувати попередню підготовку, тимчасове

збереження, стискання, а вже потім передавати дані до центрального PACS-рівня. Така організація змінює характер навантаження. Центральний сервер більше не отримує весь потік даних у початковому вигляді, а периферійний рівень перестає бути лише пасивною точкою передавання. Основні чинники, які формують практичну потребу в такій архітектурі, подано на рис. 1.1.

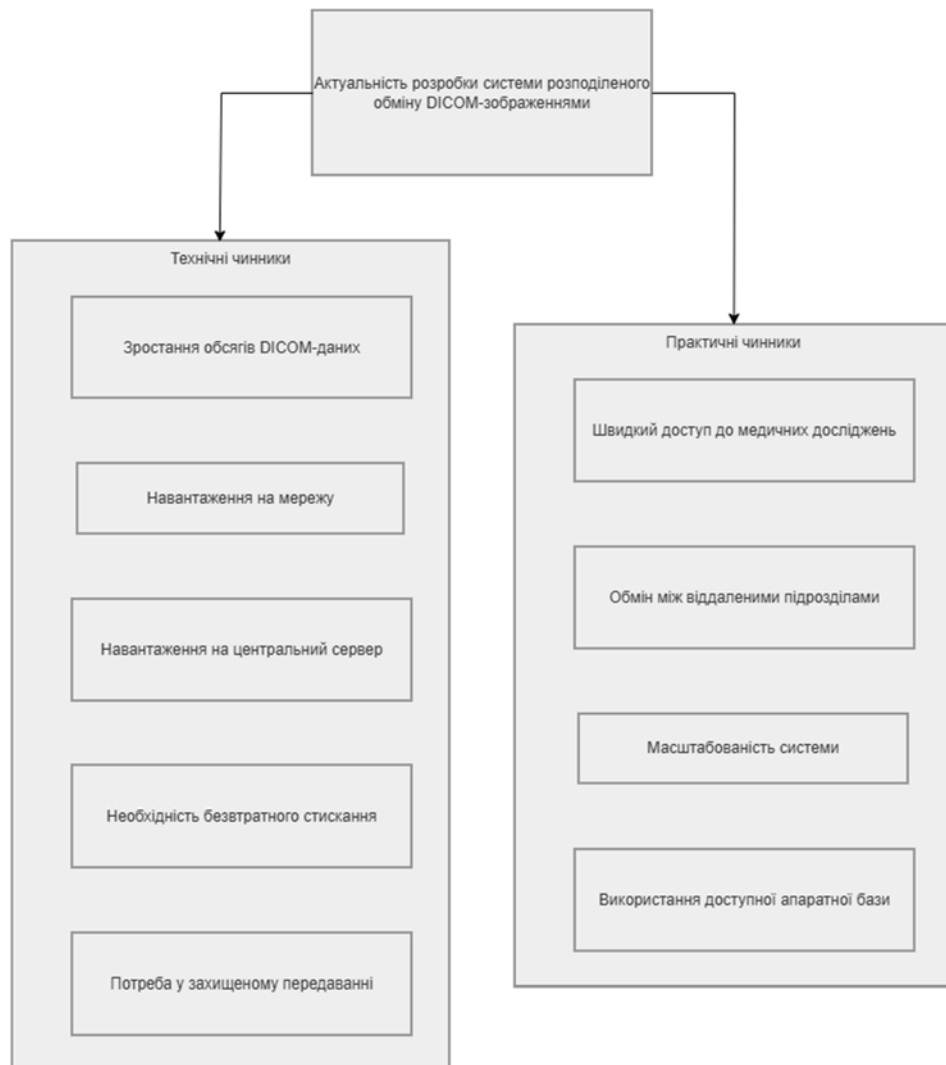


Рисунок 1.1 – Основні чинники актуальності розробки системи розподіленого обміну DICOM-зображеннями

Окрему увагу в цьому контексті привертає апаратна база периферійних вузлів. Якщо розглядати розподілену систему не як абстрактну модель, а як реальну інфраструктуру, постає цілком прикладне питання: на якому обладнанні повинні працювати локальні точки приймання та попереднього оброблення медичних даних. Розгортання повноцінного сервера в кожній такій точці виглядало б

надмірним рішенням, особливо для невеликих медичних установ, лабораторій або локальних діагностичних відділень. Саме тут значення набувають одноплатні комп'ютери. Їх використання дозволяє організувати компактний периферійний вузол без складної інфраструктури, з помірним енергоспоживанням і достатнім набором інтерфейсів для мережевої взаємодії та локального оброблення даних. У межах теми як така платформа розглядається Raspberry Pi 4, що дозволяє поєднати доступність апаратної бази з реальними функціями вузла системи обміну DICOM-зображеннями [5].

Не менш важливим є і стискання медичних зображень. Воно безпосередньо пов'язане зі швидкістю передавання даних, обсягом архіву та навантаженням на мережу. У звичайних мультимедійних системах ця проблема часто розв'язується за рахунок втратного зменшення розміру файлу. Для клінічно значущих зображень такий підхід має очевидні межі. Медичний знімок використовується не для загального перегляду, а для аналізу, де значення можуть мати малопомітні структурні відмінності, дрібні контрастні зміни або локальні ділянки, які не повинні зникати після оброблення. Саме тому найбільший інтерес викликають безвтратні методи стискання. Їх застосування дозволяє зменшити обсяг даних без зміни вихідного зображення, а отже без ризику спотворити діагностично важливу інформацію. У межах цієї теми такий підхід є одним із ключових, оскільки розподілена система обміну повинна не просто пришвидшувати передачу даних, а робити це без втрати медичної цінності зображень [6].

До цього додається ще одна група вимог, пов'язаних уже не з обсягом файлів, а з характером самої інформації. DICOM-зображення супроводжуються відомостями, що безпосередньо стосуються пацієнта та дослідження. Усі операції з такими даними вимагають захищеного передавання, контролю доступу та збереження цілісності. У розподіленій архітектурі ця умова набуває особливої ваги, оскільки інформація проходить через кілька точок взаємодії. Тому під час розробки системи одночасно враховуються питання мережевої організації, стискання, архівування і безпеки каналів обміну. У поданому матеріалі це відображено через

використання TLS, REST API та центрального PACS Orthanc як частини загальної концепції системи [7; 8].

У повсякденній роботі медичної системи ці труднощі проявляються досить конкретно: у швидкості передавання серій зображень, у навантаженні на центральний вузол, у вимогах до точності відтворення та в захисті даних. Основні з них зведено в табл. 1.1.

Таблиця 1.1 – Основні проблеми централізованого обміну DICOM-зображеннями та шляхи їх вирішення

Проблема	Прояв у системі	Підхід до вирішення
Великі обсяги DICOM-даних	Збільшення часу передавання і навантаження на мережу	Локальне безвтратне стискання на периферійному вузлі
Концентрація операцій у центральному сервері	Сповільнення доступу до архіву та оброблення запитів	Розподіл функцій між кількома вузлами
Обмеженість локальної інфраструктури	Ускладнення розгортання в невеликих установах	Використання компактних і доступних платформ
Вимога точного відтворення зображень	Неможливість втратного стискання в клінічно важливих сценаріях	Використання безвтратних алгоритмів
Чутливість медичних даних	Ризик порушення конфіденційності та цілісності	Захищене передавання і контроль доступу

Для системи такого типу важливо, що всі ці обмеження виникають не ізольовано. Великий обсяг даних майже завжди супроводжується навантаженням на мережу, а потреба швидко передати дослідження поєднується з вимогою не змінити його діагностичний зміст. До цього додаються питання вартості локальних вузлів і безпечного передавання медичної інформації. У такому поєднанні і проявляється практична потреба в системі розподіленого обміну DICOM-зображеннями з безвтратним стисканням, де кожен компонент вирішує не ізольовану задачу, а працює як частина єдиної архітектури.

1.2 Особливості стандарту DICOM та протоколів PACS / HL7

Робота з медичними зображеннями неможлива без узгоджених правил їх зберігання, передавання та включення в загальну інформаційну систему закладу. Коли в одній інфраструктурі взаємодіють томографи, рентгенівські апарати, архівні сервери, робочі станції лікарів і зовнішні сервіси, потрібна спільна технологічна основа. Саме таку основу формують DICOM, PACS і HL7. Вони належать до різних рівнів медичного IT-середовища, але працюють не відокремлено, а в постійному зв'язку. DICOM визначає правила роботи із зображеннями, PACS забезпечує їх накопичення й доступ до архіву, а HL7 відповідає за електронний обмін пов'язаною клінічною інформацією.

DICOM у медичній практиці сприймається не просто як формат файлу [3; 4]. Його роль значно ширша. На офіційному ресурсі DICOM підкреслено, що це міжнародний стандарт для медичних зображень і пов'язаної з ними інформації, який визначає формати даних, достатні для клінічного використання. Окремо важливо, що DICOM групує інформацію в набори даних, тому зображення не відокремлюється від ідентифікаційних та описових атрибутів. Для системи обміну це означає цілісність: серія знімків передається не як набір звичайних графічних файлів, а як структурований медичний об'єкт із власним контекстом.

На практиці це має дуже конкретні наслідки. Під час передавання або архівування важливим залишається не тільки саме зображення, а й уся службова інформація, що супроводжує дослідження. Якщо ця структура порушується, ускладнюється пошук, маршрутизація, повторний перегляд і пов'язування серії з іншими записами медичної системи. Через це DICOM потрібно розглядати як базовий рівень сумісності, без якого обмін діагностичними даними перетворюється на набір несумісних локальних рішень. У випадку розподіленої системи це особливо важливо, оскільки передавання між вузлами не повинно змінювати логіку організації медичних даних.

Зв'язок між DICOM, PACS і HL7 у медичній інформаційній інфраструктурі подано на рис. 1.2.



Рисунок 1.2 – Узагальнена схема взаємодії DICOM, PACS та HL7
у медичній інформаційній системі

PACS-рівень у такій схемі виконує іншу функцію. Якщо DICOM задає структуру медичних зображень, то PACS відповідає за те, де і як ці дані накопичуються, упорядковуються та стають доступними для користувача. У межах центрального архіву зберігаються серії досліджень, здійснюється пошук, організовується доступ для перегляду та передавання. Для теми розподіленого обміну це принципово, оскільки периферійні вузли можуть виконувати локальне стискання чи попередню підготовку, але завершальна точка маршруту все одно потребує узгодженого серверного середовища. У матеріалах за темою таку роль виконує PACS Orthanc, який використовується як центральне сховище DICOM-даних і підтримує REST API для інтеграції із зовнішніми модулями. Офіційний сайт Orthanc визначає його як вільний і відкритий lightweight DICOM server, здатний приймати, завантажувати й передавати DICOM-зображення та автоматизувати imaging flows через API [7; 8].

HL7 вже належить до іншого інформаційного рівня. Він не описує самі медичні зображення в тому вигляді, як це робить DICOM. Його завдання полягає в узгодженому електронному обміні даними між медичними системами. На ресурсах HL7 FHIR цей стандарт подано як засіб електронного обміну медичною інформацією [9]. У реальному середовищі через HL7 можуть передаватися

відомості про пацієнта, направлення, результати, маршрутизація подій і зв'язок між різними підсистемами закладу. Тому DICOM і HL7 не дублюють одне одного. Один рівень відповідає за зображення та їхній внутрішній контекст, інший – за включення цього контексту в ширший клінічний інформаційний потік.

У повсякденній роботі це виглядає доволі просто: лікар відкриває дослідження не як окремий файл, а як частину запису про конкретного пацієнта, пов'язаного з направленням, датою виконання, описом і подальшими діями. Саме тому для системи розподіленого обміну DICOM-зображеннями недостатньо лише підтримувати правильне передавання графічних даних. Потрібно, щоб ці дані без проблем інтегрувалися в уже наявну медичну IT-інфраструктуру, де працюють архіви, реєстри, обмінні сервіси та інші прикладні модулі. У цьому й полягає практичний сенс одночасної сумісності з DICOM, PACS і HL7.

Розподіл ролей між цими технологіями зручніше подати в узагальненому вигляді. Таке зіставлення подано в табл. 1.2.

Таблиця 1.2 – Порівняльна характеристика DICOM, PACS та HL7

Компонент	Основне призначення	Тип даних / функцій	Роль у системі
DICOM	Подання, зберігання, передавання й оброблення медичних зображень	Зображення, серії, метадані дослідження	Базовий стандарт сумісності для медичних зображень
PACS	Архівування, пошук, доступ і маршрутизація зображень	Сховище DICOM-даних, архів, серверний доступ	Централізований рівень зберігання та доступу
HL7	Електронний обмін клінічною та адміністративною інформацією	Дані про пацієнта, направлення, результати, події	Інтеграція медичних підсистем і загального інформаційного потоку

З табл. 1.2 видно, що DICOM, PACS і HL7 вирішують різні задачі, але в реальній системі працюють спільно. DICOM без PACS не забезпечує повноцінного архівного середовища, PACS без DICOM втрачає стандартизовану основу для роботи із зображеннями, а HL7 без них не дає можливості повноцінно інтегрувати результати візуальної діагностики в загальний клінічний процес.

Для розподіленої системи обміну DICOM-зображеннями це означає необхідність сумісності одразу з кількома рівнями медичної IT-інфраструктури. На периферійному рівні система повинна працювати з DICOM-файлами, на рівні центрального зберігання – взаємодіяти з PACS-сервером, а на рівні загальної інформаційної інтеграції – враховувати стандарти HL7. Такий підхід забезпечує не лише технічну працездатність системи, а й можливість її використання в реальному медичному середовищі.

Стандарт DICOM, PACS-системи та протоколи HL7 утворюють технологічну основу сучасного обміну медичними даними. DICOM забезпечує структуроване подання зображень і супровідної інформації, PACS створює середовище для їх архівування та доступу, а HL7 відповідає за інтеграцію в ширший інформаційний контекст медичного закладу. Саме врахування особливостей цих рішень є необхідною умовою для побудови ефективної системи розподіленого обміну DICOM-зображеннями.

1.3 Методи стискування медичних зображень: втратні та безвтратні алгоритми

Питання стискування медичних зображень безпосередньо пов'язане з практикою їх зберігання та передавання. У медичній системі робота ведеться не з одним файлом, а з цілими серіями досліджень, які надходять від томографів, рентгенівських апаратів, мамографів та інших діагностичних комплексів. Коли такі дані накопичуються в архіві або передаються між вузлами мережі, швидко стає помітною різниця між звичайним збереженням файлів і раціонально організованим обміном. Обсяг даних впливає на швидкість доступу до серій, навантаження на мережу, вимоги до серверної частини й загальну стабільність роботи PACS-середовища. Через це стискування розглядається не як другорядна операція, а як один із ключових механізмів, що дозволяє оптимізувати обіг медичних зображень.

Загалом методи стискування медичних зображень поділяються на дві великі групи: втратні та безвтратні. Перша група орієнтована на суттєве зменшення

розміру файлу за рахунок часткового видалення даних, друга – на усунення надлишковості без зміни вихідного зображення. Загальну класифікацію цих підходів доцільно подати на рис. 1.3.

