НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет (ННІ) ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

ПОГОДЖЕНО	**ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ**
Декан факультету (Директор ННІ)	**Завідувач кафедри**
Інформаційних технологій (назва факультету(ННІ))	Комп'ютерних систем, мереж та кібербезпеки (назва кафедри)
_____ Болбот І.М., д.т.н, проф. (підпис) (ПІБ, вчене звання і ступінь)	_____ Касаткін Д.Ю., к. пед.н., доц. (підпис) (ПІБ, вчене звання і ступінь)
«___»_____2025 р.	«___»_____2025 р.

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Технології NAT у побудові захищених автономних систем»

Спеціальність 123 «Комп’ютерна інженерія»
(код і найменування)

Освітня програма Комп’ютерні системи та мережі
(назва)

Орієнтація освітньої програми Освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Гарант освітньої програми
д.т.н., доцент _____ Мамченко С.М.
(науковий ступінь та вчене звання) (підпис) (ПІБ)

Керівник магістерської кваліфікаційної роботи
д.т.н., доцент _____ Мамченко С.М.
(науковий ступінь та вчене звання) (підпис) (ПІБ)

Виконав _____ Чеботарьов А.В
(підпис) (ПІБ)

КИЇВ-2025

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет (ННІ) ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

ЗАТВЕРДЖУЮ
Завідувач кафедри
комп'ютерних систем, мереж та кібербезпеки
к.пед.н., доц. _____ Касаткін Д.Ю.
(вчене звання і ступінь) (підпис) (ПІБ)
«__» _____ 20__ р.

З А В Д А Н Н Я

ДО ВИКОНАННЯ МАГІСТЕРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ ЗДОБУВАЧУ

Чеботарьову Андрію Вікторовичу
(прізвище, ім'я, по батькові)

Спеціальність 123 «Комп'ютерна інженерія»
(код і найменування)

Освітня програма Комп'ютерні системи та мережі
(назва)

Орієнтація освітньої програми Освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Тема магістерської кваліфікаційної роботи: «Технології NAT у побудові захищених автономних систем»

затверджена наказом ректора НУБіП України від "29" жовтня 2024р. № 1941 «С»

Термін подання завершеної роботи на кафедру 14 листопада 2025 р.

Вихідні дані до магістерської кваліфікаційної роботи

Перелік питань, що підлягають дослідженню:
1.
2.
3.

Перелік графічного матеріалу (за потреби)

Дата видачі завдання " 29 **"** жовтня 2024 р.

Керівник магістерської кваліфікаційної роботи _____ Мамченко С.М.
(підпис) (прізвище та ініціали)

Завдання прийняв до виконання _____ Чеботарьов А.В.
(підпис) (прізвище та ініціали)

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

1. ACL – Access Control List, список контролю доступу
2. API – Application Programming Interface, інтерфейс прикладного програмування
3. CPU – Central Processing Unit, центральний процесор
4. DMZ – Demilitarized Zone, демілітаризована зона
5. DNAT – Destination Network Address Translation, трансляція адреси призначення
6. DoS – Denial of Service, відмова в обслуговуванні
7. gRPC – Remote Procedure Call Framework, протокол викликів віддалених процедур
8. HTTPS – Hypertext Transfer Protocol Secure, захищений протокол передачі даних
9. INET – Internet Segment, зовнішній мережевий сегмент
10. IPSec – Internet Protocol Security, протокол захисту мережевих з'єднань
11. MQTT – Message Queuing Telemetry Transport, протокол телеметрії
12. NAT – Network Address Translation, трансляція мережевих адрес
13. PAT – Port Address Translation, трансляція адрес із підстановкою порту
14. PLC – Programmable Logic Controller, програмований логічний контролер
15. OT – Operational Technology, технологічний сегмент мережі
16. SNAT – Source Network Address Translation, трансляція адреси джерела
17. SQL – Structured Query Language, мова структурованих запитів
18. SSH – Secure Shell, захищений мережевий протокол
19. Syslog – System Logging Protocol, протокол системного журналювання
20. UI – User Interface, користувацький інтерфейс
21. VPN – Virtual Private Network, віртуальна приватна мережа
22. WAN – Wide Area Network, глобальна мережа

ВСТУП

У сучасних мережевих архітектурах, що характеризуються високим рівнем децентралізації, автономності вузлів та підвищеними вимогами до безпеки, технології трансляції мережевих адрес (Network Address Translation, NAT) відіграють ключову роль у побудові захищених середовищ взаємодії. NAT забезпечує перетворення адрес між внутрішніми приватними сегментами та зовнішніми мережами, дозволяючи реалізувати контрольований обмін даними, фільтрацію пакетів, приховування топології та запобігання несанкціонованому доступу. Завдяки цим властивостям NAT є базовим механізмом формування захищених автономних систем, що функціонують у корпоративних, промислових і критичних інформаційних інфраструктурах.

З технічної точки зору, застосування NAT у таких середовищах дозволяє не лише розширювати обмежений адресний простір IPv4, але й виконувати динамічну маршрутизацію, сегментацію мережі, балансування навантажень і глибоку інспекцію трафіку (DPI). Це особливо актуально для систем, у яких передача даних відбувається між автономними вузлами - шлюзами, контролерами, серверами й клієнтськими застосунками, що мають різні рівні довіри та ізоляції. У таких системах NAT виконує функцію посереднього рівня безпеки, який приховує внутрішні ресурси, мінімізує ризики атак типу spoofing *і* port scanning, а також забезпечує узгодженість протоколів взаємодії.

Актуальність теми визначається тим, що в умовах гібридних і автономних мереж із підвищеними вимогами до конфіденційності даних класичні підходи до NAT не завжди забезпечують належний рівень безпеки та масштабованості. Потребують дослідження оптимізовані моделі трансляції адрес, здатні адаптуватися до динамічних мережевих топологій, підтримувати інтеграцію з VPN, Firewall і IDS/IPS-системами, а також гарантувати стійкість до мережевих атак при мінімальних затримках трафіку.

Мета роботи - розробити архітектуру та програмну реалізацію моделі NAT у контексті створення захищеної автономної системи, яка забезпечує керовану маршрутизацію, ізоляцію трафіку та стійкість до зовнішніх загроз.

Для досягнення мети необхідно розв'язати такі **завдання**:

1. провести систематизацію видів NAT (Static, Dynamic, PAT, Bidirectional, Hairpin) та оцінити їхню придатність для автономних систем.
2. Проаналізувати механізми взаємодії NAT із протоколами TCP/IP, VPN, Firewall, IDS/IPS.
3. Розробити архітектурну модель NAT-підсистеми із включенням модулів обробки запитів, таблиць трансляції та моніторингу трафіку.
4. Реалізувати прототип системи у середовищі Java Swing, що забезпечує симуляцію NAT-процесів, візуалізацію таблиць маршрутизації та аналіз потоків даних.
5. Провести тестування продуктивності, стійкості та рівня захисту запропонованої моделі.

Об'єкт дослідження - процес побудови та функціонування захищених автономних систем із використанням технологій NAT.

Предмет дослідження - методи трансляції, маршрутизації та захисту мережевого трафіку у багаторівневих системах із розподіленими вузлами.

Методи дослідження - аналітичний, моделювання та експериментальний. У процесі реалізації застосовуються технології Java Swing**,** Java Networking API**,** Packet Analyzer Libraries**,** SQLite для логування трансляцій і JUnit для модульного тестування.

Наукова новизна полягає у створенні узагальненої моделі функціонування NAT у складі захищених автономних систем із можливістю динамічної адаптації до змін топології та інтеграції з компонентами мережевої безпеки. Реалізація системи на Java забезпечить платформну незалежність,

наочну візуалізацію процесів трансляції та можливість інтеграції у реальні корпоративні середовища.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Опис предметної області та основних процесів функціонування системи

Предметна область технологій NAT у побудові захищених автономних систем охоплює архітектуру мережевих інфраструктур, у яких здійснюється трансляція приватних адрес внутрішніх вузлів у публічний простір задля забезпечення безпеки, масштабованості та контролю трафіку. Такі системи є базовими для сучасних корпоративних, промислових та IoT-рішень, де велика кількість пристроїв потребує централізованого доступу до зовнішніх сервісів без прямого розкриття своєї адресації. Механізми NAT (Network Address Translation) дозволяють вирішити цю задачу, забезпечуючи трансляцію приватних IP-адрес у публічні (SNAT, DNAT, PAT), а також застосування політик доступу через ACL-списки і фаєрволи, що істотно підвищує рівень ізоляції внутрішнього середовища від зовнішніх загроз. Загальна структура предметної області представлена на рис. 1.1.

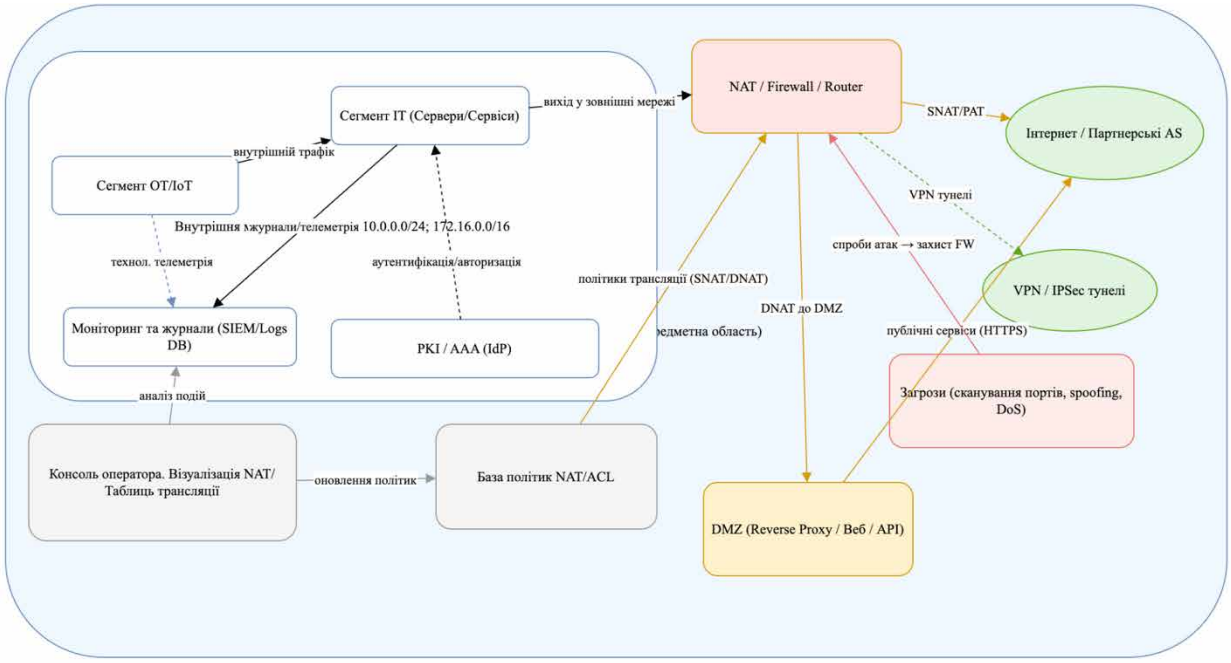


Рис. 1.1 – Структура предметної області та основних взаємодій системи NAT у захищеній автономній архітектурі

Вона включає сегментовану внутрішню мережу (IT-та OT/IoT-зони), модулі NAT/Firewall, DMZ-зону з проксі-сервісами, зовнішній інтернет-сегмент та елементи взаємодії з оператором через Java Swing-консоль. Кожен з елементів системи виконує певну роль у забезпеченні цілісності обміну даними: OT-сегмент формує телеметрію та запити до внутрішніх серверів, IT-сегмент виконує обчислення, зберігає дані та керує політиками безпеки, а NAT-шлюз здійснює динамічну трансляцію адрес, веде журнали та ініціює VPN/IPSec-тунелі для захищеного міжмережевого обміну. Зовнішні підсистеми представлені інтернет-провайдерами, партнерськими автономними системами, а також загрозами у вигляді спроб сканування портів, spoofing-атак та DoS-трафіку.

У таблиці 1.1 подано структуровану характеристику основних компонентів, що беруть участь у функціонуванні предметної області.

Таблиця 1.1 – Основні компоненти та процеси предметної області

№	Компонент	Функціональне призначення	Основні процеси
1	OT/IoT-сегмент	Збір телеметрії, передача виміряних параметрів до серверів	Формування вхідного трафіку, технол. повідомлення
2	IT-сегмент	Обробка запитів, ведення БД, контроль доступу	Аутентифікація, маршрутизація, аудит
3	NAT/Firewall	Маскування IP-адрес, фільтрація, переадресація	SNAT, DNAT, PAT, перевірка політик ACL
4	DMZ-зона	Ізоляція публічних ресурсів, веб/API-доступ	Обробка запитів від зовнішніх клієнтів
5	VPN/IPSec тунелі	Захищене з’єднання між автономними системами	Шифрування, маршрутизація з сертифікатами
6	PKI/AAA-модуль	Сертифікація, аутентифікація користувачів	Генерація токенів, перевірка прав доступу
7	Консоль оператора (Java Swing)	Моніторинг NAT-таблиць, візуалізація сесій, зміна політик	Інтерактивне керування та аналіз трафіку

Предметна область охоплює повний цикл взаємодії між внутрішніми вузлами, шлюзом трансляції та зовнішнім середовищем, забезпечуючи контрольовану маршрутизацію, безпечну інтеграцію і централізований моніторинг. Розроблювана система має на меті дослідження алгоритмів NAT-трансляції, аналіз таблиць сесій і реалізацію програмного середовища для симуляції захищеної автономної мережі з використанням Java Swing-інтерфейсу та вбудованих засобів журналювання.

1.2 Теоретико-методологічні засади та стан наукових досліджень

Сучасні дослідження у сфері побудови захищених автономних систем із використанням технологій NAT (Network Address Translation) спрямовані на подолання обмежень класичної IPv4-адресації, підвищення стійкості до мережевих атак і забезпечення масштабованості багатосегментних архітектур. Теоретичну основу становлять моделі трансляції адрес SNAT, DNAT, PAT, CGN (Carrier-Grade NAT), а також протоколи NAT-Traversal, які дозволяють встановлювати захищені сеанси між вузлами, розташованими за фаєрволами або маршрутизаторами з приватною адресацією [1], [2].

Науковці виділяють три базові рівні реалізації NAT-механізмів: локальний рівень абонентського обладнання, операторський рівень (CGN) та міжавтономний рівень, де здійснюється повторна трансляція трафіку при взаємодії автономних систем. Узагальнена структура таких сценаріїв наведена на рис. 1.2, який відображає порівняння моделей одношарового та дворівневого NAT відповідно до стандартів IETF IMC [3].

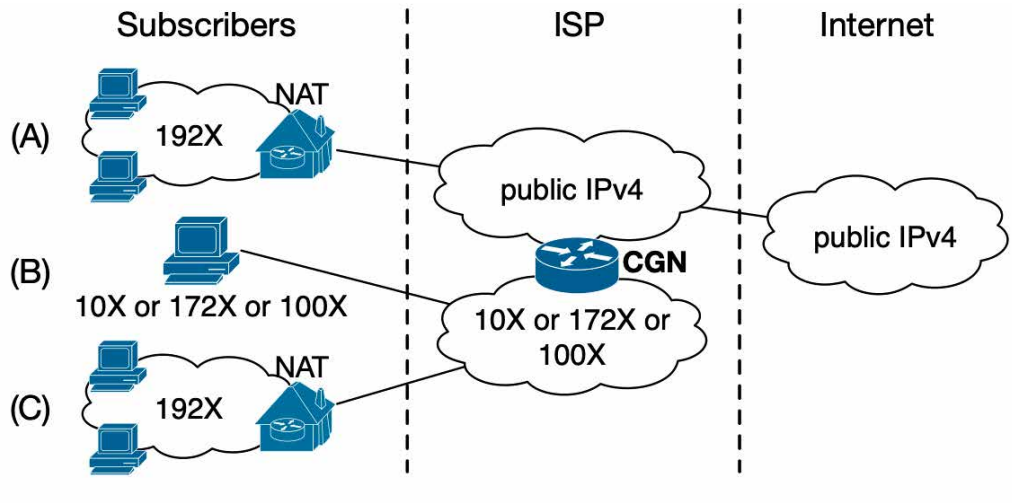


Рис. 1.2 – Типові схеми NAT і CGN-архітектур (за IMC'16 A Multi-Perspective Analysis of Carrier-Grade NAT Deployment [3])

У контексті взаємодії користувачів через мережі з NAT-захистом актуальним є застосування протоколів STUN, TURN і ICE, які забезпечують обхід трансляції адрес і підтримують безпосередні P2P-з'єднання, необхідні для мультимедійних сервісів (WebRTC, SIP). На рис. 1.3 показано принцип роботи STUN-сервера, який визначає публічну адресу клієнта, встановлює канал зв'язку між вузлами через NAT і підтримує передачу аудіо- та відеоданих у зашифрованому вигляді [4].

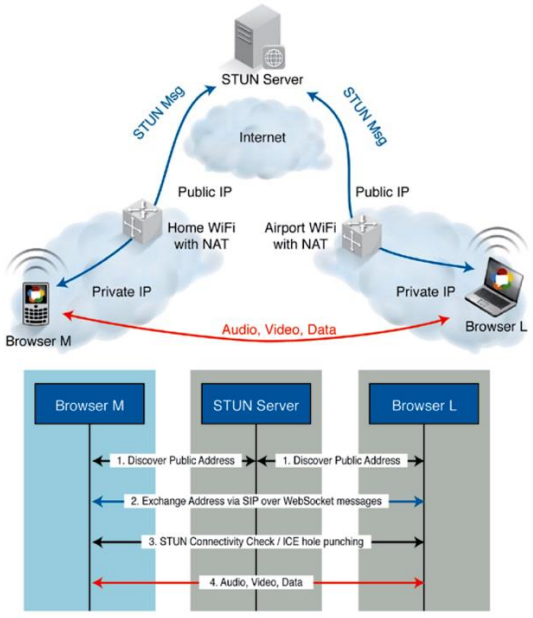


Рис. 1.3 – Схема взаємодії клієнтів через STUN-сервер у середовищі NAT (WebRTC architecture [4])

Іншим напрямом наукових досліджень є узгодження NAT із криптографічними протоколами, зокрема IPSec, що вимагає підтримки NAT-Traversal (NAT-T). На рис. 1.4 показано процес інкапсуляції ESP-пакетів у UDP/4500 для одночасного з'єднання декількох VPN-клієнтів через один зовнішній IP-адрес, що забезпечує сумісність захищених тунелів у мережах NAT [5].

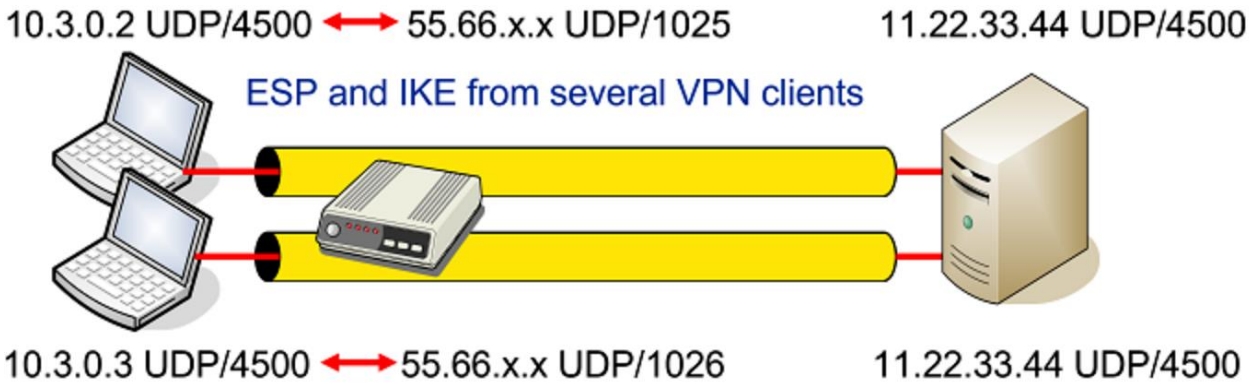


Рис. 1.4 – Механізм NAT-Traversal для VPN/IPSec-з'єднань (за strongSwan Docs [5])

Для мультимедійних застосувань, таких як VoIP та WebRTC, широко досліджується взаємодія протоколів сигналізації (SIP), STUN-запитів і медіа-потоків (рис. 1.5). У цій моделі STUN сервіс виступає посередником, який пробиває "дірку" в NAT та дозволяє прямий обмін медіаданими між вузлами [6].

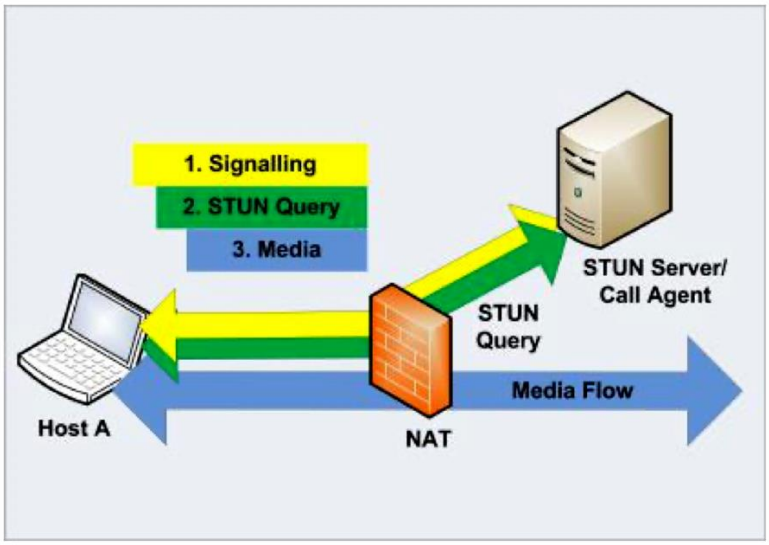


Рис. 1.5 – Процес STUN-запитів і передачі медіа-потоків у мережі з NAT (за IJCA Survey [6])

Останні дослідження зосереджуються на моделях інтелектуального аналізу та адаптації NAT-таблиць, де застосовуються алгоритми машинного навчання для виявлення аномалій у трафіку, динамічного перерозподілу портів та оптимізації QoS у сегментованих мережах. Ці підходи формують методологічну основу розробки нашої системи, у якій NAT-функціонал поєднується з автоматизованим керуванням та моніторингом через Java Swing інтерфейс, а також із інтеграцією VPN та PKI-модулів для захищеного взаємного доступу.