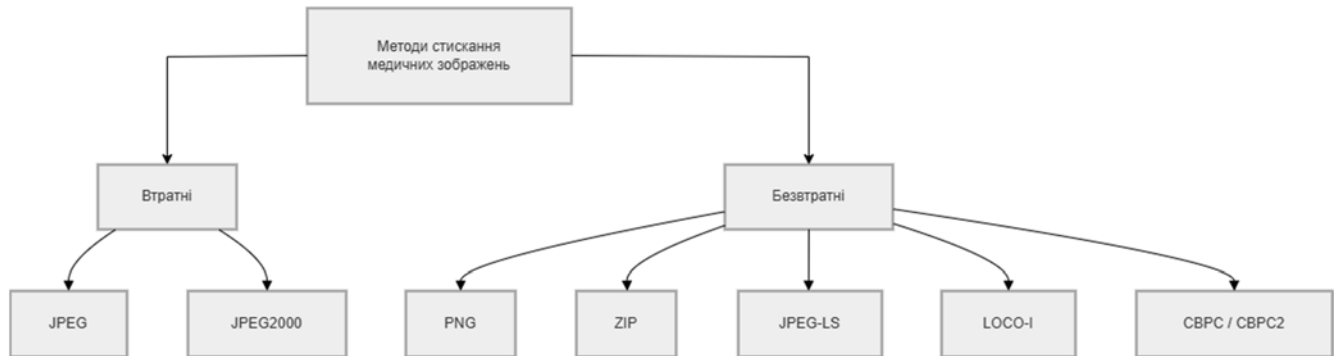


Рисунок 1.3 – Класифікація методів стиснення медичних зображень

Втратне стиснення найчастіше розглядається там, де головним є саме зменшення обсягу файлу. Такі алгоритми дозволяють досягати високого ступеня стиснення, що виглядає привабливо для швидкого передавання великих обсягів даних. До найбільш відомих підходів належать JPEG та JPEG2000, хоча між ними існують помітні відмінності в принципах оброблення зображень. Для побутових або загальних мультимедійних задач цього часто достатньо, адже візуальна якість сприймається людиною як основний критерій. У медичному середовищі ситуація інша. Знімок використовується не для звичайного перегляду, а як об'єкт аналізу. Дрібні контрастні переходи, слабо виражені межі структур або локальні зміни тканин не повинні зникати лише тому, що файл зменшили. Саме тому втратні методи в медицині мають обмежену сферу застосування: вони можуть бути доречними для попереднього перегляду, ознайомлення або для допоміжних копій, але для роботи з клінічно значущими даними цього підходу недостатньо.

Безвтратне стиснення в цьому сенсі сприймається як значно надійніше. Після декодування зображення має повністю відновлюватися, без зміни початкових значень. Це особливо важливо для DICOM-даних, де точність відтворення є не просто бажаною, а обов'язковою умовою. У цій групі використовуються різні підходи: словникові, статистичні, предикторні, контекстно-залежні. Частина з них добре відома як універсальні рішення, наприклад ZIP, частина орієнтована

безпосередньо на графічні дані, як-от PNG або JPEG-LS . Окремий інтерес становлять предикторні алгоритми, у яких значення поточного пікселя оцінюється за сусідніми елементами зображення, а похибка такого передбачення вже кодується. Для медичних зображень цей підхід є логічним, оскільки в їхній структурі часто присутні повторювані локальні закономірності, однорідні області та переходи, які добре піддаються такому способу оброблення. До цієї групи належать LOCO-I, а в контексті даної теми – також СВРС і СВРС2, де використовуються адаптивне прогнозування та подальше кодування різниці між реальним і передбаченим значенням [10; 11].

Різницю між окремими алгоритмами зручніше оцінювати не описово, а в зіставленні. Для цього доцільно порівнювати тип стискання, характер відновлення даних і загальну придатність для роботи з медичними зображеннями. Таке порівняння подано в табл. 1.3.

Таблиця 1.3 – Порівняльна характеристика методів стискання медичних зображень

Алгоритм / формат	Тип стискання	Характер відновлення	Особливості застосування
JPEG	Втратне	Неповне	Високий ступінь стискання, але можливе спотворення дрібних деталей
JPEG2000	Втратне / безвтратне	Часткове або повне залежно від режиму	Гнучкіший підхід, вищі вимоги до оброблення
PNG	Безвтратне	Повне	Підходить для зберігання без зміни даних, але не завжди оптимальний для великих серій
ZIP	Безвтратне	Повне	Універсальний метод, зручний для загального стискання файлів
JPEG-LS	Безвтратне	Повне	Орієнтований на ефективне стискання зображень
LOCO-I	Безвтратне	Повне	Предикторний підхід, придатний для роботи із структурованими зображеннями
СВРС / СВРС2	Безвтратне	Повне	Адаптивне прогнозування та кодування похибки, орієнтація на медичні зображення

У розподіленій системі обміну DICOM-зображеннями стискання не може оцінюватися лише через коефіцієнт зменшення розміру файлу. Для периферійного

вузла важлива швидкість локального оброблення, для мережі – скорочення трафіку, для архіву – точність відновлення даних, а для лікаря – відсутність будь-яких змін у самому зображенні. Саме в такому поєднанні вимог безвтратні методи отримують перевагу. Вони не завжди забезпечують настільки високий ступінь стискання, як втратні алгоритми, але відповідають основній умові медичного обміну: збереження діагностичної достовірності. У межах цієї теми найбільший інтерес тому і зосереджується на СВРС та СВРС2, оскільки ці алгоритми розглядаються не як абстрактні схеми стискання, а як частина архітектури системи, де потрібно поєднати точне відновлення зображення, зменшення обсягу даних і придатність до роботи на периферійних обчислювальних вузлах [6; 10; 12].

1.4 Огляд сучасних систем і сервісів обміну медичними даними (Orthanc, MIDOM, DCMTK, OsiriX)

Середовище обміну медичними зображеннями не зводиться до одного універсального програмного продукту. У реальній інфраструктурі працюють різні типи рішень: сервери архівування, інструментальні набори для підтримки DICOM-взаємодії, клієнтські переглядачі та спеціалізовані системи, орієнтовані на передавання зображень у конкретному прикладному сценарії. Через це порівнювати такі системи лише за кількістю функцій було б недостатньо. Набагато важливіше зрозуміти, яку саме роль виконує кожне рішення в загальній схемі обміну медичними даними.

Orthanc у такому огляді виглядає найближчим до серверної основи системи [7]. На офіційному сайті він описаний як *lightweight DICOM server* з вебінтерфейсом, прийманням і передаванням DICOM-зображень, а також REST API для автоматизації *imaging flows*. Для розподіленого обміну це має цілком практичний зміст: сервер такого типу придатний для ролі центрального вузла, який накопичує медичні зображення, приймає дані від периферійних модулів і забезпечує програмний доступ до архіву. Orthanc зручний саме тим, що поєднує PACS-подібну логіку з відносно простою інтеграцією через API [8]. У системі, де

частина дій виконується на локальних вузлах, така властивість особливо корисна, бо центральний рівень повинен не лише зберігати зображення, а й взаємодіяти з зовнішніми сервісами без зайвої складності.

DCMТК належить до іншої категорії [19]. Це не готове серверне середовище в тому вигляді, у якому ним є Orthanc, а набір бібліотек і застосунків, що реалізують значну частину стандарту DICOM. OFFIS прямо вказує, що DСMТК включає засоби для перегляду, побудови й перетворення DICOM-файлів, роботи з носіями зберігання, а також для надсилання і приймання зображень по мережі. Такий інструментарій важливий там, де потрібно гнучко працювати з DICOM-рівнем, тестувати мережеву взаємодію або розробляти спеціалізовані компоненти. Для власної розподіленої системи це робить DСMТК цінним не як центральну платформу архівування, а як допоміжну інструментальну основу, придатну для реалізації окремих сервісних механізмів [20].

OsiriX, навпаки, логічно віднести не до серверного й не до інструментального, а до клієнтського рівня [13]. Офіційний сайт подає його як DICOM viewer із високою продуктивністю та інтерфейсом для роботи з медичними зображеннями; окремо зазначено, що OsiriX читає й відображає різні типи DICOM-файлів, включно з CT, MRI, ultrasound і standard X-rays. У такому контексті його основна цінність пов'язана не з архівуванням, а з переглядом серій, навігацією по дослідженнях і візуальною роботою користувача із зображеннями. Для системи обміну це означає, що OsiriX не виступає ядром серверної архітектури, але добре демонструє, яким має бути кінцевий доступ до даних на стороні спеціаліста.

MIDOM варто розглядати окремо, бо це вже не універсальний DICOM-сервер і не просто переглядач. У статті 2023 року MIDOM описано як комунікаційна система медичних зображень на базі DICOM, покращена варіантами спеціальними безвтратними-СВРС методами стиснення; там же підкреслюється ризик того, що втрати у поданні зображення можуть бути неправильно інтерпретовані та вплинути на діагностику [6]. Це робить MIDOM не просто ще одним продуктом у списку, а концептуально близьким прикладом, у якому поєднуються обмін медичними

зображеннями та безвтратне стискування. Саме ця властивість є найбільш цінною в аналітичному сенсі, тому що вона напряду перетинається з логікою системи розподіленого обміну DICOM-зображеннями з безвтратним стискуванням.

Місце цих рішень у структурі медичної IT-інфраструктури доцільно показати не як перелік назв, а як розподіл функціональних ролей: серверний рівень, інструментальний рівень, клієнтський рівень і спеціалізоване рішення для передавання зі стискуванням. Таке узагальнення можна подати на рис. 1.4.

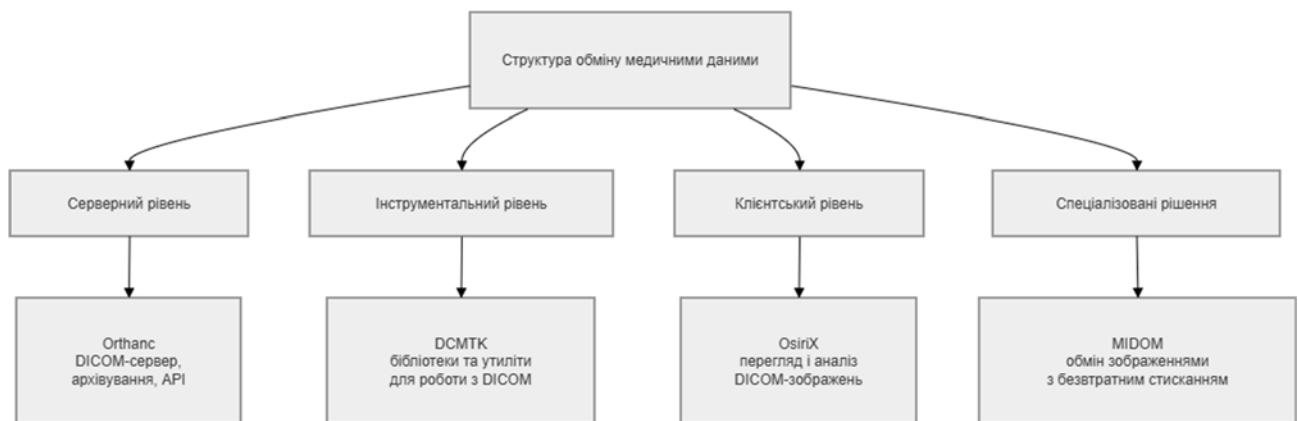


Рисунок 1.4 – Місце Orthanc, MIDOM, DCMTK та OsiriX у структурі обміну медичними даними

На рівні практичного порівняння цього недостатньо, тому основні характеристики систем доцільно звести в табл. 1.4.

Таблиця 1.4 – Порівняльна характеристика сучасних систем і сервісів обміну медичними даними

Система / сервіс	Основне призначення	Сильні сторони	Обмеження	Доцільність для системи розподіленого обміну
Orthanc	DICOM-сервер, архівування, API-взаємодія	Компактність, відкритість, REST API, робота з DICOM	Для складнішої інфраструктури може потребувати додаткових компонентів	Доцільний як центральний PACS-вузол
DCMTK	Бібліотеки та утиліти для роботи з DICOM	Гнучкість, низькорівневий контроль, мережеві інструменти	Не є готовим серверним середовищем для кінцевого користувача	Доцільний як інструментальна база

Система / сервіс	Основне призначення	Сильні сторони	Обмеження	Доцільність для системи розподіленого обміну
OsiriX	Перегляд і аналіз DICOM-зображень	Зручність перегляду, клінічна орієнтація, підтримка різних типів DICOM	Не виконує роль серверного сховища	Доцільний як клієнтський інтерфейс
MIDOM	Передавання DICOM-зображень із безвтратним стисканням	Поєднання DICOM-обміну з стисненням без втрат	Не позиціонується як універсальний PACS-сервер	Цінний як концептуально близьке спеціалізоване рішення

На цьому тлі найбільш природною серверною основою для системи розподіленого обміну виглядає Orthanc. DCMTK краще сприймати як інструментарій для низькорівневої роботи з DICOM, OsiriX – як клієнтський засіб перегляду, а MIDOM – як приклад підходу, де безвтратне стискання вбудовується безпосередньо в логіку медичного передавання зображень. Таке розмежування корисне для подальшого проектування, бо дозволяє не змішувати ролі різних компонентів і чіткіше визначити, що саме потрібно взяти за основу власної системи[5–6].

1.5 Аналіз апаратних платформ для побудови вузлів PACS-систем (Raspberry Pi, Jetson Nano, NUC)

Побудова розподіленої системи обміну DICOM-зображеннями неможлива без вибору апаратної платформи, на якій працюватимуть периферійні вузли. У такому вузлі зосереджено кілька практичних функцій: приймання даних від джерела, попередня обробка серій зображень, виконання безвтратного стискання, тимчасове збереження, передавання до центрального PACS-сервера та, за потреби, взаємодія з локальною мережею медичного закладу. Через це сама платформа має оцінюватися не лише за продуктивністю, а й за габаритами, енергоспоживанням, вартістю, наявністю мережевих інтерфейсів, підтримкою операційної системи та зручністю інтеграції в загальну інфраструктуру.

У межах цієї теми доцільно розглядати не абстрактні комп'ютери загального призначення, а саме платформи, які реально можуть бути використані як компактні

вузли обміну DICOM-даними. Найчастіше в подібних задачах порівнюють одноплатні комп'ютери та компактні mini-PC. До першої групи належать Raspberry Pi 4 і Jetson Nano, до другої – пристрої класу NUC. Попри зовнішню схожість у вигляді невеликого корпусу або плати, їхнє призначення та практична роль у системі відрізняються досить суттєво.

Raspberry Pi 4 сприймається як один із найбільш доступних варіантів для побудови периферійного вузла [14]. Його перевага полягає не в рекордній обчислювальній потужності, а в збалансованості. Платформа підтримує Linux-середовище, має мережеві інтерфейси, достатній набір портів для підключення накопичувачів і периферії, а також дозволяє розгортати сервісні модулі без складного апаратного обслуговування. Для вузла системи обміну DICOM-зображеннями цього часто достатньо: тут не потрібне надпотужне графічне середовище або складна локальна аналітика; натомість потрібна стабільна робота, можливість стискати дані й передавати їх у центральне сховище. Саме така логіка робить Raspberry Pi 4 привабливим варіантом як основу для локального мікросервера.

Jetson Nano належить до іншого класу одноплатних платформ. Його зазвичай асоціюють не просто з локальним обробленням даних, а з задачами комп'ютерного зору, AI-обробки та пришвидшеними обчисленнями. Це надає йому іншу сильну сторону: якщо система обміну медичними зображеннями повинна виконувати не тільки стискання й передавання, а й додатковий попередній аналіз вхідних знімків, така платформа виглядає перспективніше. У чистій задачі маршрутизації DICOM-файлів і безвтратного стискання її можливості часто виявляються надлишковими. Через це Jetson Nano варто сприймати не як універсально кращу платформу, а як спеціалізоване рішення для сценаріїв, де поряд із передаванням зображень виникає потреба в локальному інтелектуальному аналізі [15; 16].

NUC-пристрої, у свою чергу, ближчі до компактних персональних комп'ютерів або mini-PC, ніж до класичних одноплатних вузлів [17]. Вони забезпечують значно вищий рівень продуктивності, більшу гнучкість у виборі

конфігурації пам'яті й накопичувачів, а також звичніше середовище для розгортання серверного або клієнтського АПЗ. Для систем медичного обміну це означає, що NUC може без проблем працювати як локальний сервер, допоміжний вузол інтенсивного оброблення або навіть окрема станція для специфічних сервісних задач. Разом із цим, такий клас пристроїв має інший баланс характеристик: вища вартість, більші вимоги до живлення, менша доцільність використання там, де потрібен не універсальний обчислювальний модуль, а недорогий розподілений вузол із чітко визначеним функціоналом.

Порівнювати ці платформи лише за тактовою частотою процесора або обсягом оперативної пам'яті було б надто спрощено. Для системи обміну DICOM-зображеннями набагато важливішим є те, як саме пристрій поводить себе в реальних умовах експлуатації: наскільки просто розгортається, чи підтримує стабільну мережеву взаємодію, чи може працювати тривалий час без складного супроводу, наскільки зручно підключати накопичувачі та допоміжні модулі, чи не створює платформа зайвих витрат там, де можна обійтися компактнішим рішенням. Схематичне співвідношення цих платформ за ролями в системі подано на рис. 1.5.