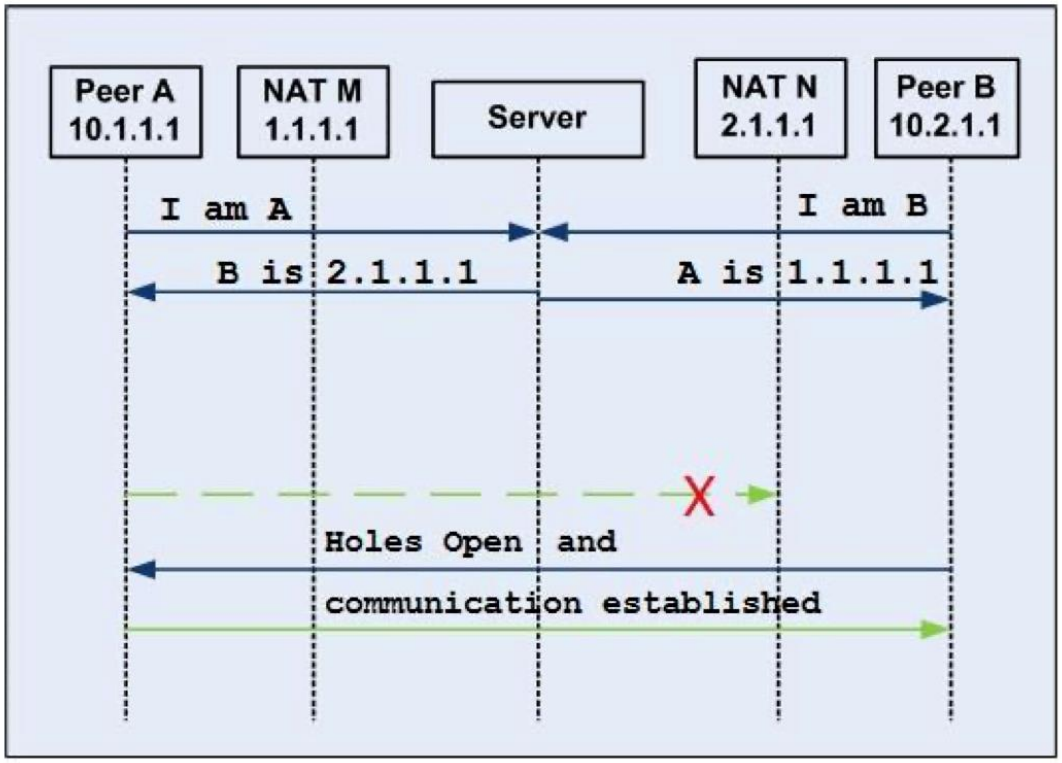


Рис. 1.6 – Схема взаємодії Peer-to-Peer через сервер посередник у умовах NAT та Firewall (за RFC 5128 [7])

Класичні механізми адресної трансляції (SNAT, DNAT, PAT, CGN) із сучасними підходами до обходу NAT-бар'єрів (STUN/TURN/ICE) та криптографічними тунельними протоколами (NAT-Traversal для IPSec/VPN). На відміну від існуючих рішень, запропонована модель орієнтована не лише на маршрутизацію та ізоляцію мереж, а й на адаптивне управління політиками трансляції з урахуванням телеметрії та аналітичних метрик безпеки.

Практична реалізація наукової ідеї полягає у створенні програмного середовища моделювання NAT-процесів у Java Swing, що дозволить досліджувати зміну таблиць трансляції, динаміку портів, затримки трафіку й реакцію системи на атаки (сканування, DoS, spoofing). Крім того, передбачається інтеграція інтелектуальних алгоритмів виявлення аномалій та оптимізації NAT-таблиць для підвищення ефективності використання ресурсів автономної системи.

У результаті дослідження буде розроблено уніфіковану архітектурну модель, яка забезпечить стійку роботу автономних систем у багаторівневому середовищі NAT, узгодження політик безпеки між внутрішніми сегментами, VPN-тунелями й зовнішніми автономними доменами, що становить новий етап розвитку підходів до безпечного маршрутизаційного середовища в умовах дефіциту IPv4-адрес і кіберзагроз.

1.3 Аналіз існуючих рішень

На сучасному етапі розвитку комп'ютерних мереж реалізація NAT-технологій є базовим механізмом для забезпечення масштабованості IPv4-адресного простору, побудови безпечних автономних систем та ізоляції внутрішніх сегментів. Серед провідних підходів виділяють Carrier-Grade NAT (CGN/LSN), корпоративний NAT-шлюз із комбінованими SNAT/DNAT політиками, провайдерські NAT44-архітектури, а також інтегровані NAT-платформи з системами кореляції журналів і AAA-аутентифікацією. Нижче наведено огляд найбільш показових рішень.

Першим прикладом є архітектура CGN/LSN (Carrier-Grade NAT), що використовується у великих провайдерських середовищах для одночасного обслуговування великої кількості приватних клієнтських адрес через спільний публічний пул (рис. 1.7). Вона дозволяє створювати дворівневі NAT-ланцюги (NAT444) і забезпечує масштабування IPv4-простору завдяки динамічній трансляції портів [1].

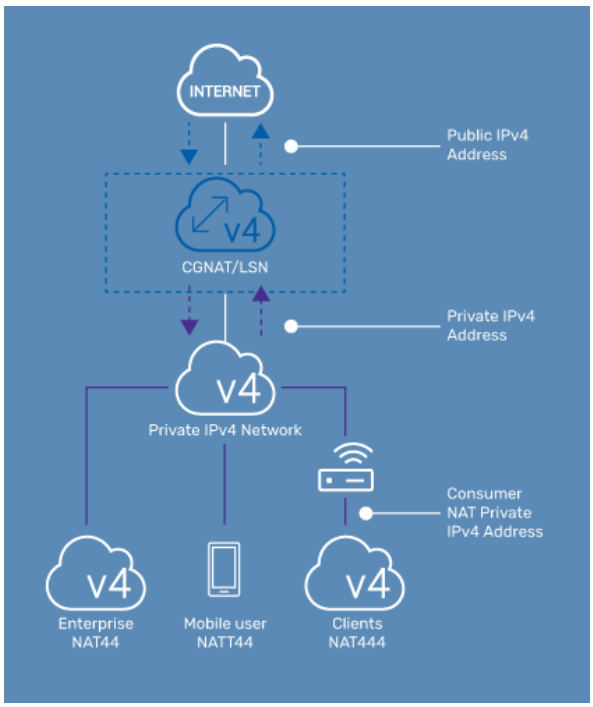


Рис. 1.7 – Архітектура Carrier-Grade NAT/LSN для багаторівневих IPv4-мереж (A10 Networks CGNAT Deployment Guide)

Другим поширеним рішенням є Source-Destination NAT у корпоративних сегментах, яке поєднує вихідну й зворотну трансляцію для керування внутрішнім і зовнішнім трафіком (рис. 1.8). Такі рішення реалізовані у системах Palo Alto NGFW, FortiGate та Cisco Firepower, що забезпечують одночасний контроль SNAT, DNAT і PAT політик, а також їхню інтеграцію з VLAN-сегментацією та фаєрволами рівня 7 [2].

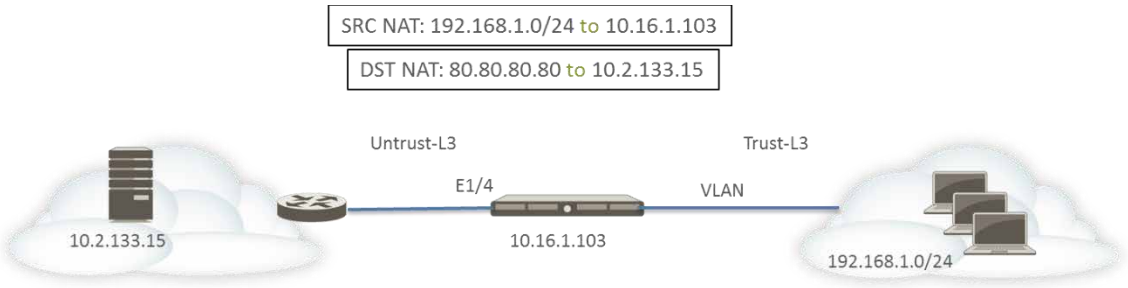


Рис. 1.8 – Приклад одночасної SNAT та DNAT трансляції у корпоративному NAT-шлюзі (Palo Alto Networks NAT Example)

У середовищах провайдерів середнього рівня активно застосовується NAT44-CPE/CGN модель, що відповідає рекомендаціям RFC 6598 (рис. 1.9). Вона передбачає використання приватних адрес RFC 1918 у клієнтських доменах та їх повторну трансляцію у публічні адреси CGN-рівня

ISP. Такий підхід підвищує гнучкість керування абонентським трафіком, але ускладнює трасування інцидентів і аудит [3].

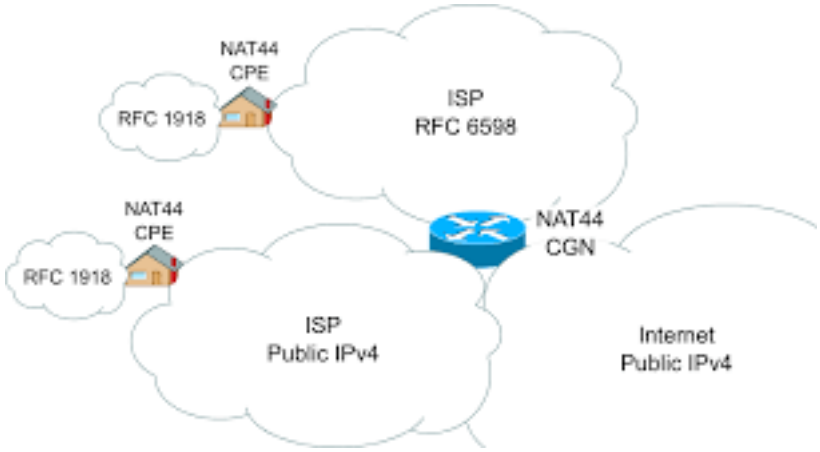


Рис. 1.9 – NAT44 CPE/CGN модель з подвійною трансляцією (згідно з RFC 6598 та Cisco CGN Deployment)

Четвертим прикладом є інтегрована система NAT із журналюванням та AAA-кореляцією, де NAT-транзакції відстежуються у реальному часі через API та веб-інтерфейс (рис. 1.10). Такі системи реалізовані у рішеннях Fortinet Carrier-Grade NAT та F5 CGNAT, що підтримують кореляцію NAT-таблиць із AAA-записами користувачів і забезпечують зворотне відновлення ланцюгів SNAT-з'єднань [4].

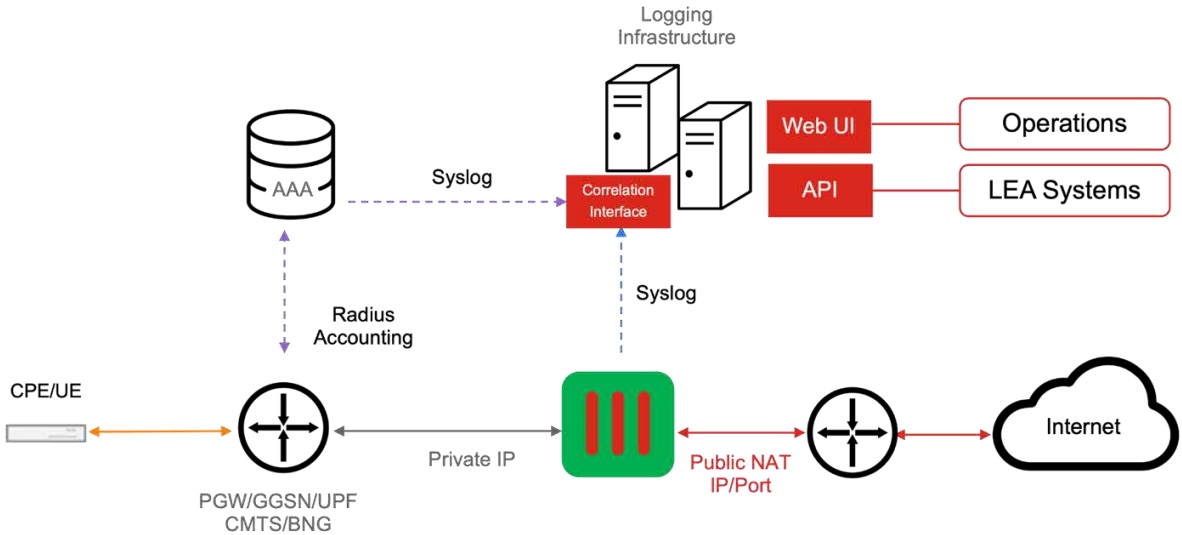


Рис. 1.10 – Архітектура інтегрованого CGNAT з AAA та журналюванням (Fortinet Carrier-Grade NAT Architecture Guide)

П'ятий напрям - хмарні керовані NAT шлюзи (AWS NAT Gateway, Azure NAT Gateway, Google Cloud NAT), які дозволяють централізовано керувати

вихідними підключеннями автономних кластерів без необхідності розгортання власної інфраструктури. Їхня перевага полягає у масштабованості та простоті інтеграції з VPN, але вони мають обмеження щодо L7-контролю та залежність від постачальника хмарних послуг [5].

У таблиці 1.2 наведено порівняльний аналіз розглянутих рішень і місце нашої розробки в контексті наявних технологій.

Таблиця 1.2 – Порівняльний аналіз існуючих рішень та розроблюваної системи

№	Рішення	Основні характеристики	Переваги	Обмеження	Сумісність із автономними системами
1	A10 CGN/LSN	Масштабний CGNAT, динамічне керування портами, моніторинг	Висока продуктивність, телеметрія	Висока вартість, складне впровадження	Так (провайдерський рівень)
2	Palo Alto NGFW	SNAT/DNAT/PAT, інтеграція з L7 фаєрволом	Глибока фільтрація, VPN, IPS	Ліцензійні витрати	Так (корпоративний рівень)

Продовження таблиці 1.2

3	Cisco NAT44/CGN	Дворівнева NAT (локальна та CGN)	Надійність, відповідність RFC 6598	Високі вимоги до логування	Так (ISP та корпоративний рівень)
4	Fortinet/F5 CGNAT	Кореляція NAT-таблиць та AAA, журналювання через API	Прозорий моніторинг, автоматизація	Висока ресурсоємність	Так (операторський та корпоративний)
5	AWS/Azure Cloud NAT	Хмарний керований NAT, API та моніторинг	Масштабованість, простота розгортання	Vendor lock-in, обмежений контроль	Так (гібридні системи)
6	Наша розробка (Java Swing NAT Lab)	Моделювання таблиць SNAT/DNAT,	Освітньо-дослідницький	Невелика продуктивність у	Так (лабораторні та

		візуалізація, аналітика трафіку	інструмент, адаптація для IoT/OT	реальному середовищі	демонстр аційні AS)

Результати аналізу показують, що існуючі рішення орієнтовані на практичне використання у провайдерських чи корпоративних мережах, але більшість із них закриті та недоступні для гнучкого наукового моделювання. Запропонована нами система має наукову цінність через можливість імітації NAT-таблиць, візуалізації зв'язків та аналізу анонімізації трафіку у реальному часі, що сприяє розвитку методів побудови захищених автономних мереж та адаптивних механізмів адресної трансляції.

1.4 Структурне представлення та принципи роботи системи

Структурна модель розробленої системи (рис. 1.11) відображає логічну організацію компонентів NAT-інфраструктури, яка поєднує функції трансляції адрес, управління політиками, шифрування трафіку та моніторингу подій у межах захищеної автономної системи. Центральним елементом архітектури є ядро NAT/Firewall, що реалізує модулі SNAT, DNAT, PAT/NAPT, а також допоміжні сервіси ALG (SIP/FTP), Hairpin NAT, і NAT-T (IPSec) - забезпечуючи як прямий, так і зворотний напрямок маршрутизації та тунелювання трафіку між внутрішніми й зовнішніми мережами. Система підтримує динамічну побудову NAT-таблиць і автоматичне очищення таймаутів, що підвищує ефективність використання пулів адрес та мінімізує ризики конфліктів з'єднань.

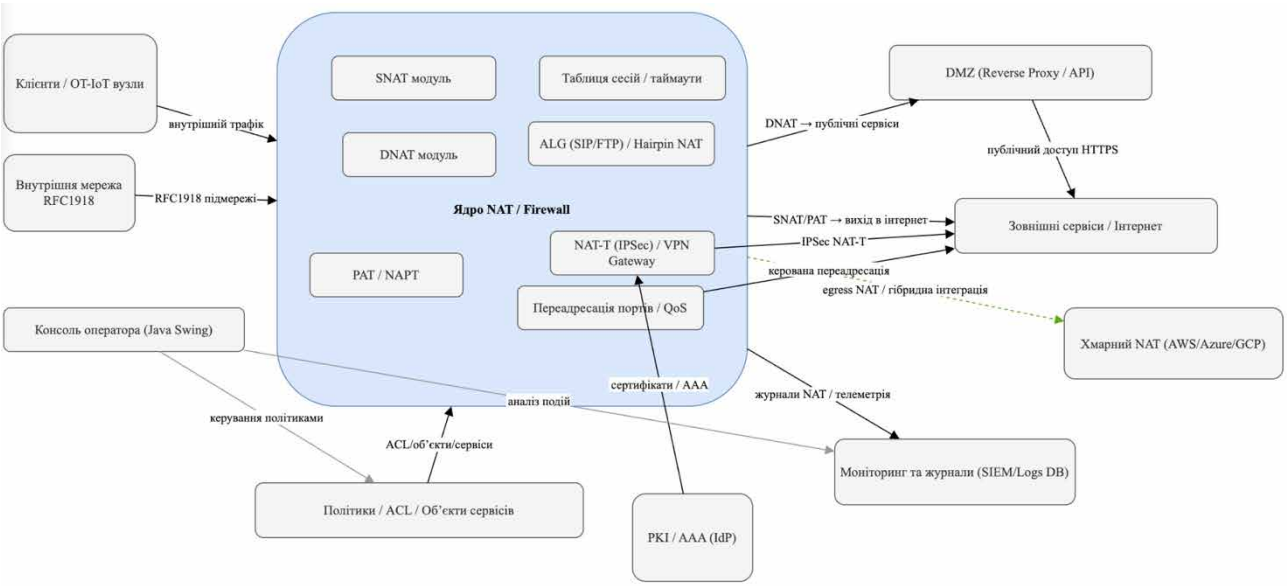


Рис. 1.11 – Структурна схема системи NAT у захищеній автономній мережі

Взаємодія компонентів побудована за принципом подієво-орієнтованої обробки, коли консоль оператора (реалізована на Java Swing) слугує інтерфейсом для керування політиками трансляції, збору телеметрії та аналізу подій. Модулі політик - ACL, об'єкти сервісів, PKI/AAA (IdP) - виконують функції автентифікації, авторизації та сертифікаційного контролю, синхронізуючись із ядром через API. Підсистема журналювання (SIEM/Logs DB) отримує записи з NAT-ядра та забезпечує аудит дій користувачів і аналіз мережевих атак.

У зовнішньому контурі передбачено DMZ-сегмент, який забезпечує розміщення публічних веб-ресурсів і API-сервісів із використанням DNAT-правил. Доступ до інтернету та партнерських автономних систем реалізується через SNAT/PAT механізми з підтримкою VPN-тунелів та IPSec інкапсуляції. Для розширення масштабованості застосовується гібридна інтеграція з хмарними NAT-шлюзами (AWS/Azure/GCP), що дозволяє переносити частину навантаження у зовнішні середовища з гарантованим egress-керуванням і телеметрією.

Аналіз структурної схеми системи (рис. 1.11) дозволяє визначити, що поєднання модулів NAT-трансляції, політик доступу, шифрування та моніторингу формує цілісну архітектуру, здатну забезпечувати ізоляцію

внутрішніх сегментів, адаптивну маршрутизацію трафіку й захист інформаційних потоків у реальному часі. Така організація компонентів демонструє підхід до побудови стійких автономних мереж, у яких баланс між продуктивністю, безпекою та керованістю досягається за рахунок централізованого контролю NAT-процесів і автоматизованої аналітики подій.

1.5 Аналіз вимог системи

Аналіз вимог системи є ключовим етапом проєктування захищених автономних мереж із використанням технологій NAT, оскільки саме на цьому етапі визначаються функціональні, технічні та безпекові характеристики, від яких залежить ефективність, масштабованість і стійкість архітектури. Вимоги формуються на основі аналізу предметної області (п. 1.1), теоретико-методологічних засад (п. 1.2) та огляду існуючих рішень (п. 1.3), що дало змогу виокремити основні напрями подальшої розробки.

Система NAT для побудови захищених автономних мереж має забезпечувати стабільну трансляцію адрес, гнучке управління політиками доступу, інтеграцію з механізмами шифрування та автентифікації, а також повну відповідність сучасним стандартам безпеки мережевого рівня.

Функціональні вимоги описують базову логіку роботи системи, її призначення та взаємодію компонентів у процесі трансляції, маршрутизації й контролю доступу. Визначені характеристики забезпечують виконання NAT-функцій, моніторинг подій і взаємодію з іншими підсистемами (VPN, PKI, SIEM). Функціональні вимоги представлені в таблиці 1.3

Таблиця 1.3 – Функціональні вимоги до системи NAT у захищеній автономній мережі

№	Вимога	Опис
1	Підтримка SNAT, DNAT, PAT	Реалізація повного набору механізмів трансляції адрес і портів для забезпечення взаємодії між приватними та публічними мережами.