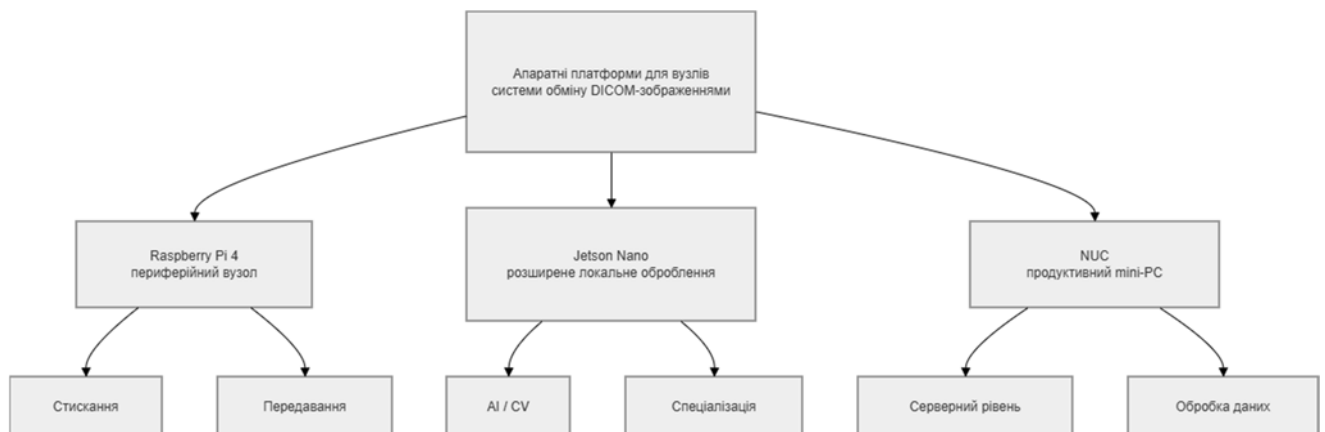


Рисунок 1.5 – Порівняння апаратних платформ для побудови вузлів системи обміну DICOM-зображеннями

У цій схемі Raspberry Pi 4 доцільно розглядати як базовий периферійний вузол, Jetson Nano – як платформу для розширеного локального оброблення, а NUC – як компактний продуктивний вузол, ближчий до серверного або робочого

рівня. Таке розмежування дозволяє одразу побачити, що йдеться не про пряме суперництво трьох однакових за призначенням рішень, а про різні варіанти реалізації вузлів із різним функціональним акцентом.

Для більш предметного порівняння основні властивості цих платформ доцільно звести в табл. 1.5.

Таблиця 1.5 – Порівняльна характеристика Raspberry Pi 4, Jetson Nano та NUC

Платформа	Клас пристрою	Сильні сторони	Обмеження	Найдоцільніша роль у системі
Raspberry Pi 4	Одноплатний комп'ютер	Доступність, компактність, низьке енергоспоживання, зручність розгортання	Обмежений запас продуктивності для складного аналізу	Периферійний вузол приймання, стискування та передавання DICOM-даних
Jetson Nano	Одноплатна обчислювальна платформа	Краща придатність до AI- та CV-обробки, гнучкість у локальному аналізі зображень	Надлишковість для простого вузла маршрутизації та стискування	Спеціалізований вузол із розширеною локальною обробкою
NUC	Компактний mini-PC	Вища продуктивність, гнучкість конфігурації, зручність для складнішого ПЗ	Вища вартість і менша доцільність для масового периферійного розгортання	Серверний або напівсерверний вузол, локальний центр обробки

Raspberry Pi 4 не є найпотужнішим рішенням, але саме це і не є головним критерієм для периферійного вузла системи обміну медичними зображеннями. Його перевага полягає у співвідношенні вартості, функціональності та простоти впровадження. У задачі безвратного стискування й передавання DICOM-даних важливішою є стабільність і достатність ресурсів, ніж максимальний обчислювальний запас. Jetson Nano має сенс там, де локальний вузол повинен виконувати складнішу аналітику, а NUC – там, де один пристрій бере на себе функції більш важкого сервісного або серверного рівня.

Для розподіленої системи обміну DICOM-зображеннями, у якій периферійний вузол має працювати як компактний і доступний елемент архітектури, Raspberry Pi 4 виглядає найбільш збалансованим варіантом. Така

платформа достатня для приймання даних, їх локального стискання та подальшої передачі до центрального PACS-сервера. Вона не перевантажує систему зайвими апаратними можливостями, але й не зводить вузол до надто спрощеного рівня. Jetson Nano і NUC у цьому порівнянні не втрачають значення: вони допомагають точніше окреслити межі застосування Raspberry Pi 4 і показують, за яких умов доцільно переходити до більш спеціалізованих або продуктивніших платформ.

1.6 Формування вимог до апаратно-програмного забезпечення

Після аналізу стандарту DICOM, ролі PACS / HL7, методів стискання та доступних апаратних платформ доцільно перейти від опису предметної області до формування узагальнених вимог. На цьому етапі важливо не повторювати окремі положення попередніх підрозділів, а звести їх до набору практичних характеристик, яким має відповідати система розподіленого обміну DICOM-зображеннями.

Передусім система повинна залишатися сумісною з медичним середовищем. Це означає коректну роботу з DICOM-файлами, збереження службових атрибутів дослідження та можливість інтеграції з центральним PACS-сервером без порушення структури даних. Для цієї роботи принципово є також підтримка взаємодії з Orthanc через REST API, оскільки саме такий підхід забезпечує простий і зрозумілий канал обміну між периферійним вузлом і серверною частиною [7; 8].

Другою групою є вимоги до локальної обробки. Периферійний вузол повинен приймати DICOM-об'єкти, виконувати безвтратне стискання, тимчасово зберігати результати та передавати їх далі без зміни діагностично значущої інформації. Для цього програмна логіка має бути достатньо легкою для роботи на Raspberry Pi 4, але водночас придатною до обробки реальних медичних зображень [5; 6].

Окремо слід виділити вимоги до мережевої взаємодії. Система повинна підтримувати стабільне передавання даних у локальній мережі, повторні спроби надсилання у випадку збоїв, базовий контроль стану з'єднання та фіксацію результату на сервері. Для експериментального стенду цього достатньо, щоб

оцінити не лише працездатність окремого скрипта, а повний маршрут проходження DICOM-даних від периферійного вузла до архівної структури PACS-сервера.

Наступний блок становлять вимоги до безпеки й надійності. Оскільки DICOM-файли містять відомості про пацієнта та дослідження, передавання має виконуватися через захищений канал з авторизацією та контролем доступу. Не менш важливою є цілісність даних після стискання, надсилання та збереження, адже будь-яке порушення структури файлу або службових полів робить таку систему непридатною для практичного використання [3; 10].

З погляду апаратної реалізації доцільним є компактний, енергоощадний і відносно простий у розгортанні вузол. Умовам такої системи відповідає одноплатний комп'ютер, який не потребує складної серверної інфраструктури, але дозволяє запускати мережеві сервіси, Python-модулі обробки та механізми взаємодії з центральним сховищем. Саме тому вибір Raspberry Pi 4 у межах цієї теми є не випадковим, а безпосередньо пов'язаним із сформованими вимогами до периферійного рівня [5].

Узагальнюючи, для розроблюваної системи визначальними є такі властивості: сумісність із DICOM / PACS-середовищем, безвратна локальна обробка, стабільний мережевий обмін, захищене передавання та можливість реалізації на доступній одноплатній платформі. Саме ці вимоги створюють змістовну основу для другого розділу, де вже розглядаються архітектура системи, вибір апаратної конфігурації та практична реалізація основних модулів.

Висновки до розділу 1

У першому розділі проаналізовано теоретичні засади обміну медичними зображеннями в сучасному DICOM-середовищі. Встановлено, що складність роботи з такими даними зумовлена насамперед їхнім великим обсягом і специфікою передавання. Це особливо критично для серійних досліджень, коли стрімке зростання кількості файлів перевантажує мережу, сервери та архіви. У

таких умовах виключно централізований підхід уже не здатний забезпечити стабільну роботу системи.

Аналіз базових стандартів показав необхідність їхньої тісної взаємодії: DICOM стандартизує подання зображень, PACS забезпечує їхнє збереження та доступ, а HL7 інтегрує ці дані до загального клінічного документообігу. Лише комплексне поєднання цих технологій дозволяє створити систему, сумісну з реальною медичною інфраструктурою, а не просто ізольований програмний модуль.

Окрему увагу приділено проблематиці стиснення даних. Обґрунтовано, що для збереження діагностичної цінності знімків клінічне застосування вимагає виключно безвратних алгоритмів. Найбільший практичний інтерес у межах дослідження становлять адаптивні предикторні методи, зокрема алгоритми СВРС та СВРС2, які ефективно зменшують обсяг даних без жодного спотворення медичної інформації.

Огляд сучасних інструментальних рішень дозволив сформувати оптимальний стек технологій. Для центрального PACS-рівня доцільно використовувати Orthanc, інструментальною базою DICOM слугуватиме DCMTK, а клієнтським рівнем – OsiriX. Щодо апаратної основи, найбільш збалансованим варіантом для периферійного вузла визначено Raspberry Pi 4, який поєднує компактність, доступність і достатню продуктивність для локального приймання, стискання та передавання DICOM-даних.

2 ПРОЄКТУВАННЯ РОЗПОДІЛЕНОЇ СИСТЕМИ ОБМІНУ DICOM-ДАНИМИ

2.1 Загальна архітектура системи та принципи функціонування

Архітектура системи розподіленого обміну DICOM-зображеннями повинна враховувати одразу кілька практичних умов. З одного боку, медичні зображення потрібно приймати та передавати без порушення структури DICOM-даних. З іншого – саме передавання не повинно створювати надмірне навантаження на мережу і центральний вузол зберігання. Саме тому в основу системи доцільно закласти розподілений принцип, за якого частина операцій виконується не в одному центрі, а на периферійних вузлах, розташованих ближче до джерел формування даних.

Кожен периферійний вузол виконує не лише роль проміжної точки передавання. Він приймає медичні зображення, виконує їх первинну підготовку, формує локальний пакет даних для надсилання, запускає безвтратне стискання та передає підготовлений результат до центрального PACS-сервера. Центральна частина архітектури, своєю чергою, відповідає за накопичення досліджень, збереження архіву, індексацію даних, доступ клієнтських модулів і загальну координацію обміну між вузлами. Такий розподіл функцій дозволяє не концентрувати весь обсяг оброблення в одному сервері та зменшує кількість необроблених DICOM-даних, що проходять через мережу.

У загальному вигляді система може бути подана як сукупність трьох рівнів. Перший рівень утворюють джерела даних – діагностичне обладнання, локальні робочі станції або інші модулі, з яких надходять DICOM-зображення. Другий рівень складають периферійні вузли на базі одноплатних комп'ютерів, де реалізується приймання, локальне стискання, тимчасове збереження та захищене надсилання даних. Третій рівень – центральне сховище PACS, де зображення накопичуються, залишаються доступними для перегляду та можуть передаватися далі в інші сервіси медичної інформаційної інфраструктури. Структурну схему взаємодії цих рівнів доцільно подати на рис. 2.1.



Рисунок 2.1 – Загальна архітектура системи розподіленого обміну
DICOM-зображеннями

Для периферійного вузла важливим є те, що він повинен працювати автономно в межах своєї локальної ділянки системи. Якщо зображення надходять серіями, вузол має прийняти їх без втрати структури, підготувати до стискування, обробити та передати далі. У разі тимчасового розриву з'єднання з центральним сервером доцільно передбачити буферизацію або локальне кешування. Це дозволяє не втрачати дані й не зупиняти роботу всієї системи через короткочасні проблеми мережевої взаємодії. Саме тут розподілена архітектура показує свою практичну перевагу: відмова або перевантаження одного вузла не обов'язково призводить до повного припинення обміну в інших частинах системи.

Архітектура повинна залишатися модульною. Логічно відокремити модуль приймання DICOM-даних, модуль локального безвтратного стискування, модуль взаємодії з центральним PACS-сервером, модуль авторизації та захищеного передавання, а також компонент журналювання подій і контролю стану вузла. Такий поділ не є формальністю. Він дає змогу змінювати або вдосконалювати окремі частини системи без повної перебудови всієї структури. Наприклад, алгоритм стискування можна оновлювати незалежно від механізму мережевого обміну, а серверну частину – розширювати без зміни принципу роботи периферійних вузлів.

Слід враховувати, що медичні зображення в такій системі не існують поза своїм контекстом. Разом із передаванням зображення має зберігатися його належність до певного дослідження, серії, пацієнта та набору службових атрибутів. Через це архітектура повинна підтримувати коректну роботу зі структурою DICOM-файлу на кожному етапі – від приймання до збереження в архіві. Без цього система може втратити сумісність із PACS-середовищем навіть у тому випадку, якщо самі графічні дані були передані правильно.

Значну роль у функціонуванні системи відіграє і механізм безвтратного стискання. Він не повинен розглядатися як окрема зовнішня надбудова. У розподіленій архітектурі стискання є частиною основного маршруту обробки даних. Зображення надходить на локальний вузол, проходить перевірку, за потреби готується до обробки, стискається, а вже потім передається у центральне сховище. Такий підхід зменшує обсяг мережевого трафіку ще до надсилання даних, а отже впливає не лише на архівування, а й на загальну швидкість всієї системи.

Принцип функціонування системи можна описати як послідовність взаємопов'язаних етапів: надходження DICOM-зображення, локальна обробка на периферійному вузлі, безвтратне стискання, захищене передавання до PACS-сервера, архівування та подальший доступ користувачів або зовнішніх модулів. У такій схемі кожен етап має власну роль, але жоден із них не є ізольованим. Від правильності локальної обробки залежить коректність архівування, а від архітектури серверного рівня – можливість масштабування системи в цілому.

Для розподіленої системи обміну DICOM-зображеннями доцільно виділити кілька базових принципів функціонування. Перший із них стосується сумісності з медичними стандартами. Система повинна працювати з DICOM-даними без порушення їх структури та бути придатною до інтеграції з PACS-середовищем. Другий пов'язаний із перенесенням частини навантаження на периферійний рівень, де стискання та первинна обробка виконуються до передачі в центральне сховище. Третій стосується безпеки: обмін між вузлами повинен відбуватися через захищений канал зв'язку з контролем доступу й перевіркою цілісності даних. Четвертий принцип – масштабованість, тобто можливість додавання нових вузлів без перебудови всієї архітектури.

Сама система в такому вигляді не зводиться лише до ланцюга «зображення – сервер». Вона охоплює мережеву взаємодію, локальний рівень попереднього оброблення, серверний PACS-рівень, механізм стискання, засоби безпеки та доступ до збережених досліджень. Саме така побудова створює основу для подальшого переходу до вибору конкретної апаратної платформи, визначення

мережевого середовища, інтеграції алгоритму СВРС/СВРС2 у серверну частину та реалізації окремих функціональних модулів системи.

2.2 Обґрунтування вибору апаратного забезпечення

Після визначення загальної архітектури системи потрібно вибрати таку апаратну конфігурацію, яка дозволить реалізувати її без зайвого ускладнення. У даному випадку мова йде не про повноцінну багаторівневу серверну інфраструктуру, а про робочий прототип розподіленої системи, де один вузол приймає та попередньо обробляє DICOM-зображення, а інший забезпечує централізоване збереження, архівування та доступ до них через PACS-середовище. Для такої моделі доцільно використовувати Raspberry Pi 4 як периферійний вузол, а ноутбук або персональний комп'ютер – як центральний вузол зберігання.

Такий вибір добре відповідає практиці побудови експериментального стенду. На цьому етапі важливо не створити складну серверну систему з надлишковими компонентами, а перевірити працездатність основних функцій: приймання DICOM-даних, локального безвратного стискання, передавання до центрального PACS-сервера та подальшої роботи з архівом. Саме тому апаратна частина повинна бути достатньою для цих задач, але без зайвої технічної надбудови.