2	Динамічне формування таблиць NAT	Автоматичне створення та очищення записів після завершення сесій або таймауту.
3	Централізоване керування ACL-правилами	Єдина точка управління політиками доступу через графічний інтерфейс адміністратора (Java Swing).
4	Інтеграція з VPN/IPSec NAT-T	Можливість тунелювання трафіку через NAT-вузли з підтримкою NAT Traversal.
5	Логування і моніторинг сесій	Реєстрація з'єднань і змін конфігурацій у SIEM/Logs DB з можливістю аналітики.
6	Автентифікація через PKI/AAA	Підтримка сертифікатів, токенів та ролей доступу (RBAC).
7	Візуалізація політик і трафіку	Відображення поточних NAT-таблиць, маршрутів і стану модулів у GUI.

Технічні вимоги визначають апаратні й програмні характеристики, які гарантують продуктивність і надійність системи. У контексті побудови автономної мережі вони охоплюють показники пропускної здатності, сумісності, стабільності та масштабованості. Технічні вимоги представлені у таблиці 1.4.

Таблиця 1.4 – Технічні вимоги до системи

№	Вимога	Значення / опис
1	Пропускна здатність NAT-ядра	≥ 1 Гбіт/с при середньому навантаженні.
2	Максимальна затримка обробки	≤ 5 мс для локального трафіку, ≤ 15 мс для тунельованого.
3	Обсяг таблиць NAT	Не менше 100 000 одночасних записів без деградації швидкодії.
4	Надійність (MTBF)	Не менше 10 000 годин без критичних збоїв, автоматичне відновлення процесів.
5	Підтримка середовищ	Linux, Windows Server, Docker / Kubernetes.

Продовження таблиці 1.4

6	Масштабованість	Горизонтальний розподіл навантаження між вузлами NAT-кластеру.
7	Інтеграція з SIEM / API	REST/HTTPS із TLS 1.3 для передачі журналів і метрик безпеки.

Вимоги цієї категорії визначають комплекс заходів, спрямованих на забезпечення конфіденційності, цілісності та доступності даних, а також на протидію загрозам несанкціонованого доступу, підміни пакетів і атак типу DoS. Вимоги до безпеки системи представлені у таблиці 1.5

Таблиця 1.5 – Вимоги до безпеки системи

№	Вимога	Опис
1	Ідентифікація та автентифікація	Використання OAuth 2.0 / OpenID Connect для доступу до адміністративного інтерфейсу.
2	Захист каналів зв'язку	TLS 1.3 для API та GUI, IPSec/WireGuard для міжвузлових тунелів.
3	Виявлення аномалій трафіку	Аналіз NAT-сесій для виявлення spoofing-, DoS- і scan-атак у реальному часі.
4	Захист журналів і конфігурацій	Шифрування, контроль цілісності (SHA-256), обмеження доступу RBAC.
5	Безпечне оновлення компонентів	OTA-оновлення з перевіркою цифрових підписів і контролем версій.
6	Відповідність стандартам	ISO/IEC 27001, NIST SP 800-41, RFC 7857.

Аналіз вимог свідчить, що система NAT для автономних мереж має поєднувати функції маршрутизації, трансляції адрес, шифрування й аналітики безпеки у єдиному середовищі з централізованим управлінням. Згідно з викладеними вимогами, майбутня реалізація забезпечить підтримку багаторівневої адресної трансляції (SNAT/DNAT/PAT), динамічне керування політиками ACL, інтеграцію з VPN-тунелями та захищеними протоколами, а також автоматизований збір телеметрії для виявлення аномалій трафіку. Сформульовані параметри створюють методологічну основу для розробки архітектури системи, моделювання її процесів і подальшої реалізації програмного комплексу в середовищі Java Swing із підтримкою PKI, AAA, SIEM та IPSec NAT-Traversal, що забезпечить адаптивність, стійкість і безпечну взаємодію компонентів у межах автономної мережі.

1.6 Постановка завдання

Постановка завдання передбачає визначення структури вхідних і вихідних даних, принципів їх оброблення та функціональної взаємодії основних модулів системи. Об’єктом дослідження є процес маршрутизації та трансляції мережевого трафіку між внутрішніми та зовнішніми сегментами у захищеній автономній мережі, а предметом - програмно реалізовані механізми NAT, моніторингу та керування політиками доступу.

До вхідних даних системи належать:

- параметри мережевих інтерфейсів (IP-адреси, порти, протоколи TCP/UDP);
- правила доступу та конфігурації ACL;
- параметри VPN/IPSec-тунелів і сертифікати автентифікації (PKI/AAA);
- трафік користувачів та службових процесів з приватного сегмента (RFC1918);
- дані журналів і телеметрії, що надходять від сенсорів системи моніторингу.

До вихідних даних належать:

- таблиці активних NAT-трансляцій (SNAT, DNAT, PAT);
- журнали подій, лог-файли з даними про сесії, маршрути та часові мітки;
- аналітичні звіти щодо навантаження, статистики з’єднань, виявлених аномалій;
- візуалізація станів мережі та політик у графічному інтерфейсі адміністратора (Java Swing);
- повідомлення безпеки й сигнали SIEM-підсистеми.

Основні процеси системи включають: аналіз вхідного трафіку, визначення політики маршрутизації, виконання NAT-трансляції, контроль доступу за ACL, шифрування та тунелювання міжмережевих з'єднань, журналювання й моніторинг подій, а також візуалізацію результатів у режимі реального часу.

Завдання полягає у створенні системи, яка забезпечує динамічну трансляцію адрес, збір і аналіз телеметрії, централізоване управління політиками доступу та захист каналів зв'язку між сегментами автономної мережі з подальшою інтеграцією з модулями PKI, VPN і SIEM.

1.7 Висновки до першого розділу

У першому розділі здійснено системний аналіз предметної області та теоретичних засад побудови захищених автономних мереж із використанням технологій NAT. Розглянуто принципи функціонування механізмів трансляції мережевих адрес (SNAT, DNAT, PAT, CGNAT), а також особливості їх інтеграції у середовищах із підвищеними вимогами до безпеки та ізоляції трафіку. Було досліджено взаємодію NAT-рішень з допоміжними технологіями – VPN, IPSec, TLS, STUN/TURN/ICE – що забезпечують сумісність між внутрішніми сегментами мережі та зовнішніми каналами зв'язку.

Проведено аналіз сучасних апаратних і програмних реалізацій NAT-підсистем, зокрема Cisco, Fortinet, Palo Alto, F5 та AWS/Azure Cloud NAT, що дало змогу визначити їхні архітектурні особливості, сильні сторони та обмеження. Сформовано структурну модель захищеної автономної мережі, яка включає ядро NAT/Firewall, DMZ-сегмент, VPN-шлюз, SIEM-підсистему та контури OT/IoT і IT-сервісів, між якими здійснюється контрольований обмін даними.

На основі аналізу визначено функціональні, технічні та безпекові вимоги до системи, зокрема щодо пропускної здатності, надійності, часів реакції,

підтримки шифрування, журналювання, RBAC-контролю та інтеграції з зовнішніми SIEM/AAA-системами. Сформульовано постановку завдання, що передбачає розроблення програмно-апаратного комплексу NAT з модульною архітектурою, підтримкою VPN-тунелювання, централізованим моніторингом і можливістю масштабованої симуляції процесів трансляції адрес у межах дослідного стенду.

Таким чином, у результаті проведеного аналізу обґрунтовано актуальність створення інтелектуального стенду для дослідження NAT-механізмів, визначено архітектурні принципи майбутньої системи, сформовано вимоги до її функціонування та закладено теоретико-методологічну основу для подальшого проєктування і реалізації програмного емулятора, описаного у другому розділі.

2 ПРОЄКТУВАННЯ СИСТЕМИ ТРАНСЛЯЦІЇ АДРЕС У ЗАХИЩЕНІЙ АВТОНОМНІЙ МЕРЕЖІ

2.1 Функціональна схема та логічна архітектура підсистеми NAT

Функціональна схема системи відображає узагальнену архітектуру взаємодії основних компонентів захищеної автономної мережі з реалізованим ядром NAT/Firewall, яке виконує функції трансляції адрес, маршрутизації та контролю доступу в режимі реального часу. На рисунку 2.1 представлено логічну архітектуру підсистеми NAT, у якій кожен модуль відіграє важливу роль у циклі оброблення мережевого трафіку, імітуючи процеси взаємодії між OT/IoT-сегментом, NAT-ядром, політиками безпеки та зовнішніми доменами. *Логічну* архітектуру системи показано на рис. 2.1.

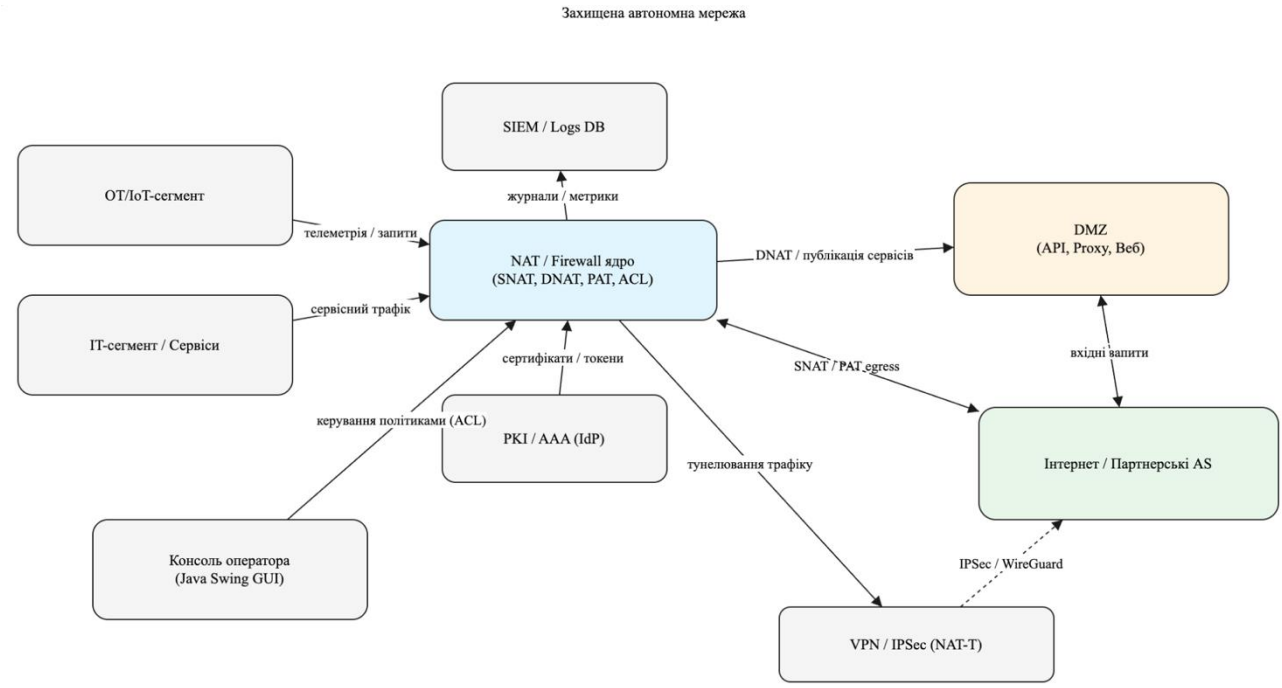


Рис. 2.1 – Функціональна схема та логічна архітектура підсистеми NAT у межах захищеної автономної мережі

Представлена архітектура формує основу для розроблення програмного середовища моделювання NAT-процесів, яке забезпечує дослідження механізмів адресної трансляції, маршрутизації подій і контролю політик безпеки. Центральним елементом системи є NAT / Firewall-ядро (SNAT,

DNAT, PAT, ACL), через яке проходить увесь потік інформації, що генерується сенсорами та сервісами. У цьому модулі здійснюється трансляція адрес, фільтрація пакетів, керування доступом і передача журналів до SIEM / Logs DB, який забезпечує аудит дій і аналітику безпеки.