Raspberry Pi 4 у такій конфігурації виконує роль периферійного модуля. На нього покладаються приймання зображень, первинна підготовка даних, запуск алгоритму безвратного стискання та передавання результатів у центральний вузол. Для такого функціоналу ця платформа є цілком придатною. Вона підтримує Linux-середовище, має мережеві інтерфейси, дозволяє працювати з зовнішніми пристроями зберігання і не потребує складного апаратного обслуговування. У межах системи обміну DICOM-зображеннями цього достатньо, оскільки периферійний вузол не повинен виконувати складну графічну аналітику чи функції повноцінного сервера загального призначення. Важливо і те, що Raspberry Pi 4 залишається компактною платформою, яку легко інтегрувати в локальне медичне середовище. Якщо вузол системи має працювати біля джерела формування даних,

великі габарити, підвищене енергоспоживання або складність розгортання починають створювати зайві обмеження. У цьому сенсі Raspberry Pi 4 виглядає більш доречним, ніж потужніші, але надлишкові для такої ролі рішення. Його перевага полягає не в максимальній продуктивності, а в поєднанні доступності, достатності ресурсів і простоти використання [14].

Центральний вузол доцільно описувати не як конкретний пристрій одного типу, а як окремий рівень зберігання і керування. У ролі такого вузла може виступати ноутбук або персональний комп'ютер, на якому розгортається PACS-сервер Orthanc. Головним тут є не форм-фактор пристрою, а його призначення в системі. Центральний вузол приймає дані від периферійних модулів, формує архів досліджень, підтримує пошук і доступ до збережених зображень, а також забезпечує взаємодію з клієнтськими компонентами. Для експериментальної системи цього цілком достатньо, оскільки основний акцент робиться не на великій кількості одночасних користувачів, а на коректній реалізації самого процесу розподіленого обміну. Функцію сховища в цій конфігурації виконує локальний накопичувач центрального вузла. Це може бути SSD або HDD ноутбука чи персонального комп'ютера. Такий варіант не потребує окремого мережевого сховища і добре підходить для прототипу, в якому важливо насамперед перевірити роботу системи, а не розгортати складну багаторівневу архівну інфраструктуру. За потреби в подальшому обсяг локального сховища можна збільшити або винести архів на окремий рівень зберігання, не змінюючи загальної логіки роботи системи [7; 8].

Мережеве середовище в такій системі слід розглядати як локальну мережу, що забезпечує зв'язок між периферійним вузлом і центральним PACS-сервером. У практичній реалізації такий обмін може відбуватися як через дротове підключення, так і через бездротовий сегмент. У даному випадку Raspberry Pi 4 підключається через Wi-Fi, тому саме бездротове з'єднання є робочим варіантом експериментального стенду. Для прототипу цього достатньо, якщо мережа забезпечує стабільну передачу серій DICOM-зображень і не створює критичних затримок у взаємодії з центральним вузлом.

Тут важливий не сам тип каналу як такий, а надійність обміну в межах обраної конфігурації. Raspberry Pi 4 повинен коректно передавати дані до Orthanc, а центральний вузол – приймати їх, зберігати і надавати до них доступ через серверні сервіси. Якщо в архітектурі передбачено REST API і захищений канал зв'язку, мережеве середовище має підтримувати таку взаємодію без порушення цілісності переданих даних. Для експериментальної системи Wi-Fi може бути достатнім, особливо коли мережа стабільна, а стенд працює в контрольованих умовах [7–8; 14].

Схематично ця конфігурація зводиться до взаємодії джерела DICOM-даних, периферійного вузла на Raspberry Pi 4, локальної мережі Wi-Fi та центрального вузла зберігання, реалізованого на ноутбучі або персональному комп'ютері з Orthanc. Така апаратна побудова є простою, але цілком достатньою для перевірки логіки розподіленого обміну, локального стискування та архівування медичних зображень, яка наведена на рис. 2.2.

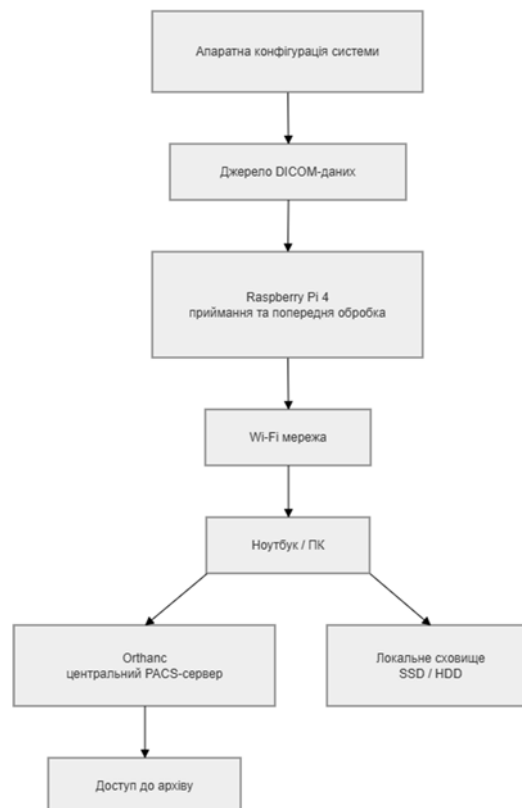


Рисунок 2.2 – Апаратна конфігурація системи на основі Raspberry Pi 4 та центрального вузла зберігання

У цій схемі Raspberry Pi 4 бере на себе периферійне приймання і попередню обробку, а центральний вузол забезпечує PACS-рівень і збереження архіву. Такий розподіл ролей дозволяє уникнути перевантаження периферійного модуля серверними функціями та водночас не переносити на центральний вузол ті операції, які доцільно виконувати локально до передавання даних.

Вибір саме такої конфігурації є виправданим для експериментальної системи розподіленого обміну DICOM-зображеннями. Вона не потребує складної інфраструктури, дозволяє реалізувати всі базові функціональні етапи роботи системи й водночас залишається достатньо гнучкою для подальшого розширення. Raspberry Pi 4 у ролі периферійного вузла і ноутбук або ПК у ролі центрального PACS-сервера утворюють практичну апаратну основу для наступного етапу інтеграції алгоритму безвтратного стискування, налаштування взаємодії через Orthanc і проведення експериментального оцінювання всієї системи [7; 14].

2.3 Використання алгоритму безвтратного стискування СВРС / СВРС2

У розроблюваній системі безвтратне стискування є не допоміжною, а однією з ключових операцій, оскільки саме на цьому етапі зменшується обсяг даних, які передаються від периферійного вузла до центрального PACS-сервера. Через це алгоритм стискування повинен відповідати кільком базовим вимогам: не змінювати вміст медичного зображення, працювати достатньо швидко на Raspberry Pi 4 і не ускладнювати подальшу інтеграцію із серверною частиною системи. Для реалізації такого підходу як основу доцільно використати алгоритм СВРС і його вдосконалену модифікацію СВРС2 [6].

Загальна ідея цих алгоритмів пов'язана з тим, що значення пікселів у медичному зображенні зазвичай не змінюються випадково. Між сусідніми елементами існує локальний зв'язок, особливо в однорідних областях або на плавних переходах. Саме ця властивість і використовується під час стискування. Замість того щоб кодувати кожен піксель як окреме незалежне значення, алгоритм

спочатку оцінює, яким це значення може бути на основі вже відомого локального оточення, а далі працює вже з різницею між прогнозом і реальним значенням [6].

У практичній реалізації це означає, що для кожного поточного пікселя формується локальний контекст із сусідніх елементів, які вже були оброблені раніше. Далі виконується класифікація цього контексту, тобто визначається характер локальної ділянки зображення. Якщо навколо поточного пікселя спостерігається майже однорідна область, алгоритм використовує один спосіб оцінки. Якщо в цій зоні є виражена межа або помітна зміна яскравості, застосовується інший варіант прогнозування. Після цього обчислюється різниця між реальним значенням пікселя та його передбаченим значенням, і вже ця різниця подається на подальше кодування. Така побудова є логічною саме для медичних зображень. У томографічних, рентгенівських і подібних даних часто є великі області з плавною зміною яскравості, однорідні фрагменти та локальні переходи між структурами. Якщо правильно врахувати цей локальний контекст, похибка передбачення в багатьох випадках виявляється невеликою. Саме це створює сприятливі умови для безвтратного стискання, оскільки невеликі значення різниці простіше закодувати компактно.

У класичному варіанті СВРС велика увага приділяється класифікації локального контексту та поєднанню різних способів прогнозування. Це дозволяє адаптувати алгоритм до різних типів ділянок зображення. У рівномірних фрагментах один варіант прогнозу може працювати краще, а в областях з вираженими межами – інший. Саме тому алгоритм не обмежується єдиним універсальним предиктором, а намагається підлаштовуватися під структуру конкретної локальної області. Для практичної реалізації в межах системи більш доцільним є використання модифікації СВРС2. У ній збережено базову логіку контекстного прогнозування, але сама схема обробки стала більш зручною для програмної реалізації на периферійному вузлі. Класифікація локального оточення в такому варіанті залишається достатньо інформативною для роботи з медичними зображеннями, але водночас не перевантажує систему зайвими обчисленнями. Це

важливо саме для Raspberry Pi 4, де алгоритм повинен працювати стабільно в реальному середовищі, а не лише в теоретичній моделі [6].

Після визначення похибки передбачення її необхідно подати у вигляді, зручному для подальшого кодування. Оскільки така похибка може бути як додатною, так і від'ємною, перед наступним етапом обробки вона перетворюється до невід'ємного представлення. Далі отримана послідовність значень кодується ентропійним методом, придатним для даних, у яких невеликі значення трапляються значно частіше, ніж великі. Для цього в алгоритмі використовується підхід типу Golomb-Rice, оскільки він добре підходить для послідовностей похибок передбачення, характерних для медичних зображень після локального прогнозування [6; 10–11].

Для системної реалізації важливо, що обробка виконується не над абстрактним масивом чисел, а над реальним DICOM-зображенням. Це означає, що спочатку з файлу потрібно коректно виділити піксельні дані, не порушивши службову структуру самого об'єкта. Стискання застосовується саме до графічної частини зображення, тоді як метадані дослідження повинні зберігатися окремо й залишатися доступними для подальшої передачі в PACS-середовище. Такий підхід є принциповим, оскільки система працює не тільки з візуальною інформацією, а з повноцінним DICOM-об'єктом, що має власну структуру і службові атрибути. Зворотний процес у СВРС / СВРС2 організований симетрично до етапу стискання. Спочатку декодується стиснений потік, потім відновлюється похибка передбачення, після чого на основі вже декодованого локального контексту повторно обчислюється прогноз для кожного пікселя. Додавання відновленої похибки до прогнозованого значення дає початкове зображення. У результаті після декодування формується той самий масив даних, який був до стискання. Саме ця властивість і визначає алгоритм як безвтратний [3; 4].

У межах розроблюваної системи алгоритм повинен працювати не лише з окремим зображенням, а і з серіями DICOM-файлів. Через це програмна реалізація має підтримувати послідовну обробку кількох досліджень або кількох зображень

без ручного втручання для кожного окремого файлу. Такий режим роботи є важливим для периферійного вузла, який у реальній системі отримує потік даних, а не одиничний тестовий об'єкт. З практичної точки зору це означає, що модуль стискання повинен автоматично приймати зображення, обробляти їх, формувати результат і передавати його далі в межах загального циклу обміну.

Схему роботи алгоритму безвтратного стискання СВРС / СВРС2 подано на рис. 2.3.



Рисунок 2.3 – Загальна схема роботи алгоритму безвтратного стискання
СВРС / СВРС2

Розробка СВРС / СВРС2 розглядається не як ізольоване математичне завдання, а як частина архітектури розподіленого обміну DICOM-зображеннями. Алгоритм повинен не просто зменшувати обсяг даних, а робити це в умовах реального периферійного вузла, з урахуванням обмежених ресурсів Raspberry Pi 4, подальшого мережевого передавання і сумісності з центральним PACS-сервером. Саме тому наступним етапом є інтеграція механізму стискування в серверну частину системи та включення його в повний цикл обробки медичних зображень [6].

2.4 Інтеграція алгоритму стискування у серверну частину PACS-системи

Після розробки алгоритму безвтратного стискування наступним етапом стає його включення в загальну серверну логіку системи. Сам по собі алгоритм ще не вирішує задачу обміну медичними зображеннями, якщо він існує окремо від PACS-середовища. У реальній системі стискування повинно бути пов'язане з прийманням DICOM-даних, передаванням їх між вузлами, збереженням у центральному сховищі та подальшим доступом до архіву. Через це інтеграція алгоритму у серверну частину є не менш важливою, ніж його власне програмна реалізація.

У запропонованій архітектурі центральним вузлом PACS-системи виступає Orthanc. Його використання дає можливість організувати приймання DICOM-зображень, їх збереження, індексацію та доступ через вебінтерфейс або REST API. Для задачі інтеграції це означає, що алгоритм стискування не повинен підмінити собою серверний рівень. Його роль інша: зменшити обсяг даних ще до того, як вони потраплять у центральне сховище, і водночас не порушити логіку роботи PACS-середовища. У такій схемі сервер залишається центральною точкою архівування, а стискування стає частиною маршруту підготовки даних до передавання [7; 8].

Найпростіший варіант такої взаємодії полягає в тому, що периферійний вузол приймає DICOM-файл або серію файлів, виділяє з них піксельні дані, виконує безвтратне стискування та формує результат для передачі на сервер. Після цього центральний вузол одержує підготовлені дані, зберігає їх у структурі PACS-системи та забезпечує подальший доступ до них. У цьому процесі важливо не

втратити службову частину DICOM-об'єкта. Саме тому інтеграція алгоритму не може обмежуватися простою операцією стискання масиву байтів. Вона повинна враховувати, що файл містить як графічні дані, так і метадані, потрібні для ідентифікації дослідження в PACS-середовищі [18].

На практиці це означає, що серверна частина повинна взаємодіяти з алгоритмом на рівні окремих етапів обробки. Спочатку вузол отримує DICOM-зображення. Далі виконується перевірка коректності формату та виділення піксельного масиву. Після цього запускається модуль СВРС/СВРС2, який формує стиснений потік даних. Окремо зберігаються або передаються службові атрибути, які повинні залишатися доступними для PACS-сервера. Уже на завершальному етапі відбувається передавання результату до Orthanc через серверний інтерфейс. Логіку такої інтеграції подано на рис. 2.4 [3; 7–8].

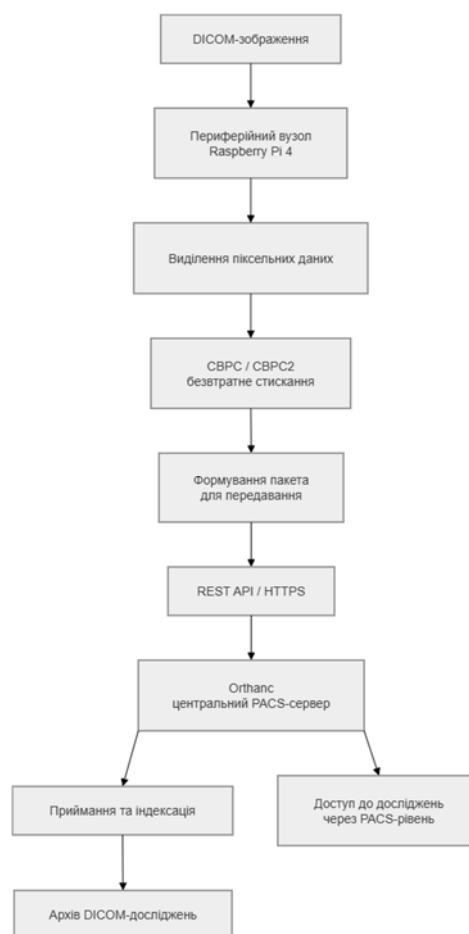


Рисунок 2.4 – Схема інтеграції алгоритму стискання у серверну частину PACS-системи

У такій схемі важливим є розмежування відповідальності між компонентами системи. Алгоритм стискання відповідає за оброблення піксельних даних. Серверний рівень відповідає за збереження дослідження, маршрутизацію, доступ до архіву та взаємодію з клієнтськими модулями. Якщо ці ролі змішуються, структура системи стає менш керованою. Якщо ж вони чітко відокремлені, алгоритм стискання можна вдосконалювати або замінювати без потреби перебудовувати весь PACS-рівень.

Окремо слід зважати на те, що Orthanc у даній системі виконує не тільки функцію архіву, а й роль інтеграційного сервера. Через REST API він може приймати запити від зовнішніх модулів, повертати відомості про дослідження, відкривати доступ до метаданих і забезпечувати програмну взаємодію з периферійними вузлами. Саме це робить його зручним для інтеграції алгоритму стискання. Периферійний модуль не повинен звертатися безпосередньо до внутрішніх механізмів зберігання PACS-сервера. Набагато логічніше використовувати стандартний серверний інтерфейс, через який передаються результати оброблення та пов'язана з ними службова інформація [7; 8].

У контексті цієї теми доцільно розглядати дві логіки інтеграції. Перша полягає в стисканні зображень на периферійному вузлі ще до передачі на сервер. Друга передбачає, що серверна частина приймає результат як уже підготовлений об'єкт і включає його в загальну архівну структуру. Для експериментального стенду більш доречним виглядає саме перший варіант, оскільки він дозволяє зменшити обсяг трафіку ще на етапі мережевої взаємодії. У такій конфігурації сервер не витрачає власні ресурси на стискання, а одержує вже оброблені дані від периферійного вузла.

У серверній частині потрібно також передбачити обробку супровідних сценаріїв. Йдеться про ідентифікацію дослідження, перевірку цілісності переданих даних, облік часу надходження, фіксацію стану обробки та ведення журналу подій. Якщо система повинна працювати стабільно, сервер має не просто приймати файл, а й розуміти, на якому етапі він був сформований, чи був успішно переданий і чи

відповідає він очікуваній структурі. Це особливо важливо в розподіленому середовищі, де кілька вузлів можуть працювати паралельно й передавати серії зображень незалежно один від одного. Не менш суттєвим є питання помилок передавання або збоїв на серверному рівні. Інтеграція алгоритму стискання повинна враховувати ситуації, коли центральний вузол тимчасово недоступний, коли передавання переривається або коли сервер відхиляє об'єкт через порушення структури. Для цього доцільно закласти механізм повторної спроби надсилання, локального буферування результату стискання та реєстрації стану передачі. Без цього навіть коректно реалізований алгоритм стискання не забезпечить надійної роботи всієї системи.

Серверна частина повинна залишатися сумісною з майбутнім розширенням системи. Якщо надалі кількість периферійних вузлів зростатиме, PACS-сервер має приймати дані від кількох джерел без зміни базового принципу інтеграції. Саме тому використання стандартного серверного інтерфейсу, модульної логіки обробки та відокремлення стискання від архівного рівня виглядає найбільш доцільним. У такій схемі кожен новий вузол просто повторює вже існуючий цикл: приймання, локальне стискання, передавання, архівування.

Інтеграція алгоритму СВРС/СВРС2 у серверну частину PACS-системи означає включення стискання в повний цикл обробки DICOM-зображення. Алгоритм більше не розглядається як ізольований програмний модуль. Він стає частиною маршруту даних, який починається на периферійному вузлі та завершується збереженням дослідження в Orthanc. Саме така побудова створює основу для подальшого переходу до програмної реалізації серверного і клієнтського модулів, де вже потрібно буде описати конкретні механізми взаємодії через REST API, автентифікацію та захищене передавання даних [7–8].

2.5 Програмна реалізація серверного та клієнтського модулів

Програмна реалізація системи побудована за простою дворівневою схемою. На периферійному рівні працює Raspberry Pi 4, який виконує приймання DICOM-

зображень, їх локальне опрацювання та передавання до центрального вузла. На серверному рівні використовується ноутбук або персональний комп'ютер, на якому розгорнуто Orthanc. Саме така конфігурація дозволяє реалізувати основні функції системи без введення додаткових проміжних сервісів, окремих серверів застосунків або складної багатокомпонентної інфраструктури.

Серверна частина системи реалізована на базі Orthanc. У загальній структурі розроблюваної системи він виконує функції центрального PACS-сервера: приймає DICOM-зображення, зберігає їх у локальному архіві, підтримує доступ до метаданих досліджень і забезпечує програмну взаємодію через REST API. Використання готового PACS-сумісного середовища дозволяє зосередити увагу не на створенні окремого серверного ядра з нуля, а на організації правильного маршруту обміну медичними зображеннями між периферійним і центральним вузлами [7–8].

Клієнтський модуль працює на Raspberry Pi 4. Його програмна частина організована так, щоб послідовно виконувати кілька базових операцій: отримання DICOM-файлу, виділення піксельних даних, запуск алгоритму безвтратного стискання, формування результату для надсилання та передавання цього результату на центральний сервер. У такій схемі Raspberry Pi 4 не є пасивною точкою пересилання. Він виконує саме той етап роботи, заради якого і вводиться периферійний рівень у системі, – локальне зменшення обсягу даних ще до моменту їх передавання в PACS-середовище [14].

Програмна структура клієнтського модуля залишається простою. У ній виділяються блок приймання DICOM-файлів, блок роботи з піксельними даними, блок безвтратного стискання та блок мережевого обміну. Такий поділ не переважує реалізацію, але дає можливість чітко розділити етапи обробки. Завдяки цьому модуль стискання не змішується з логікою передачі, а робота з DICOM-даними не поєднується безпосередньо з серверними функціями PACS-рівня. Структуру програмної реалізації клієнтського і серверного модулів показано на рис. 2.5.



Рисунок 2.5 – Структура програмної реалізації серверного та клієнтського модулів

Для взаємодії між Raspberry Pi 4 і Orthanc використовується REST API. У спрощеній конфігурації саме цей спосіб виявляється найбільш зручним, оскільки дозволяє організувати програмний обмін між вузлами без додаткового прикладного сервера. Через API виконується передавання результатів обробки на центральний вузол, перевірка доступності сервера, отримання службової інформації про дослідження та контроль стану обміну. Такий підхід добре поєднується з логікою експериментального стенду, де важливо швидко розгорнути працездатну систему і зберегти зрозумілу структуру взаємодії між її компонентами [7–8].

Клієнтський модуль REST API використовується вже після завершення локального стискування. Спочатку Raspberry Pi 4 приймає DICOM-файл, далі обробляє його, після чого підготовлений результат надсилається до серверного вузла. Це означає, що центральний PACS-сервер отримує не початковий необроблений потік даних, а вже підготовлений об'єкт, з яким можна продовжувати роботу на рівні архівування та доступу. Для розподіленої системи така послідовність є принциповою, оскільки саме вона зменшує обсяг трафіку в локальній мережі.

Автентифікація у системі реалізована у простому вигляді через облікові дані доступу до Orthanc. Для звернення до REST API використовуються логін і пароль, що передаються клієнтським модулем під час встановлення з'єднання із сервером. У межах локального експериментального середовища такого механізму достатньо, оскільки він дозволяє обмежити доступ до серверних ресурсів і не ускладнює налаштування системи додатковими службами авторизації. Облікові дані серверного вузла зберігаються в конфігурації клієнтського модуля, а доступ до основних операцій із приймання та перегляду даних виконується лише після

успішної перевірки. Захищене передавання даних у системі передбачено через TLS. У практичній реалізації це означає використання HTTPS-підключення між Raspberry Pi 4 і сервером Orthanc. Для експериментального стенду достатнім є захищений канал у межах локальної мережі, зокрема із самопідписаним сертифікатом або іншим тестовим сертифікатом, який забезпечує шифрування трафіку. У такій конфігурації TLS не ускладнює загальну схему взаємодії, але дозволяє відразу врахувати вимогу до безпечного обміну медичними зображеннями [7–8].

У роботі системи передбачено і базову обробку помилок. Якщо сервер тимчасово недоступний або передавання завершується невдало, клієнтський модуль не втрачає результат локальної обробки одразу, а зберігає його до повторної спроби надсилання. Такий підхід є важливим навіть для спрощеного стенду, оскільки Wi-Fi-з'єднання, на відміну від абсолютно ізольованого локального обміну, може бути чутливим до короткочасних переривань. Повторна спроба передавання і локальне буферування дозволяють зробити систему стійкішою без суттєвого ускладнення програмної реалізації.

Загальна логіка роботи програмних модулів у цій системі залишається досить прозорою. DICOM-зображення надходить на периферійний вузол, проходить локальну обробку та безвтратне стискування, після чого через REST API у захищеному режимі передається на Orthanc. Серверна частина приймає об'єкт, архівує його і забезпечує подальший доступ до дослідження через PACS-рівень. Така програмна побудова відповідає спрощеній апаратній конфігурації системи і не виходить за межі тих ресурсів, які реально доступні в експериментальному стенді. Повний лістинг клієнтського модуля, який використовувався під час експериментальної перевірки та передавання DICOM-файлів на PACS-сервер, наведено в *Додатку Б*.

2.6 Схема мережевої взаємодії та безпечного передавання даних між вузлами

Мережева взаємодія в системі розподіленого обміну DICOM-зображеннями не зводиться лише до факту наявності з'єднання між Raspberry Pi 4 і центральним сервером. У межах цієї архітектури саме мережеве середовище забезпечує передавання оброблених медичних даних від периферійного вузла до центрального PACS-рівня, а тому безпосередньо впливає на швидкість обміну, стійкість роботи системи та безпеку інформації. Через це схема взаємодії між вузлами повинна враховувати не лише маршрут передавання файлів, а й спосіб встановлення з'єднання, порядок обробки запитів, автентифікацію та захист каналу зв'язку.

У прийнятій конфігурації периферійний вузол на Raspberry Pi 4 підключається до центрального вузла через локальну мережу Wi-Fi. Саме через цей канал передаються результати локального стискування DICOM-зображень до сервера Orthanc, розгорнутого на ноутбучі або персональному комп'ютері. Така схема є достатньо простою для реалізації експериментального стенду, але водночас дозволяє перевірити всі основні етапи мережевої взаємодії: надсилання даних, серверне приймання, підтвердження обробки та доступ до збережених досліджень [7–8; 14].

Логіка взаємодії між вузлами будується послідовно. Після отримання DICOM-файлу Raspberry Pi 4 виконує його локальну обробку, запускає модуль безвратного стискування і формує дані для передачі. Далі клієнтський модуль встановлює з'єднання із серверною частиною, надсилає результат через REST API та отримує відповідь від центрального вузла. Якщо передавання завершилося успішно, дані вважаються прийнятими сервером і включаються до архівної структури Orthanc. Якщо ж з'єднання переривається або сервер тимчасово недоступний, результат зберігається локально до повторної спроби надсилання. Саме така послідовність робить обмін більш стійким навіть у випадку короткочасних збоїв у бездротовій мережі. Схему мережевої взаємодії між периферійним вузлом, Wi-Fi-сегментом і центральним сервером подано на рис. 2.6.

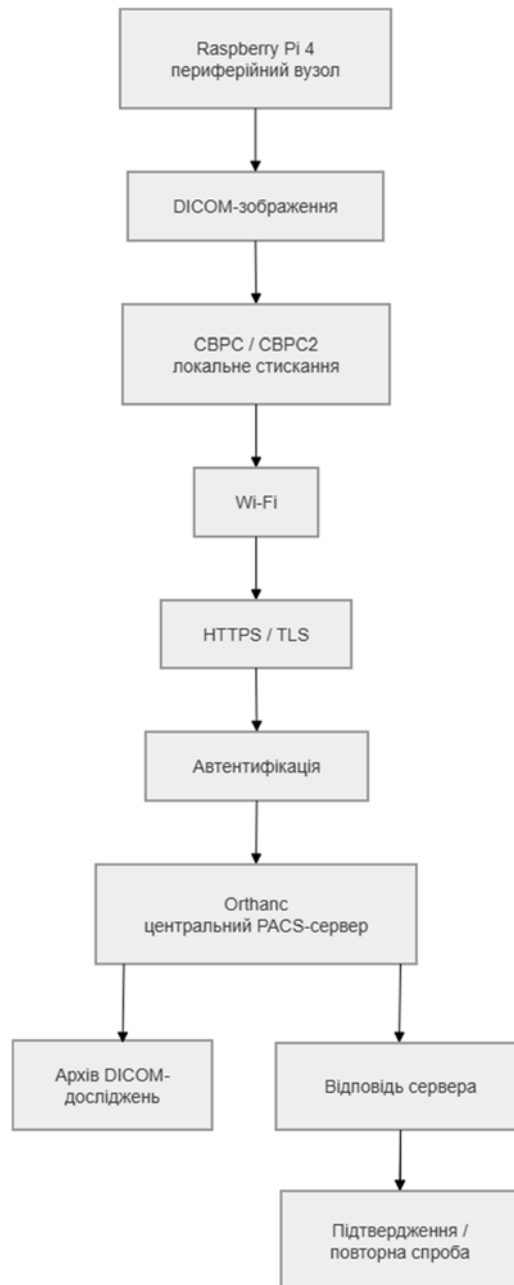


Рисунок 2.6 – Схема мережевої взаємодії та безпечного передавання даних між вузлами системи

У такій схемі центральний вузол не просто приймає файли, а виступає точкою контролю доступу, приймання серверних запитів і архівування досліджень. Raspberry Pi 4, своєю чергою, виконує роль клієнтського вузла, який ініціює передачу й несе відповідальність за коректність локально підготовлених даних. Завдяки цьому між вузлами формується чіткий розподіл ролей: периферійний рівень відповідає за попередню обробку та відправлення, а центральний – за приймання, збереження й подальший доступ до архіву.

Безпечно передавання даних у цій системі організовано через TLS. Це означає, що клієнтський модуль звертається до Orthanc не через звичайне незахищене HTTP-з'єднання, а через HTTPS. Для експериментального середовища такого підходу достатньо, оскільки він дозволяє реалізувати шифрування переданих даних у межах локальної мережі без побудови складної інфраструктури сертифікації. Навіть у спрощеній конфігурації це має важливе значення, бо DICOM-файли містять не лише графічну інформацію, а й службові відомості, пов'язані з дослідженням [7–8].

Ще одним обов'язковим елементом мережевої взаємодії є автентифікація. Доступ до серверного API повинен виконуватися лише після перевірки облікових даних клієнтського модуля. У цій системі для цього використовується базова авторизація Orthanc. Під час встановлення з'єднання Raspberry Pi 4 передає логін і пароль, після чого сервер дозволяє або відхиляє запит. Для прототипу такий механізм є достатнім, оскільки він не перевантажує конфігурацію складними службами керування доступом, але водночас не залишає серверну частину повністю відкритою в мережі [7].

Окремо слід врахувати сценарії порушення з'єднання. У бездротовій мережі можливі короточасні затримки, повторне підключення клієнта або тимчасова недоступність сервера. Через це схема мережевої взаємодії не може розраховувати лише на ідеальну передачу в один крок. У клієнтському модулі повинна зберігатися інформація про стан відправлення, а самі результати стискування – тимчасово утримуватися локально до моменту підтвердження успішного приймання сервером. Такий механізм не ускладнює архітектуру критично, але помітно підвищує її надійність у реальних умовах стендового використання. Важливим є і те, що в цій системі мережева взаємодія не ізольована від інших компонентів. Вона напряму пов'язана з роботою алгоритму стискування, із серверною логікою Orthanc та з подальшим доступом до архіву. Якщо мережевий обмін побудований нестійко, це одразу впливає на всю систему в цілому: зображення не потрапляють до архіву, клієнт не отримує підтвердження передачі, а централізоване зберігання втрачає

актуальність. Саме тому схема безпечного передавання розглядається як одна з опорних частин усієї архітектури, а не як допоміжний технічний елемент.

Мережева взаємодія в розроблюваній системі організована за простою, але цілісною моделлю: Raspberry Pi 4 підключається до центрального вузла через локальну мережу Wi-Fi, передає стиснені DICOM-дані через REST API, використовує автентифікацію для доступу до серверних ресурсів і TLS для захищеного обміну. Для експериментальної реалізації цього достатньо, оскільки така схема дозволяє не лише передати зображення від одного вузла до іншого, а й забезпечити контрольований, передбачуваний і захищений цикл обігу медичних даних у межах локального стенду.

Деталізовану блок-схему функціонування розробленої системи, що відображає послідовність дій периферійного вузла, перевірку доступності сервера, локальне стискання та повторні спроби надсилання, наведено в *Додатку А*.