Інтеграція з підсистемами PKI / AAA (IdP) гарантує автентичність користувачів і довіру до взаємодіючих вузлів через цифрові сертифікати. Консоль адміністратора, реалізована у середовищі Java Swing GUI, надає зручний інтерфейс для конфігурування ACL-правил, моніторингу таблиць трансляцій і збору телеметрії. Така структура дає змогу реалізувати лабораторне моделювання роботи NAT-механізмів у контексті побудови захищених автономних систем.

Зовнішня взаємодія здійснюється через DMZ-зону, у якій розміщуються API-сервіси, проксі-модулі та веб-інтерфейси. Передача даних до партнерських автономних систем відбувається за допомогоют SNAT / PAT egress, а опублікування внутрішніх сервісів – через DNAT. Для міжмережевого обміну застосовуються захищені тунелі IPSec / WireGuard (NAT-T), що забезпечують конфіденційність і цілісність переданих даних.

Архітектура підсистеми NAT демонструє важливість поєднання механізмів адресної трансляції, криптографічного захисту та централізованого моніторингу. Завдяки цьому система досягає високої стійкості, адаптивності до зміни топології та сумісності з іншими мережевими компонентами автономного середовища. *У*загальнену характеристику модулів підсистеми наведено у табл. 2.1.

Таблиця 2.1 – Основні модулі та функції підсистеми NAT у захищеній автономній мережі

№	Модуль	Функціональне призначення
1	NAT / Firewall-ядро	Трансляція адрес, контроль політик, маршрутизація трафіку
2	PKI / AAA (IdP)	Сертифікація, автентифікація користувачів, видача токенів

Продовження таблиці 2.1

3	Консоль адміністратора (Java Swing GUI)	Керування політиками, моніторинг сесій, аналітика
4	SIEM / Logs DB	Збір і кореляція журналів, аудит безпеки
5	DMZ / API Proxy	Публікація внутрішніх сервісів, обробка зовнішніх запитів
6	VPN / IPSec (NAT-T)	Захищене тунелювання між сегментами та партнерами
7	OT / IoT- та IT-сегменти	Генерація телеметрії, сервісних запитів і даних трафіку

Розглянута модель становить методологічну основу для подальшого розроблення програмного прототипу, який відтворює процеси адресної трансляції, маршрутизації, аналітики трафіку та контролю безпеки в межах багаторівневої архітектури захищеної автономної системи.

2.2 Електрична та монтажна схема дослідного стенду емулятора

Електрична та монтажна схема дослідного стенду емулятора призначена для відтворення апаратного середовища, у якому реалізується оброблення подій системи NAT, маршрутизація, фільтрація та аналіз трафіку. Архітектура побудована на принципах модульності, енергоізоляції та нормалізації сигналів, що забезпечує коректну роботу системи в умовах змінного навантаження і відсутності зовнішніх збоїв. Кожен модуль виконує окрему функцію в контурі обробки даних і утворює логічно завершену структуру дослідного стенду.

Текст посилання на рисунок: до складу апаратного комплексу дослідного стенду входять основні пристрої, зображені на рис. 2.2 – 2.6**.**

На рис. 2.2 показано маршрутизатор Cisco ISR C821, який виконує функції NAT-ядра системи. Він забезпечує трансляцію приватних адрес у публічні, підтримує політики SNAT, DNAT та PAT, а також базові фаєрвол-правила, що контролюють вхідний та вихідний трафік. Через цей модуль

реалізується маршрутизація між внутрішніми сегментами OT/IoT і зовнішніми доменами, формуючи центральний рівень безпеки.

Рис. 2.2 – Маршрутизатор Cisco ISR C821, що виконує функції NAT-ядра та базового фаєрвола.

На рис. 2.3 представлено PoE-комутатор на 48 портів, який виконує функції централізованої комутації та живлення периферійних OT/IoT-вузлів. Його застосування дозволяє скоротити кількість кабельних з'єднань і реалізувати подачу 48 V DC по Ethernet-лініях. Крім того, він підтримує VLAN-сегментацію, що дає можливість ізолювати різні рівні мережі під час експериментів.

Рис. 2.3 – PoE-комутатор 48-портовий, який забезпечує живлення периферійних OT/IoT-вузлів і серверних сегментів.

На рис. 2.4 зображено апаратний Firewall-шлюз, який реалізує політики ACL та виконує глибоку фільтрацію мережевого трафіку. Він відповідає за контроль сеансів, виявлення DoS та spoofing-атак, а також за підтримку VPN/IPSec тунелювання. Встановлення цього модуля в контурі між NAT-ядром і зовнішнім каналом забезпечує додатковий рівень захисту при обміні даними.

Рис. 2.4 – Апаратний Firewall-шлюз, який реалізує політики ACL і фільтрацію мережевого трафіку.

На рис. 2.5 наведено сервер SIEM/Logs DB, що забезпечує збір телеметрії, журналів подій та аналітику мережевої активності. Цей модуль акумулює лог-дані від NAT-ядра, фаєрвола та VPN-підсистеми, проводить кореляційний аналіз і формує звіти про аномалії та порушення безпеки. Завдяки цьому реалізується повний цикл моніторингу системи в режимі реального часу.

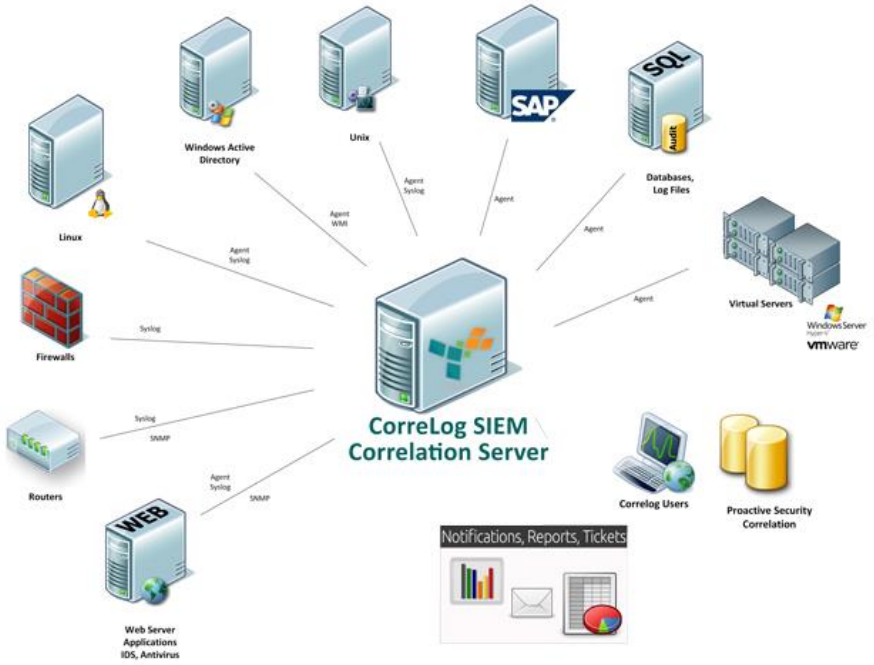


Рис. 2.5 – Сервер SIEM/Logs DB, що виконує збір телеметрії та аналітику безпекових подій.

На рис. 2.6 показано VPN/IPSec-підсистему, яка забезпечує захищене тунелювання трафіку між автономними сегментами та віртуальними мережами. Вона підтримує режими NAT-Traversal та WireGuard, що дозволяє

моделювати реальні канали взаємодії з мінімальною затримкою й повною криптографічною ізоляцією даних.

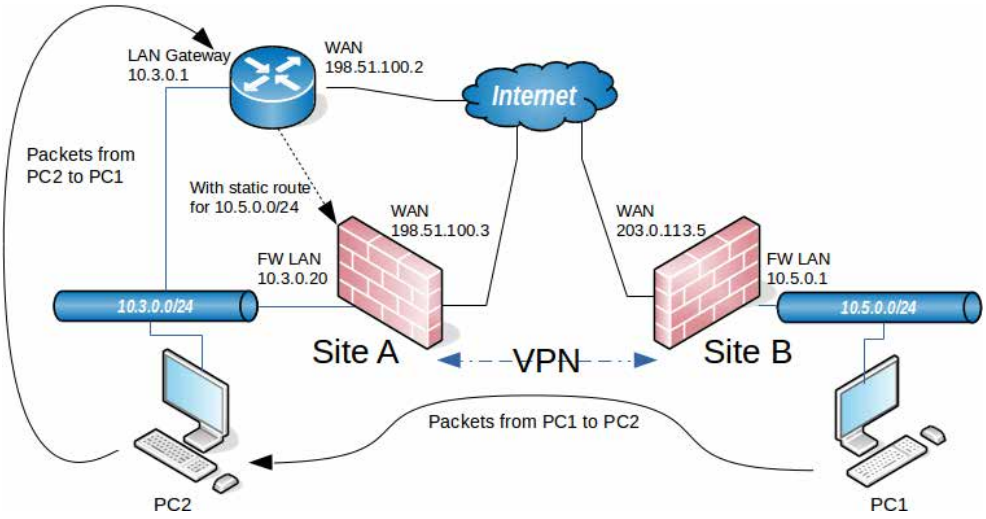


Рис. 2.6 – VPN/IPSec-підсистема, яка забезпечує захищене тунелювання між автономними сегментами.

Електричну схему живлення і захисту системи показано на рис. 2.7.