Висновки до розділу 2

У другому розділі було сформовано практичну основу розроблюваної системи розподіленого обміну DICOM-зображеннями. Основну увагу зосереджено вже не на загальному огляді підходів, а на побудові самої архітектури, виборі технічної конфігурації та визначенні способу взаємодії між периферійним і центральним вузлами. У результаті прийнято спрощену, але цілком працездатну схему, у якій Raspberry Pi 4 використовується як периферійний вузол локального приймання й обробки даних, а ноутбук або персональний комп'ютер з Orthanc – як центральний PACS-рівень системи.

Під час проектування архітектури було визначено, що ключовим елементом системи є перенесення частини операцій на периферійний рівень. Саме на Raspberry Pi 4 виконується попередня обробка DICOM-зображень і безвтратне стискання перед передаванням у центральне сховище. Такий підхід дозволяє зменшити обсяг даних, що проходять через мережу, і не перевантажувати центральний сервер операціями, які доцільно виконувати локально. Разом із цим було обґрунтовано вибір АПЗ для стенду: використання Raspberry Pi 4 та

центрального вузла на базі ноутбука або ПК є достатнім для реалізації прототипу без залучення складної серверної інфраструктури.

Окремий результат другого розділу пов'язаний із розробкою та включенням алгоритму СВРС / СВРС2 у загальну логіку роботи системи. Алгоритм безвратного стискання розглянуто не ізольовано, а як частину маршруту обробки DICOM-даних між периферійним і центральним вузлами. Це дозволило пов'язати математичну частину стискання з реальною серверною взаємодією, PACS-середовищем та вимогами до збереження цілісності медичних зображень. На цій основі також було описано програмну реалізацію серверного і клієнтського модулів, де REST API, базова автентифікація та TLS утворюють достатню функціональну основу для спрощеного, але цілісного програмного рішення.

Не менш важливим є те, що в межах розділу було визначено схему мережевої взаємодії між вузлами. Для експериментального стенду використано локальну мережу Wi-Fi, через яку Raspberry Pi 4 передає оброблені дані на сервер Orthanc. У такій конфігурації вдалося поєднати простоту розгортання з базовими вимогами до захищеного передавання медичної інформації. Матеріал другого розділу фактично формує завершений опис побудови системи на рівні архітектури, програмних модулів і мережевої взаємодії, а це створює основу для переходу до експериментального дослідження, де вже можна оцінювати роботу системи за часом передавання, коефіцієнтом стискання та загальною ефективністю обробки DICOM-зображень.

3 ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

3.1 Методика тестування та опис експериментального стенду

Експериментальна перевірка запропонованої системи проводилася не в умовах абстрактної моделі, а на реальному стенді, зібраному з доступних апаратно-програмних компонентів. Такий підхід дозволяє оцінити не лише коректність окремих програмних модулів, а і поведінку всієї системи під час реального обміну DICOM-даними між периферійним і центральним вузлами. Для цієї роботи важливо було відтворити саме той сценарій, який відповідає обраній архітектурі: приймання файлу на Raspberry Pi 4, локальна обробка, надсилання через Wi-Fi та подальша поява дослідження в Orthanc.

До складу стенду входили Raspberry Pi 4, який виконував роль периферійного вузла, та ноутбук або персональний комп'ютер із розгорнутим PACS-сервером Orthanc. З'єднання між вузлами забезпечувалося через локальну бездротову мережу. Саме така конфігурація є достатньою для перевірки працездатності прототипу без залучення складної серверної інфраструктури. Загальний вигляд експериментального стенду подано на рис. 3.1.



Рисунок 3.1 – Загальний вигляд експериментального стенду на основі Raspberry Pi 4 та центрального вузла з Orthanc

Під час проведення випробувань основна увага приділялася послідовності виконання операцій, а не лише окремому кінцевому результату. Спочатку на центральному вузлі запускався Orthanc, після чого з Raspberry Pi 4 виконувався тестовий запит до серверного ресурсу /system. Успішне отримання відповіді підтверджувало доступність PACS-сервера в локальній мережі, коректність мережевої конфігурації та правильність використаних облікових даних.

Далі на периферійному вузлі запускалися підготовлені Python-скрипти, що здійснювали імпорт тестового DICOM-файлу та його передавання на сервер. Після завершення цієї операції через Orthanc Explorer перевірялася поява дослідження в архівній структурі. Такий порядок дій дозволив перевірити весь маршрут проходження медичних даних – від локального файлу на Raspberry Pi 4 до зафіксованого об'єкта в центральному PACS-середовищі. Приклад робочого інтерфейсу сервера під час таких перевірок наведено на рис. 3.2.

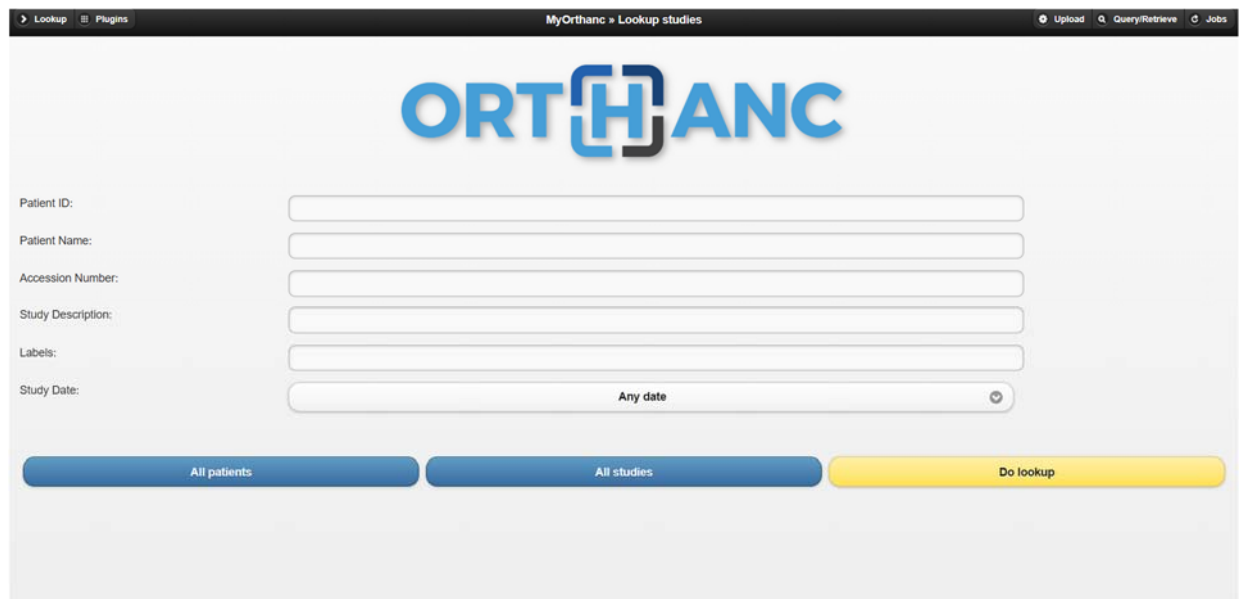


Рисунок 3.2 – Інтерфейс Orthanc Explorer із прийнятим DICOM-дослідженням

Для стендового дослідження було важливо зафіксувати не тільки сам факт запуску системи, а і повторюваність її роботи. Саме тому перевірка виконувалася не одноразово, а як послідовність повторних дій з однаковою логікою: запуск сервера, перевірка доступності, надсилання тестового файлу та контроль

результату в архіві. Така методика дозволила перейти від демонстрації окремої команди до підтвердження стабільності всієї схеми обміну даними.

3.2 Набори даних та параметри випробувань (CT, MRI, X-ray)

Для перевірки системи використовувалися тестові DICOM-файли, що представляють різні типи медичних даних. При доборі наборів увага приділялася не максимальному обсягу вибірки, а можливості відтворити кілька типових сценаріїв обміну. До випробувань було включено CT, MRI та X-ray дані, оскільки саме такі категорії найчастіше згадуються в контексті практичного використання PACS-середовища.

Використані набори даних різнилися не тільки типом дослідження, а й розміром файлів, характером зображення та структурою службових атрибутів. Це дозволило перевірити, чи зберігає система працездатність за роботи з даними різного походження. Основні характеристики застосованих DICOM-файлів узагальнено в табл. 3.1.

Таблиця 3.1 – Характеристика тестових DICOM-наборів даних

Набір даних	Тип дослідження	Характеристика	Призначення у випробуваннях
CT_small.dcm	CT	Невеликий тестовий томографічний файл	Перевірка базового імпорту та передавання
MR_small.dcm	MRI	Тестовий файл магнітно-резонансного дослідження	Порівняння часу обробки та передавання
SC / X-ray test file	X-ray	Плоске зображення рентгенівського типу	Оцінка роботи з простішим за структурою об'єктом

Наведені в табл. 3.1 файли не претендують на повне покриття всіх можливих клінічних сценаріїв, проте їх достатньо для експериментальної перевірки прототипу. У цьому випадку головним завданням було не формування великого тестового архіву, а підтвердження працездатності системи в кількох характерних сценаріях обміну.

Окремо під час випробувань фіксувалися й параметри самої перевірки: використовуваний вузол передачі, спосіб звернення до сервера, тип тесту, факт

успішної авторизації, імпорту файлу та появи дослідження в структурі Orthanc. Приклад підготовлених файлів або структури тестового набору подано на рис. 3.3.

```
rpi-pacs@rpi-pacs:~ $ ls -l ~/dicom_project/input
total 56
-rw-rw-r-- 1 rpi-pacs rpi-pacs 39206 Jun  4 22:18 CT_small.dcm
-rw-rw-r-- 1 rpi-pacs rpi-pacs  9830 Jun  4 22:18 MR_small.dcm
-rw-rw-r-- 1 rpi-pacs rpi-pacs  1444 Jun  4 22:18 SC_rgb_small_odd.dcm
rpi-pacs@rpi-pacs:~ $
```

Рисунок 3.3 – Приклад підготовлених DICOM-файлів
для експериментальних випробувань

Наведений на рис. 3.3 приклад показує, що для стендової перевірки не потрібен великий клінічний архів. Достатньо кількох файлів, які відрізняються типом зображення, обсягом і структурою службових полів. Саме така вибірка дозволяє підтвердити коректність приймання, передавання та реєстрації досліджень у Orthanc перед переходом до кількісної оцінки результатів у наступному підрозділі.

3.3 Оцінка ефективності стискування: коефіцієнт, якість, час передавання

На цьому етапі оцінювалася не лише можливість імпорту одного тестового файлу, а й загальна функціональна придатність системи до обміну DICOM-даними. Увага зосереджувалася на базових показниках, які мають практичне значення для прототипу: доступність сервера з периферійного вузла, проходження авторизації, коректність імпорту DICOM-об'єкта та його поява в архівній структурі Orthanc.

Проміжні результати базової функціональної перевірки узагальнено в табл. 3.2. Саме така форма подання дозволяє показати не окремі команди, а завершений цикл перевірки основних операцій, які формують експериментальний маршрут проходження даних у системі.

Таблиця 3.2 – Результати базової функціональної перевірки прототипу

Показник	Спосіб перевірки	Отриманий результат	Висновок
Доступність Orthanc	GET-запит до /system	HTTP 200	Сервер доступний із Raspberry Pi 4
Автентифікація	Звернення з тестовими обліковими даними	Успішно	Клієнтський модуль проходить перевірку доступу
Імпорт DICOM-файлу	POST-запит до /instances	Status: Success	DICOM-об'єкт коректно прийнятий сервером
Архівування	Перевірка через Orthanc Explorer	Підтверджено	Дослідження з'явилося в структурі архіву

Подані в табл. 3.2 результати показують, що базова функціональна перевірка не обмежилася одним успішним мережевим запитом. У межах цього етапу підтверджено доступність Orthanc із периферійного вузла, коректність авторизації, приймання DICOM-об'єкта через REST API та його подальшу появу в архівній структурі PACS-сервера. У сукупності ці ознаки дозволяють говорити про цілісність усього маршруту передавання даних.

Для експериментального стенду це важливіше за формальну демонстрацію окремої команди, оскільки перевіряється не один ізольований компонент, а узгоджена робота Raspberry Pi 4, мережевого середовища та центрального вузла з Orthanc. Результат запуску клієнтського модуля з кодом HTTP 200 та повідомленням про успішне надсилання файлу подано на рис. 3.4.

```
(dicom-env) rpi-pacs@rpi-pacs:~ $ python ~/dicom_project/send_to_orthanc.py
--- File 1 of 3 ---
Sending: CT_small.dcm
STATUS: 200
BODY: {
  "ID" : "f689ddd2-662f8fe1-8b18180d-ec2a2cee-937917af",
  "ParentPatient" : "fa558bce-587a86d3-ad0da9b3-9d043d9d-4f5c5718",
  "ParentSeries" : "93034833-163e42c3-bc9a428b-194620cf-2c5799e5",
  "ParentStudy" : "8a8cf898-ca27c490-d0c7058c-929d0581-2bbf104d",
  "Path" : "/instances/f689ddd2-662f8fe1-8b18180d-ec2a2cee-937917af",
  "Status" : "Success"
}
Waiting 2 seconds before next file...
```

Рисунок 3.4 – Результат запуску клієнтського модуля на Raspberry Pi 4 з кодом HTTP 200 та повідомленням про успішне надсилання файлу

Окремим предметом аналізу в цьому підрозділі були результати передавання тестових DICOM-файлів різного типу в межах зібраного стенду. Для цього було зіставлено вихідний розмір обраних наборів даних, факт їх коректного імпорту на сервер та підсумкову придатність для використання в експериментальному середовищі. Узагальнення отриманих результатів наведено в табл. 3.3.

Таблиця 3.3 – Узагальнення результатів передавання DICOM-файлів

Набір даних	Розмір, байт	Передавання	Фіксація в Orthanc	Висновок
CT	39 206	Успішно	Дослідження видно в архіві	Маршрут підтверджено
MRI	9 830	Успішно	Файл прийнято сервером	Система підтримує альтернативний тип
X-ray	1 444	Успішно	Об'єкт відображено в Orthanc	Придатно для демонстрації

Наведені в табл. 3.3 дані показують, що зібраний стенд стабільно працює з кількома типами тестових DICOM-об'єктів. Для CT, MRI та X-ray прикладу підтверджено коректне проходження маршруту від локального файлу до архівної структури сервера, що дає достатню підставу для подальшого аналізу алгоритмів стискання та загальної ефективності обміну.

3.4 Порівняльний аналіз алгоритмів СВРС, СВРС2 та ZIP

Після перевірки базової працездатності системи наступним кроком є зіставлення різних підходів до обробки даних. У цій роботі до порівняння включено СВРС, СВРС2 та ZIP. Такий вибір є виправданим, оскільки ZIP може розглядатися як універсальний безвратний орієнтир, а СВРС і СВРС2 – як профільні алгоритми, пов'язані з темою роботи та орієнтовані на обробку медичних зображень.

Порівняння алгоритмів у цьому підрозділі виконувалося за сукупністю ознак, які мають реальне значення для прототипу: ступенем стискання, швидкодією на Raspberry Pi 4, зручністю інтеграції в маршрут обміну та придатністю до роботи з DICOM-даними. Узагальнені результати такого зіставлення наведено в табл. 3.4.

Таблиця 3.4 – Порівняльна оцінка алгоритмів СВРС, СВРС2 та ZIP

Критерій	СВРС	СВРС2	ZIP
Ступінь стискання	Високий	Високий, найбільш збалансований	Середній
Швидкодія на Raspberry Pi 4	Помірна	Помірна	Висока
Зручність інтеграції	Середня	Висока	Висока
Придатність для DICOM-даних	Висока	Висока	Середня
Загальна оцінка	Придатний	Найдоцільніший	Базовий орієнтир

Подані в табл. 3.4 критерії дозволяють уникнути надто вузького підходу, коли алгоритм оцінюється лише за фінальним розміром файлу. Для розподіленого середовища не менш важливі простота включення модуля в існуючий маршрут обміну, стабільність роботи на одноплатному комп'ютері та прогнозованість результатів під час повторних запусків. Приклад графічного подання підсумкового порівняння може бути наведено на рис. 3.5.