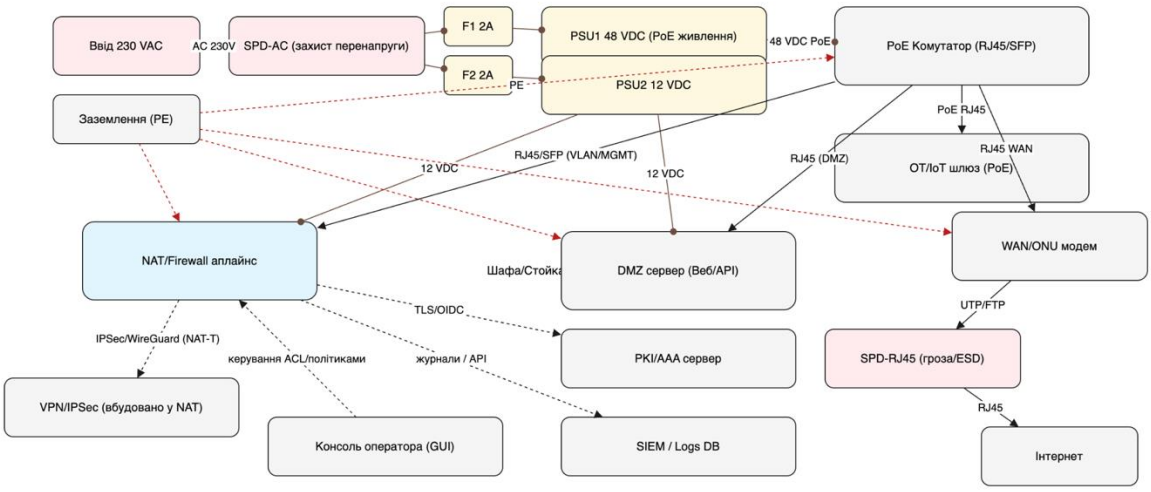


Рис. 2.7 – Електрична схема дослідного стенду системи NAT

На електричній схемі реалізовано подвійну систему живлення AC/DC з мережевим входом 230 VAC через захисний модуль SPD-AC та запобіжники F1–F2. Живлення 48 VDC забезпечується через PSU1 для PoE-комутатора та OT/IoT-шлюзів, а PSU2 (12 VDC) подає енергію на серверні модулі. Всі пристрої під’єднані до PE-шини заземлення, що забезпечує електробезпеку та нормалізацію потенціалів. У вихідному контурі передбачено SPD-RJ45 для

захисту мережевих ліній Ethernet від грозових імпульсів та ESD. *К*онструктивне компонування стенду подано на рис. 2.8.

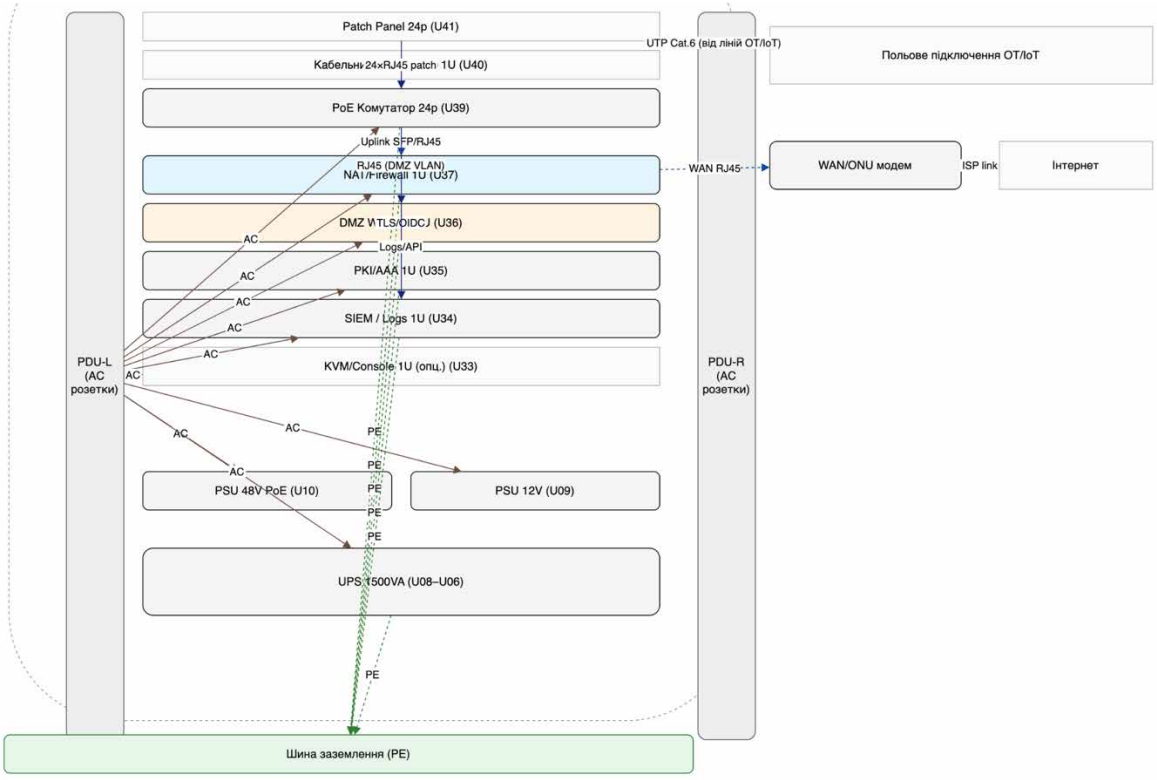


Рис. 2.8 – Монтажна схема дослідного стенду системи NAT у телекомунікаційній стійці

Монтажна структура побудована у форм-факторі rack 1U та включає послідовне розміщення модулів за функціональними рівнями: у верхній частині – Patch Panel і PoE-комутатор, далі – NAT/Firewall-ядро, DMZ та VPN/IPSec модулі, а в нижній – сервери PKI/AAA та SIEM, блоки живлення PSU і UPS. Така сегментація забезпечує раціональну вентиляцію, зручність обслуговування та ізоляцію силових ліній від сигнальних.

У табл. 2.2 наведено перелік основних апаратних компонентів дослідного стенду.

Таблиця 2.2 – Основні апаратні компоненти дослідного стенду системи NAT

№	Компонент	Тип / Модель	Призначення
1	Cisco ISR C821	Маршрутизатор / NAT-ядро	Трансляція адрес, маршрутизація, ACL

Продовження таблиці 2.2

2	PoE комутатор 48-портовий	Ubiquiti EdgeSwitch 48	Комутація, живлення OT/IoT через PoE
3	Firewall-аплайнс	Mikrotik / Fortinet	Фільтрація трафіку, політики безпеки
4	SIEM / Logs DB сервер	Dell 1U rack	Журналювання та аналітика подій
5	VPN/IPSec підсистема	WireGuard / OpenVPN	Тунелювання між автономними системами

Розроблена електрична та монтажна архітектура забезпечує стійке енергопостачання, ізоляцію контурів і коректну роботу всіх модулів під час експериментального дослідження NAT-функцій. Вона гарантує стабільність параметрів живлення, відповідність нормам заземлення PE, а також можливість масштабування та підключення нових сегментів без порушення електричної балансності системи.

2.3 Передумови створення програмного емулятора та вибір технологічного стеку

Передумови створення програмного емулятора системи NAT ґрунтуються на необхідності формування відтворюваного середовища для дослідження взаємодії апаратних і програмних компонентів у межах захищеної автономної мережі. У межах дослідного стенду апаратна частина забезпечує фізичну маршрутизацію, трансляцію адрес і комутацію трафіку, проте для проведення повноцінного аналізу поведінки NAT-механізмів, VPN-тунелювання та журналювання подій необхідне програмне середовище, здатне імітувати динаміку оброблення мережевих запитів у реальному часі.

Розроблення програмного емулятора дозволяє дослідити логіку взаємодії між внутрішніми компонентами NAT/Firewall, відтворити механізми формування таблиць відповідності (connection tracking), реалізувати імітацію сесійних потоків, а також оцінити навантаження на серверну інфраструктуру

під час одночасної обробки запитів. Використання емулятора спрощує процес верифікації алгоритмів маршрутизації, тестування правил безпеки та оптимізації параметрів продуктивності без залучення великої кількості фізичних пристроїв.

З огляду на ці вимоги, вибір технологічного стеку орієнтовано на забезпечення високої швидкодії, гнучкості конфігурацій, підтримки мережевих протоколів і сумісності з існуючими апаратними рішеннями. Програмна реалізація системи поділяється на кілька логічних рівнів: рівень серверної логіки, рівень аналітики та емуляції подій, рівень взаємодії з користувачем. Для кожного з них обрано оптимальні інструменти, що гарантують стабільність і масштабованість.

Склад обраного технологічного стеку для реалізації програмного емулятора подано в табл. 2.3.

Таблиця 2.3 – Обраний технологічний стек для реалізації програмного емулятора системи NAT

№	Компонент системи	Технологія / фреймворк	Призначення
1	Серверна логіка	Python (FastAPI)	Реалізація REST API, обробка подій NAT/Firewall, інтеграція з БД
2	База даних	PostgreSQL / SQLite	Зберігання журналів, таблиць сесій, параметрів емуляції
3	Емуляція трафіку	AsyncIO, Scapy	Імітація мережевих запитів, пакетний аналіз, SNAT/DNAT обробка
4	Інтерфейс користувача	PyQt6 / HTML Dash UI	Візуалізація стану системи, керування сценаріями емуляції
5	Безпековий контур	OpenSSL, TLS 1.3	Шифрування з’єднань, генерація ключів, захист обміну даними
6	Контейнеризація	Docker Compose	Модульна розгортка серверних компонентів і БД
7	Моніторинг і логування	Prometheus, Grafana	Збір метрик, візуалізація аналітики та контроль навантаження

Запропонований стек технологій забезпечує оптимальне співвідношення між швидкодією, надійністю та гнучкістю. Python як основна мова дозволяє

реалізувати багатопоточну обробку запитів у реальному часі, FastAPI гарантує асинхронність та масштабованість, а PostgreSQL забезпечує структуроване зберігання даних журналів і станів NAT-з'єднань. Docker-контейнеризація дає змогу швидко розгортати систему на різних середовищах без зміни конфігурації, а Prometheus і Grafana забезпечують моніторинг продуктивності й кореляцію подій під час тестів. Завдяки такій побудові програмний емулятор є універсальним інструментом для моделювання, тестування й оптимізації NAT-систем у складі захищених автономних мереж.

2.4 Формалізація специфікації повідомлень і тем MQTT

Формалізація специфікації повідомлень і тем MQTT у межах програмного емулятора є ключовим етапом, який забезпечує уніфіковану взаємодію між компонентами системи NAT, шлюзами OT/IoT, консоллю оператора, брокером DMZ і модулем аналітики SIEM. MQTT обрано як базовий транспортний протокол завдяки його легковаговості, мінімальній затримці передачі, підтримці зворотних каналів та стійкості до розривів з'єднання, що робить його оптимальним для автономних систем з обмеженими ресурсами. У дослідному стенді протокол MQTT використовується для передачі телеметрії, команд керування, аудиту подій та оновлення конфігурацій між програмними агентами та серверними підсистемами емулятора.

Логічну структуру взаємодії клієнтів, брокера та тем у системі емулятора показано на рис. 2.9.