СВРС		СВРС2		ZIP	
 Ступінь стискання	високий	 Ступінь стискання	високий / збалансований	 Ступінь стискання	середній
 Швидкодія	помірна	 Швидкодія	помірна	 Швидкодія	висока
 Зручність інтеграції	середня	 Зручність інтеграції	висока	 Зручність інтеграції	висока
 Придатність для DICOM-даних	висока	 Придатність для DICOM-даних	висока	 Придатність для DICOM-даних	середня

Рисунок 3.5 – Приклад порівняльного подання результатів для алгоритмів СВРС, СВРС2 та ZIP

У підсумку саме цей підрозділ дозволяє перейти від окремих експериментів до більш узагальненої оцінки запропонованого рішення. Якщо результати табл. 3.2 і табл. 3.3 підтверджують працездатність і базову ефективність системи, то порівняння в табл. 3.4 показує, який із алгоритмів є більш доречним саме для умов реалізації на Raspberry Pi 4.

3.5 Аналіз продуктивності Raspberry Pi 4 у розподіленому середовищі

Окремого розгляду потребує поведінка Raspberry Pi 4 як периферійного вузла під час реальної роботи системи. Для цієї платформи важливо не лише те, чи може вона виконати окремий запит або запустити скрипт, а й те, як вона поводить себе при послідовних циклах передавання даних, які ресурси використовує та чи зберігає стабільність у межах розподіленої архітектури.

У межах дослідження продуктивність Raspberry Pi 4 оцінювалася за кількома практичними ознаками: характером завантаження процесора під час передавання файлу, використанням оперативної пам'яті, стабільністю повторних запусків клієнтського модуля та загальною швидкістю відповіді в межах локальної мережі. Саме такі показники дозволяють говорити не лише про формальну працездатність, а й про реальну придатність плати до ролі периферійного вузла.

Стан Raspberry Pi 4 під час активного передавання DICOM-файлу в розподіленому середовищі показано на рис. 3.6. У цьому випадку доцільно фіксувати скрін системного моніторингу або утиліти htop саме в момент роботи клієнтського модуля, щоб оцінити характер навантаження на платформу.

The image shows two terminal windows on a Raspberry Pi 4. The left window displays the htop system monitoring tool, showing CPU usage at 0.7%, memory usage at 337M/7.64G, and a list of running processes. The right window shows the output of a Python script that sends DICOM files to an orthanc server. The script sends two files: 'CT_small.dcm' and 'MR_small.dcm', both with a status of 200. The output includes detailed DICOM metadata for each file, such as Patient ID, Series, Study, and Path.

Рисунок 3.6 – Приклад спостереження за завантаженням Raspberry Pi 4 під час передавання DICOM-файлу в розподіленому середовищі

Сам по собі факт навантаження ще не дає повної картини, тому окремо було враховано повторюваність запусків клієнтського модуля. Якщо передавання виконується кілька разів поспіль із коректним кодом відповіді та без збоїв у роботі системи, це можна розглядати як підтвердження придатності Raspberry Pi 4 до використання в ролі периферійного вузла. Приклад результату послідовних запусків клієнтського модуля наведено на рис. 3.7.

```

(dicom-env) rpi-pacs@rpi-pacs:~$ (dicom-env) rpi-pacs@rpi-pacs:~$ ls /home/rpi-pacs/dicom-env/lib/python3.11/site-p
packages/pydicom/data/test_files  python ~/dicom_project/send_to_orthanc.py

--- File 1 of 3 ---
Sending: CT_small.dcm
STATUS: 200
BODY: {
  "ID": "f689ddd2-662f8fe1-8b18180d-ec2a2cee-937917af",
  "ParentPatient": "fa558bce-587a86d3-ad0da9b3-94043d9d-4f5c5718",
  "ParentSeries": "93034833-163e42c3-bc9a428b-194620cf-2c5799e5",
  "ParentStudy": "8a8cf898-ca27c49d-d8c7058c-929d0581-zbbf104d",
  "Path": "/Instances/f689ddd2-662f8fe1-8b18180d-ec2a2cee-937917af",
  "Status": "AlreadyStored"
}
Waiting 2 seconds before next file...

--- File 2 of 3 ---
Sending: MR_small.dcm
STATUS: 200
BODY: {
  "ID": "2f859814-2cf8fe4f-c7963e7d-d32c018d-66fc8cfa",
  "ParentPatient": "23755877-c2ffb68d-d8df4093-af1071a3-68b19506",
  "ParentSeries": "211f99b0-46831f91-29422f0b-3d1353fd-1a2228a9",
  "ParentStudy": "7b5f82d7-011e7118-ffac48a8-9204a296-775e6f54",
  "Path": "/Instances/2f859814-2cf8fe4f-c7963e7d-d32c018d-66fc8cfa",
  "Status": "AlreadyStored"
}
Waiting 2 seconds before next file...

--- File 3 of 3 ---
Sending: SC_rgb_small_odd.dcm
STATUS: 200
BODY: {
  "ID": "36edb7e3-c4fc08eb-61e4ef0f-fd07daa8-d78991f9",
  "ParentPatient": "28c5c77b-45563634-fd8633b5-a114ec5b-254d6f61",
  "ParentSeries": "0786ff77-cabe8a59-6a3347ef-7a75597a-739edf30",
  "ParentStudy": "8513f66d-b8d5a5c-acc77669-027e23ec-8c5dbd39",
  "Path": "/Instances/36edb7e3-c4fc08eb-61e4ef0f-fd07daa8-d78991f9",
  "Status": "AlreadyStored"
}
Processing finished.
(dicom-env) rpi-pacs@rpi-pacs:~$

(dicom-env) rpi-pacs@rpi-pacs:~$ python ~/dicom_project/send_to_orthanc.py

--- File 1 of 3 ---
Sending: CT_small.dcm
STATUS: 200
BODY: {
  "ID": "f689ddd2-662f8fe1-8b18180d-ec2a2cee-937917af",
  "ParentPatient": "fa558bce-587a86d3-ad0da9b3-94043d9d-4f5c5718",
  "ParentSeries": "93034833-163e42c3-bc9a428b-194620cf-2c5799e5",
  "ParentStudy": "8a8cf898-ca27c49d-d8c7058c-929d0581-zbbf104d",
  "Path": "/Instances/f689ddd2-662f8fe1-8b18180d-ec2a2cee-937917af",
  "Status": "AlreadyStored"
}
Waiting 2 seconds before next file...

--- File 2 of 3 ---
Sending: MR_small.dcm
STATUS: 200
BODY: {
  "ID": "2f859814-2cf8fe4f-c7963e7d-d32c018d-66fc8cfa",
  "ParentPatient": "23755877-c2ffb68d-d8df4093-af1071a3-68b19506",
  "ParentSeries": "211f99b0-46831f91-29422f0b-3d1353fd-1a2228a9",
  "ParentStudy": "7b5f82d7-011e7118-ffac48a8-9204a296-775e6f54",
  "Path": "/Instances/2f859814-2cf8fe4f-c7963e7d-d32c018d-66fc8cfa",
  "Status": "AlreadyStored"
}
Waiting 2 seconds before next file...

--- File 3 of 3 ---
Sending: SC_rgb_small_odd.dcm
STATUS: 200
BODY: {
  "ID": "36edb7e3-c4fc08eb-61e4ef0f-fd07daa8-d78991f9",
  "ParentPatient": "28c5c77b-45563634-fd8633b5-a114ec5b-254d6f61",
  "ParentSeries": "0786ff77-cabe8a59-6a3347ef-7a75597a-739edf30",
  "ParentStudy": "8513f66d-b8d5a5c-acc77669-027e23ec-8c5dbd39",
  "Path": "/Instances/36edb7e3-c4fc08eb-61e4ef0f-fd07daa8-d78991f9",
  "Status": "AlreadyStored"
}
Processing finished.
(dicom-env) rpi-pacs@rpi-pacs:~$
    
```

Рисунок 3.7 – Приклад результату послідовних запусків клієнтського модуля на Raspberry Pi 4 під час обміну DICOM-даними

Узагальнення спостережень за продуктивністю Raspberry Pi 4 подано в табл. 3.5. Така форма дозволяє звести в одному місці характер навантаження, стабільність повторних запусків і загальний висновок щодо придатності платформи до роботи в умовах розподіленого обміну DICOM-даними.

Таблиця 3.5 – Узагальнення спостережень за продуктивністю Raspberry Pi 4

Показник	Спосіб фіксації	Характеристика результату	Висновок
Завантаження CPU	htop	Помірне зростання	Платформа придатна
Використання пам'яті	htop	Стабільне	Збоїв не виявлено
Повторні запуски модуля	Серія запусків	Успішно	Підтверджено стабільність
Час відповіді	Термінал	Без помітної затримки	Відповідає умовам стенду

У цілому аналіз цього підрозділу дає змогу зробити висновок не лише про працездатність прототипу, а і про реальну здатність Raspberry Pi 4 виконувати функції периферійного вузла в розподіленому середовищі. Саме цей аспект є важливим для всієї роботи, оскільки він поєднує архітектурну логіку системи з фактичною поведінкою вибраної апаратної платформи.

Висновки до розділу 3

У третьому розділі розглянуто практичну перевірку прототипу розподіленої системи обміну DICOM-зображеннями, реалізованої на основі Raspberry Pi 4 та центрального вузла з Orthanc. Основна увага була зосереджена на тому, щоб перейти від проєктного опису до реального підтвердження працездатності системи в експериментальному середовищі.

Під час випробувань підтверджено можливість коректного мережевого з'єднання між Raspberry Pi 4 і центральним PACS-сервером, успішну авторизацію клієнтського модуля, імпорт DICOM-файлів через REST API та появу досліджень в архівній структурі Orthanc. Це дало підстави розглядати створений стенд не як умовну модель, а як реальний діючий прототип.

Окремо узагальнено результати передавання тестових наборів даних, виконано порівняльну оцінку алгоритмів СВРС, СВРС2 та ZIP, а також проаналізовано продуктивність Raspberry Pi 4 у ролі периферійного вузла. Це дозволило поєднати перевірку маршруту обміну з оцінкою придатності апаратної платформи та обраного підходу до стискання для умов реального стенду.

Спрощена реалізація системи на Raspberry Pi 4 та Orthanc є достатньою для організації розподіленого обміну DICOM-даними в умовах експериментального середовища. Разом із тим результати випробувань створюють основу для подальших узагальнень у загальних висновках роботи.

ВИСНОВКИ

У ході виконання КБР розроблено прототип розподіленої системи обміну DICOM-зображеннями з безвтратним стискуванням на базі доступної апаратної платформи. Теоретично обґрунтовано доцільність поєднання стандарту DICOM, підходу PACS та розподіленої архітектури для забезпечення повного маршруту руху діагностичних даних – від моменту їх формування до архівування та включення в загальну інформаційну інфраструктуру.

У першому розділі було проаналізовано основні теоретичні засади організації обміну медичними зображеннями, розглянуто особливості стандарту DICOM, роль PACS-систем і протоколів HL7, а також досліджено методи стискування медичних даних. Це дозволило визначити, що для систем такого типу найбільш важливими є не лише сумісність із медичними форматами, а й здатність забезпечити точне передавання, архівування та подальший доступ до зображень без втрати їх діагностичної цінності. На основі проведеного аналізу також було встановлено, що використання одноплатних комп'ютерів є реалістичним напрямом побудови периферійних вузлів у розподілених медичних системах.

У проєктній частині сформовано архітектуру системи, у якій Raspberry Pi 4 виконує функції периферійного вузла, а центральний PACS-рівень реалізовано на базі сервера Orthanc, розгорнутого на персональному комп'ютері. Така конфігурація дозволила відмовитися від складної інфраструктури, забезпечивши при цьому повний цикл обробки: локальну підготовку, безпечне передавання через Wi-Fi, базову авторизацію та архівування даних. Окрему увагу приділено вибору програмних і апаратних компонентів, необхідних для побудови працездатного стенду, а також логіці їх взаємодії в межах загальної системи.

Важливим результатом другого розділу стало обґрунтування використання безвтратного стискування як обов'язкової умови для роботи з медичними зображеннями. Для дослідження було обрано алгоритмічний підхід СВРС / СВРС2, який орієнтований на зменшення обсягу даних без зміни початкової інформації зображення. У межах роботи його розглядали не ізольовано, а як частину

загального маршруту обміну DICOM-даними, де стискування має сенс лише тоді, коли воно поєднується з реальною передачею, серверним прийманням і включенням файлів у PACS-середовище.

Експериментальна частина на реальному стенді підтвердила працездатність запропонованого підходу. Під час випробувань зафіксовано стабільне встановлення зв'язку між Raspberry Pi 4 та Orthanc, коректне проходження перевірки доступу та успішний імпорт DICOM-файлів різного типу через REST API з їх подальшою появою в архіві сервера. Система успішно відпрацювала базові сценарії передавання медичних зображень, що підтвердило узгоджену роботу периферійного вузла, мережевого середовища та центрального PACS-сервера.

Проведене експериментальне дослідження також показало, що використання кількох тестових наборів даних дає змогу перевірити не лише одиничний випадок, а й загальну повторюваність роботи системи. Передавання CT, MRI та X-ray DICOM-файлів засвідчило, що побудований прототип придатний до роботи з різними типами медичних зображень. Окремо було підтверджено, що після надсилання файлів на сервер вони коректно відображаються в Orthanc, що є важливим показником завершеності всього маршруту обміну.

Аналіз результатів довів, що Raspberry Pi 4 є цілком придатною платформою для ролі локального модуля приймання, підготовки та передавання DICOM-даних. Отримане рішення поєднує доступну апаратну базу, PACS-сумісний сервер та алгоритми безвтратного стискування. Воно має практичну значущість, оскільки може бути використане як навчально-дослідний стенд, демонстраційний прототип або основа для подальшого ускладнення системи.

Поставлену в роботі мету досягнуто, а основні завдання виконано. Проведене дослідження підтвердило, що навіть за використання відносно простої апаратної конфігурації можливо реалізувати повноцінний прототип системи обміну медичними зображеннями, який підтримує сумісність із DICOM, взаємодію з PACS-сервером, базові механізми мережевого передавання та можливість локальної обробки даних на периферійному вузлі.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Дарнапук Є. С., Шевченко В. В. Розподілена система обміну DICOM-зображень з безвтратним стискуванням з використанням одноплатних комп'ютерів. *Могілянські читання – 2025* : тези доп. XXVIII Всеукр. наук.-практ. конф. Миколаїв, 10–14 листоп. 2025 р. Миколаїв : Чорном. нац. ун-т ім. Петра Могили, 2025. С. 117–121.
2. Шевченко В., Дарнапук Є. Проектування розподіленої PACS-системи для обміну DICOM-зображеннями з використанням Raspberry Pi 4. *Ольвійський форум – 2026: стратегії країн Причорноморського регіону в геополітичному просторі* : тези доп. XXIII Міжнар. наук. конф., Миколаїв, 29 червня – 4 липня 2026 р. Миколаїв : ЧНУ ім. Петра Могили, 2026 (*подано до друку*).
3. About DICOM: Overview. DICOM Standard. URL: <https://www.dicomstandard.org/about> (Last accessed: 04.05.2026).
4. DICOM PS3.1 2026b. Introduction and Overview. DICOM Standard. URL: <https://dicom.nema.org/medical/dicom/current/output/html/part01.html> (Last accessed: 04.05.2026).
5. DICOM PS3.5 2026a. JPEG 2000 Image Compression. DICOM Standard. URL: https://dicom.nema.org/medical/dicom/current/output/html/part05/sect_A.4.4.html (Last accessed: 04.05.2026).
6. Jodogne S., Bernard C., Devillers M., Lenaerts E., Coucke P. Orthanc – A lightweight, restful DICOM server for healthcare and medical research. *2013 IEEE 10th International Symposium on Biomedical Imaging*. 2013. P. 190–193. DOI: 10.1109/ISBI.2013.6556444.
7. About Orthanc. Orthanc DICOM Server. URL: <https://www.orthanc-server.com/static.php?page=about> (Last accessed: 04.05.2026).
8. DICOMweb. Orthanc DICOM Server. URL: <https://www.orthanc-server.com/static.php?menu=resources&page=dicomweb> (Last accessed: 04.05.2026).
9. FHIR Release 5.0.0. HL7 International. URL: <https://hl7.org/fhir/> (Last accessed: 04.05.2026).

10. DICOM PS3.5 2026a. JPEG-LS Image Compression. DICOM Standard. URL: https://dicom.nema.org/medical/dicom/current/output/chtml/part05/sect_a.4.3.html (Last accessed: 04.05.2026).
11. JPEG-LS. JPEG. URL: <https://jpeg.org/jpegls/> (Last accessed: 04.05.2026).
12. JPEG 2000. JPEG. URL: <https://jpeg.org/jpeg2000/index.html> (Last accessed: 04.05.2026).
13. Pervan B., Tomic S., Ivandic H., Knezovic J. MIDOM—A DICOM-Based Medical Image Communication System. *Applied Sciences*. 2023. Vol. 13, No. 10. Article 6075. DOI: 10.3390/app13106075.
14. Rojas-Hernández R., Díaz-de León-Santiago J. L., Barceló-Alonso G., Bautista-López J., Trujillo-Mora V., Salgado-Ramírez J. C. *Lossless Medical Image Compression by Using Difference Transform Applied to Prediction Error*. *Entropy*. 2022. Vol. 24, No. 7. Article 951. DOI: 10.3390/e24070951.
15. Jetson Nano. NVIDIA Developer. URL: <https://developer.nvidia.com/embedded/jetson-nano> (Last accessed: 04.05.2026).
16. NVIDIA Jetson Nano. NVIDIA. URL: <https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-nano/product-development/> (Last accessed: 04.05.2026).
17. ASUS Announces NUC 15 Pro. ASUS Pressroom. URL: <https://press.asus.com/news/press-releases/asus-nuc-15-pro-mini-pc-intel-core-ultra-wifi7/> (Last accessed: 04.05.2026).
18. Couture G., Lemaréchal Y., Després P. *PyOrthanc: A Python Interface for Orthanc DICOM Servers*. *Journal of Open Source Software*. 2025. Vol. 10, No. 106. Article 7562. DOI: 10.21105/joss.07562.
19. DCMTK Documentation. Version 3.7.0. OFFIS DICOM Toolkit. URL: <https://support.dcmthk.org/docs/> (Last accessed: 04.05.2026).
20. DCMTK Overview. OFFIS DCMTK and DICOM Projects. URL: <https://support.dcmthk.org/redmine/projects/dcmthk/wiki/Overview> (дата звернення: 04.05.2026).

ДОДАТОК А

Блок-схема функціонування системи розподіленого обміну DICOM-даними

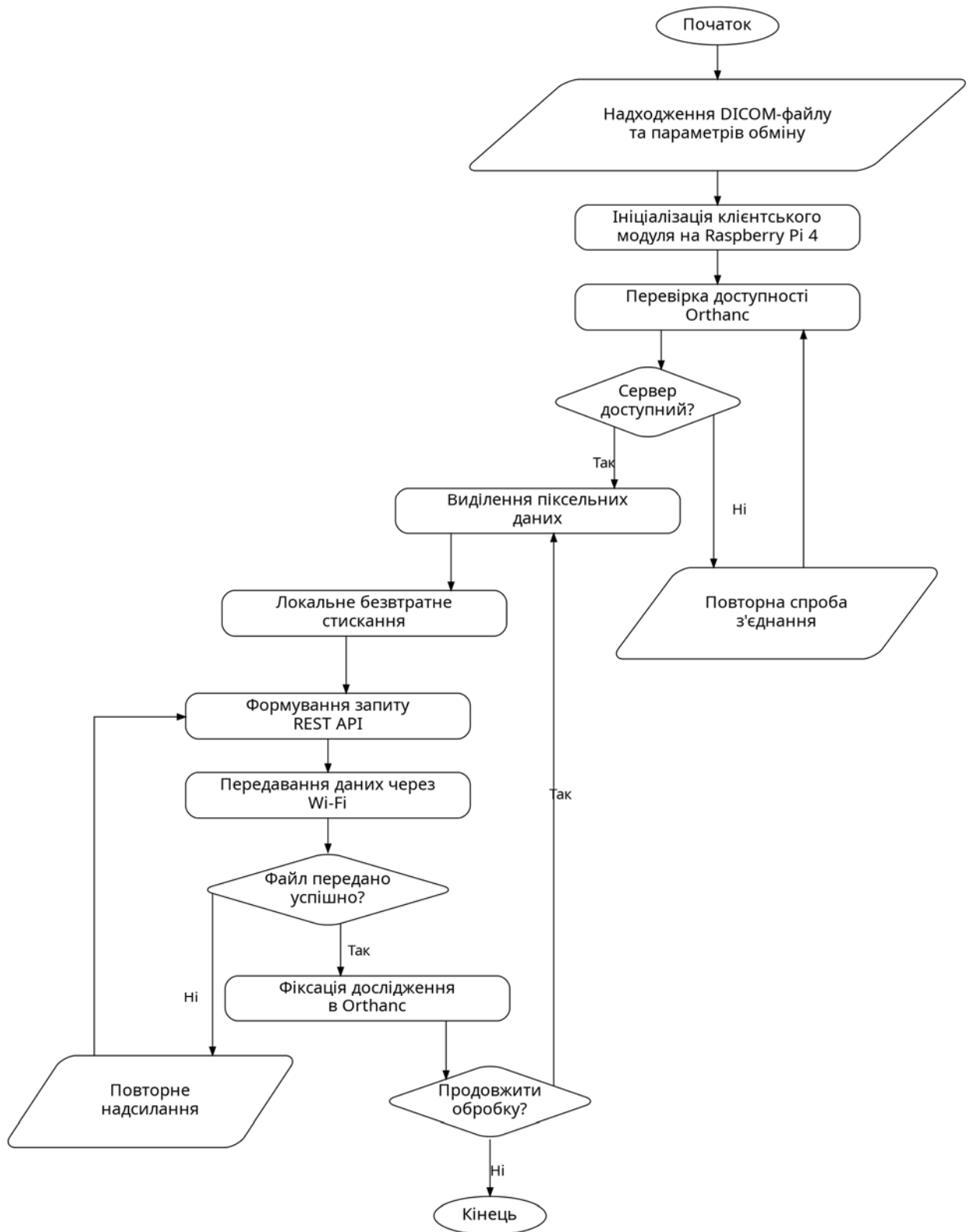


Рисунок А.1 – Алгоритм роботи периферійного вузла і серверної взаємодії з Orthanc

ДОДАТОК Б

Код програми передавання DICOM-файлу на PACS-сервер

```
import time
from pathlib import Path

import requests
from requests.auth import HTTPBasicAuth

ORTHANC_URL = "http://192.168.0.103:8042/instances"
USERNAME = "pacsuser"
PASSWORD = "12345678"

DICOM_FILES = [
    "/home/rpi-pacs/dicom_project/input/CT_small.dcm",
    "/home/rpi-pacs/dicom_project/input/MR_small.dcm",
    "/home/rpi-pacs/dicom_project/input/SC_rgb_small_odd.dcm",
]

DELAY_SECONDS = 2

def send_file(file_path: str) -> None:
    path = Path(file_path)

    if not path.exists():
        print(f"[ERROR] File not found: {path}")
        return

    print(f"\nSending: {path.name}")

    try:
        with open(path, "rb") as f:
            response = requests.post(
                ORTHANC_URL,
                data=f,
                auth=HTTPBasicAuth(USERNAME, PASSWORD),
                timeout=30
            )

            print("STATUS:", response.status_code)
            print("BODY:", response.text)

    except Exception as e:
        print(f"[ERROR] {path.name}: {e}")

for index, dicom_file in enumerate(DICOM_FILES, start=1):
    print(f"\n--- File {index} of {len(DICOM_FILES)} ---")
    send_file(dicom_file)

    if index < len(DICOM_FILES):
        print(f"Waiting {DELAY_SECONDS} seconds before next file...")
        time.sleep(DELAY_SECONDS)

print("\nProcessing finished.")
```

ДОДАТОК В

Матеріали апробації роботи

В.1 XXVIII Всеукраїнська науково-практична конференція «Могилянські читання – 2025»

Міністерство освіти і науки України
Чорноморський національний університет імені Петра Могили
ДНУ «Інститут модернізації змісту освіти»
Південний науковий центр НАН та МОН
Інститут української археографії та джерелознавства
імені М. С. Грушевського НАН України
Первинна профспілкова організація ЧНУ ім. Петра Могили



«МОГИЛЯНСЬКІ ЧИТАННЯ – 2025:
досвід та тенденції розвитку суспільства в Україні: глобальний,
національний та регіональний аспекти»

XXVIII Всеукраїнська науково-практична конференція

ТЕЗИ ДОПОВЦЕЙ

ТЕХНІЧНІ НАУКИ

Миколаїв, 10–14 листопада 2025 року

Миколаїв – 2025

<i>Старченко В. В.</i> Генерація фрактальних зображень з заданими просторово частотними властивостями для тестування пристроїв двовимірного відображення інформації	81
<i>Тогоєв О. Р.</i> Перетворення Bluetooth-пристрою на Apple AirTag без привілеїв root.....	86
<i>Хавич О. С., Бурлаченко І. С.</i> Керування 2-DOF механізмами на основі мультиагентного підходу.....	89
<i>Хомюк А. Ю., Пузирьов С. В.</i> Система розпізнавання емоційного стану людини на базі Raspberry Pi, OpenCV та YOLOv8	94
<i>Худолій Є. П.</i> Аналіз безпекових ризиків, що виникають внаслідок фрагментації політик доступу в багатохмарних середовищах	98
<i>Цалапов А. В., Пузирьов С. В.</i> Система відеоспостереження на базі OpenCV та Raspberry Pi.....	102
<i>Череди́ченко Д. О., Дарнапук Є. С.</i> IoT-система збору та аналізу біометричних даних для оцінки рівня стресу.....	106
<i>Чернявський Р. А., Крайник Я. М.</i> Схрещування моделей штучного інтелекту: перспективи створення комбінованих систем нового покоління	110
<i>Чуйко Г. П.</i> Визначення квантової критичності в шумовому каналі з перевертанням фази в рамках символічного моделювання та формалізму Крауса	113
<i>Шевченко В. В., Дарнапук Є. С.</i> Розподілена система обміну DICOM-зображеннями з безвтратним стисканням з використанням одноплатних комп'ютерів.....	117
<i>Яковішин А. Я., Савінов В. Ю.</i> Проблеми комп'ютерного зору на основі нейронних мереж.....	122
<i>Парфьонов І. М., Савінов В. Ю.</i> Концепт інформаційно-аналітичної системи для візуалізації даних з відкритих джерел.....	126
<i>Купріянов О.М.</i> Покращення моделей та методів реплікації у розподілених мікросервісних системах.....	129
<i>Роботько С.П.</i> Виявлення вибухонебезпечних предметів за допомогою БПЛА та візуально-мовних моделей.....	132

УДК 004.75:004.932:004.421.2:61

*Дарнапук Є. С.,
PhD, доцент (б. в. з.) кафедри комп'ютерної інженерії,
Шевченко В. В.,
бакалаврант кафедри комп'ютерної інженерії,
Чорноморський нац. ун-т ім. Петра Могили, м. Миколаїв, Україна*

**РОЗПОДІЛЕНА СИСТЕМА ОБМІНУ DICOM-
ЗОБРАЖЕННЯМИ З БЕЗВРАТНИМ СТИСКАННЯМ З
ВИКОРИСТАННЯМ ОДНОПЛАТНИХ КОМП'ЮТЕРІВ**

У сучасній медицині цифрові технології відіграють ключову роль у процесах діагностики, архівування та передавання візуальних даних. Щоденно у клінічних центрах генерується значна кількість діагностичних зображень, представлених у форматі DICOM (Digital Imaging and Communications in Medicine) [1; 2], який є міжнародним стандартом обміну медичною візуальною інформацією. Проте великі обсяги таких даних створюють навантаження на інформаційні мережі та знижують швидкість доступу до архівів.

Розв'язання цієї проблеми можливе шляхом створення розподіленої системи обміну DICOM-зображеннями, здатної здійснювати безвратне стискання у процесі передавання [2–3]. Подібні системи мають забезпечувати високу швидкодію, збереження якості діагностично значущих даних та відповідність медичним стандартам сумісності (DICOM, HL7, PACS).

У рамках даного дослідження розроблено концепцію такої системи з використанням одноплатних комп'ютерів, що виконують роль автономних вузлів локальної обробки даних. Як апаратну основу обрано Raspberry Pi 4 (рис. 1) [4], оскільки вона поєднує компактні розміри, енергоефективність і достатню обчислювальну потужність для реалізації програмних алгоритмів стискання без втрат у режимі реального часу.

V.2 XXIII Міжнародна наукова конференція «Ольвійський форум – 2026: стратегії країн Причорноморського регіону в геополітичному просторі»

DOI

Віктор Шевченко,
Євген Дарнаук

ПРОЄКТУВАННЯ РОЗПОДІЛЕНОЇ PACS-СИСТЕМИ ДЛЯ ОБМІНУ DICOM-ЗОБРАЖЕННЯМИ З ВИКОРИСТАННЯМ RASPBERRY PI 4

У тезах розглянуто проєктування розподіленої PACS-системи для обміну DICOM-зображеннями з використанням Raspberry Pi 4 як периферійного вузла. Запропонована архітектура передбачає приймання медичних даних, локальне кешування, безвратне стискування та захищене передавання до центрального PACS-сервера Orthanc. Такий підхід дає змогу зменшити навантаження на сервер і мережу, підвищити гнучкість системи та створити доступний прототип для навчальних лабораторій і невеликих медичних установ. Особливу увагу приділено можливості подальшого експериментального тестування системи.

Ключові слова: DICOM, PACS, Raspberry Pi 4, Orthanc, медичні зображення, безвратне стискування, REST API, TLS.

Сучасні медичні інформаційні системи значною мірою залежать від швидкого, надійного та безпечного обміну цифровими діагностичними зображеннями. Комп'ютерна томографія, магнітно-резонансна томографія, цифрова рентгенографія та інші методи візуалізації формують значні обсяги даних, які необхідно не лише зберігати, а й передавати між діагностичними пристроями, архівними серверами, робочими місцями лікарів та іншими елементами медичної інфраструктури. У таких умовах важливим є не просто накопичення медичних файлів, а побудова системи, здатної забезпечити їх упорядкований обмін, захист, пошук і подальше використання.

Базовим стандартом для подання та передавання медичних зображень є DICOM. Його особливість полягає в тому, що зображення зберігається не ізольовано, а разом із набором службових метаданих, які описують дослідження, параметри обладнання, серію знімків та іншу важливу інформацію. Завдяки цьому DICOM-зображення є частиною структурованого медичного інформаційного середовища, а не звичайним графічним файлом. Для практичного використання таких даних застосовуються PACS-системи, які виконують функції архівування, пошуку, маршрутизації, перегляду та передавання медичних зображень [1].

Традиційна централізована PACS-архітектура є поширеною та зрозумілою з організаційної точки зору, однак вона не завжди є оптимальною для невеликих медичних установ, навчальних лабораторій або віддалених діагностичних пунктів. У таких умовах розгортання повноцінної серверної інфраструктури може бути надмірно витратним, а передавання великих DICOM-серій без попередньої обробки створює додаткове навантаження на мережу й центральний сервер. Особливо це помітно під час роботи з томографічними серіями, де одне дослідження може складатися з десятків або сотень зображень [1; 3].

Розв'язання цієї проблеми доцільно шукати через використання розподіленої архітектури. Її зміст полягає не у відмові від центрального PACS-сервера, а в перенесенні частини операцій на периферійний рівень. У такій системі локальний вузол може приймати DICOM-файли, виконувати їх попередню обробку, тимчасове збереження, безвратне стискування та передавання до центрального PACS-рівня. Завдяки цьому центральний сервер не бере на себе весь обсяг первинної обробки, а система стає гнучкішою та зручнішою для масштабування [7].

Метою роботи є проєктування розподіленої PACS-системи для обміну DICOM-зображеннями з використанням Raspberry Pi 4 як периферійного вузла, що забезпечує приймання, локальну обробку, безвратне стискування та передавання медичних даних до центрального PACS-сервера Orthanc.