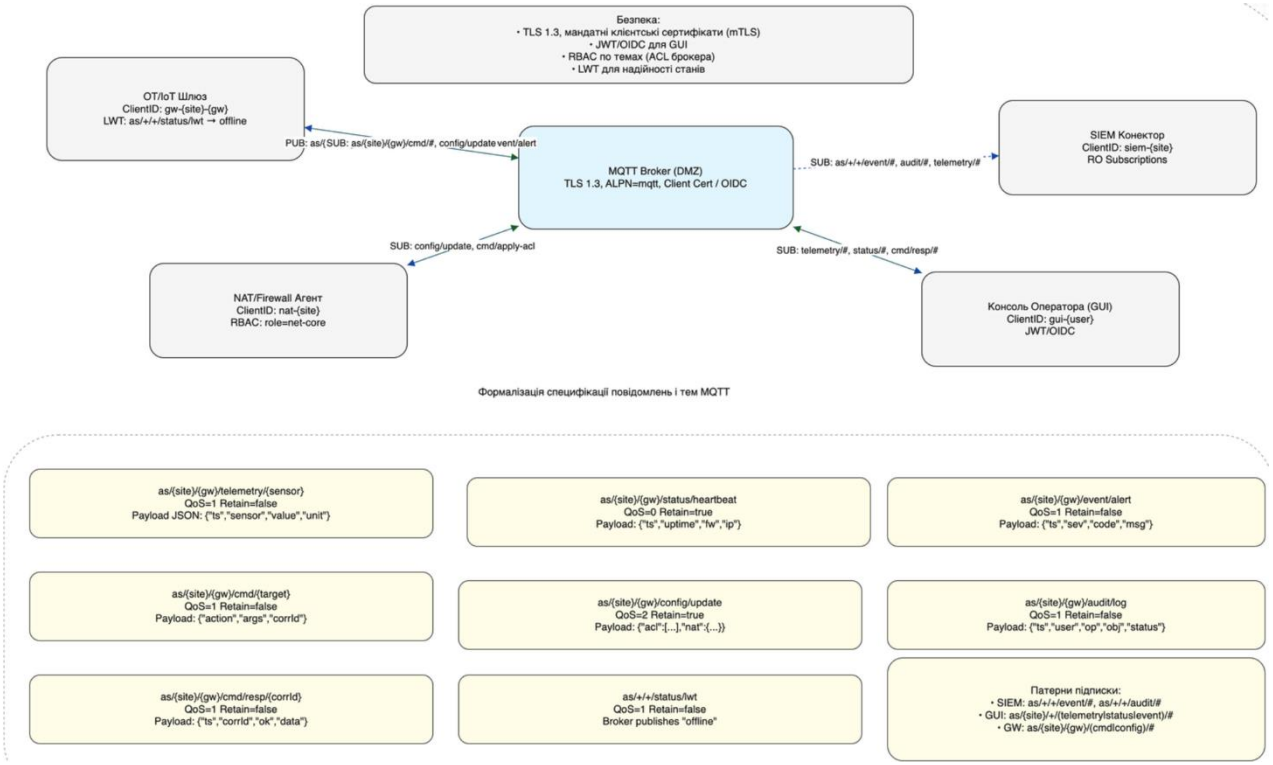


Рис. 2.9 – Формалізація специфікації повідомлень і тем MQTT у дослідному стенді емулятора

Як показано на рисунку, центральним вузлом архітектури виступає MQTT Broker (DMZ), який функціонує з використанням TLS 1.3, клієнтських сертифікатів і OIDC-аутентифікації. Він забезпечує безпечний розподіл повідомлень між п’ятьма основними компонентами:

– OT/IoT шлюзом, який передає телеметрію та статуси підключення через топіки as/{site}/gw/telemetry/# і as/+/+/status/lwt;

– агентом NAT/Firewall, який публікує конфігураційні зміни, команди ACL і дані про політики;

– консоллю оператора (GUI), що отримує статусні повідомлення, відповіді команд і керує сценаріями;

– SIEM-конектором, який здійснює пасивну підписку на канали event, audit і telemetry для збору журналів;

– MQTT-брокером, який підтримує RBAC-доступ, QoS-рівні та LWT-повідомлення для контролю надійності сеансів.

Система тем побудована за ієрархічним принципом: кожен канал включає ідентифікатори site, gateway, sensor, що дозволяє однозначно

визначати джерело даних і тип повідомлення. Використання схем JSON у payload забезпечує сумісність із REST-інтерфейсами серверної частини та полегшує обробку аналітичних подій.

Структуровану специфікацію тем і форматів повідомлень наведено у табл. 2.4.

Таблиця 2.4 – Специфікація основних тем і повідомлень MQTT у програмному емуляторі

№	Тема MQTT	Рівень QoS / Retain	Формат даних (payload)	Призначення
1	as/{site}/{gw}/telemetry/{sensor}	QoS 1 / Retain false	JSON {"ts","sensor","value","unit"}	Передавання телеметрії з OT/IoT-вузлів
2	as/{site}/{gw}/status/heartbeat	QoS 0 / Retain true	JSON {"ts","uptime","fw","ip"}	Підтримка heartbeat-сигналів шлюзу
3	as/{site}/{gw}/cmd/{target}	QoS 1 / Retain false	JSON {"action","args","corrId"}	Відправлення команд керування
4	as/{site}/{gw}/cmd/resp/{corrId}	QoS 1 / Retain false	JSON {"ts","corrId","ok","data"}	Отримання результатів виконання команд
5	as/{site}/{gw}/config/update	QoS 2 / Retain true	JSON {"acl": […], "nat": {…}}	Оновлення конфігурацій NAT-модуля
6	as/{site}/{gw}/event/alert	QoS 1 / Retain false	JSON {"ts","sev","code","msg"}	Повідомлення про аномалії та сповіщення
7	as/{site}/{gw}/audit/log	QoS 1 / Retain false	JSON {"ts","user","op","obj","status"}	Передача логів дій користувачів
8	as/+/+/status/lwt	QoS 1 / Retain false	Текст "offline"	Стан сесії (LWT) для перевірки доступності клієнтів

Запровадження формалізованої структури тем MQTT забезпечує стабільність обміну повідомленнями між усіма учасниками мережі, дозволяє автоматично відновлювати сесії після втрат з’єднання й гарантує логічну відокремленість рівнів доступу через механізми RBAC та ACL. Використання

мандатних сертифікатів і TLS-шифрування мінімізує ризики перехоплення трафіку, а збереження форматів payload у JSON-структурах забезпечує повну сумісність із модулями SIEM, GUI та API серверної частини. Таким чином, розроблена специфікація тем MQTT у складі програмного емулятора є основою надійного, стандартизованого та безпечного обміну даними між усіма підсистемами NAT-інфраструктури.

2.5 Висновки до другого розділу

У другому розділі проведено комплексне проєктування програмно-апаратного середовища дослідного стенду системи NAT та формалізовано підхід до створення емулятора серверної обробки подій у межах захищеної автономної мережі. Було розроблено електричну та монтажну схеми стенду, які забезпечують модульність, нормалізацію контурів живлення, захист інформаційних ліній і стабільність взаємодії компонентів у реальному часі. Визначено базовий набір апаратних пристроїв - маршрутизатор Cisco ISR C821, PoE-комутатор, апаратний шлюз безпеки, сервер SIEM/Logs DB та VPN/IPSec-модуль, що формують інфраструктуру для дослідження NAT-топологій і політик доступу.

Описано передумови створення програмного емулятора, який дозволяє відтворити роботу NAT-механізмів, таблиць трансляції, ACL-правил і процесів оброблення трафіку без потреби у фізичних розгортаннях великомасштабних мереж. Обґрунтовано вибір технологічного стеку - Python (FastAPI), PostgreSQL, AsyncIO, PyQt6, Docker Compose, Prometheus та Grafana - як оптимального для реалізації багаторівневої, асинхронної та масштабованої архітектури з підтримкою моніторингу і безпеки.

Також розроблено формалізовану специфікацію повідомлень і тем протоколу MQTT, що уніфікує обмін телеметрією, командами, статусами та журналами між OT/IoT-шлюзами, NAT-агентами, консоллю оператора, брокером DMZ та SIEM-модулем. Визначено стандартизовану структуру тем,

рівні QoS та JSON-формати payload, які забезпечують сумісність і надійність взаємодії підсистем.

Результати другого розділу створюють технічне підґрунтя для реалізації програмного емулятора - сформовано логічну, енергетичну та інформаційну архітектуру системи, узгоджено механізми обміну даними й забезпечено умови для подальшої розробки алгоритмів і модулів програмного середовища, описаних у наступному розділі.

3 ПРОГРАМНА ІМПЛЕМЕНТАЦІЯ ЗАХИЩЕНОЇ АВТОНОМНОЇ МЕРЕЖІ З ТЕХНОЛОГІЄЮ NAT

3.1 Аналіз механізмів трансляції адрес у NAT-інфраструктурі системи

У цьому підрозділі розглянуто принципи функціонування та типові сценарії реалізації технологій NAT, що є основою для побудови дослідного стенду захищеної автономної мережі. NAT (Network Address Translation) забезпечує перетворення приватних IP-адрес внутрішніх сегментів у глобальні адреси зовнішніх мереж, дозволяючи оптимізувати використання адресного простору IPv4, ізолювати трафік і реалізувати контроль доступу. На рис. 3.1 показано базовий механізм статичної трансляції, у якому внутрішні вузли з адресами 10.1.1.x отримують зовнішні відповідники 203.0.113.x, що дозволяє підтримувати зворотний зв’язок при комунікації з Інтернетом.

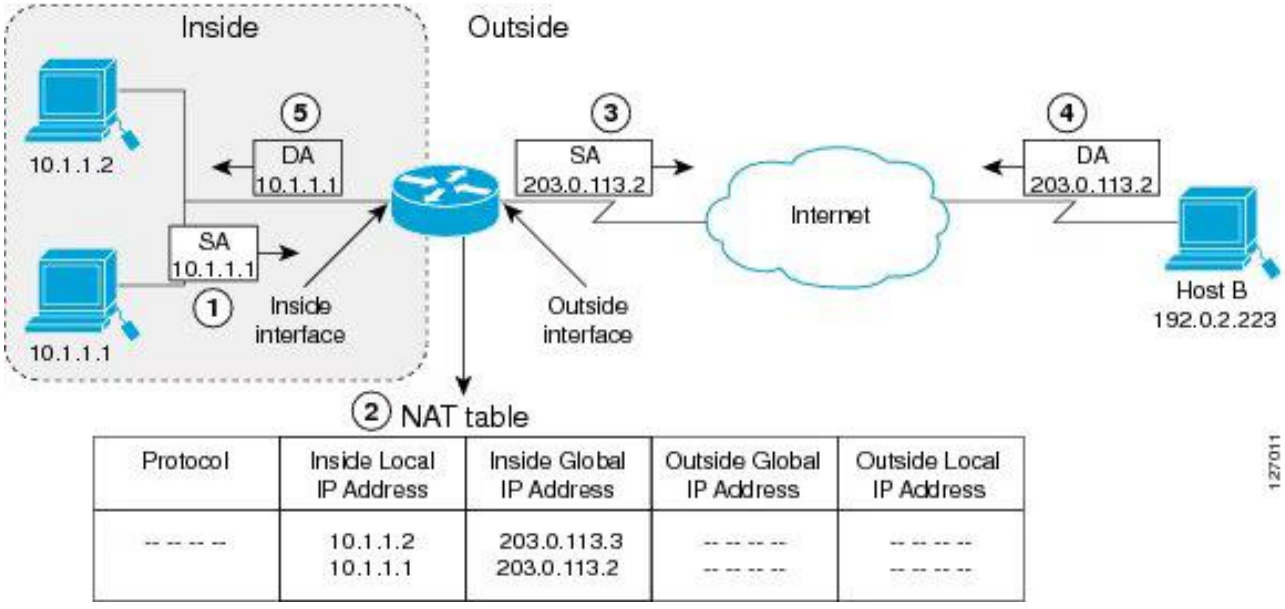
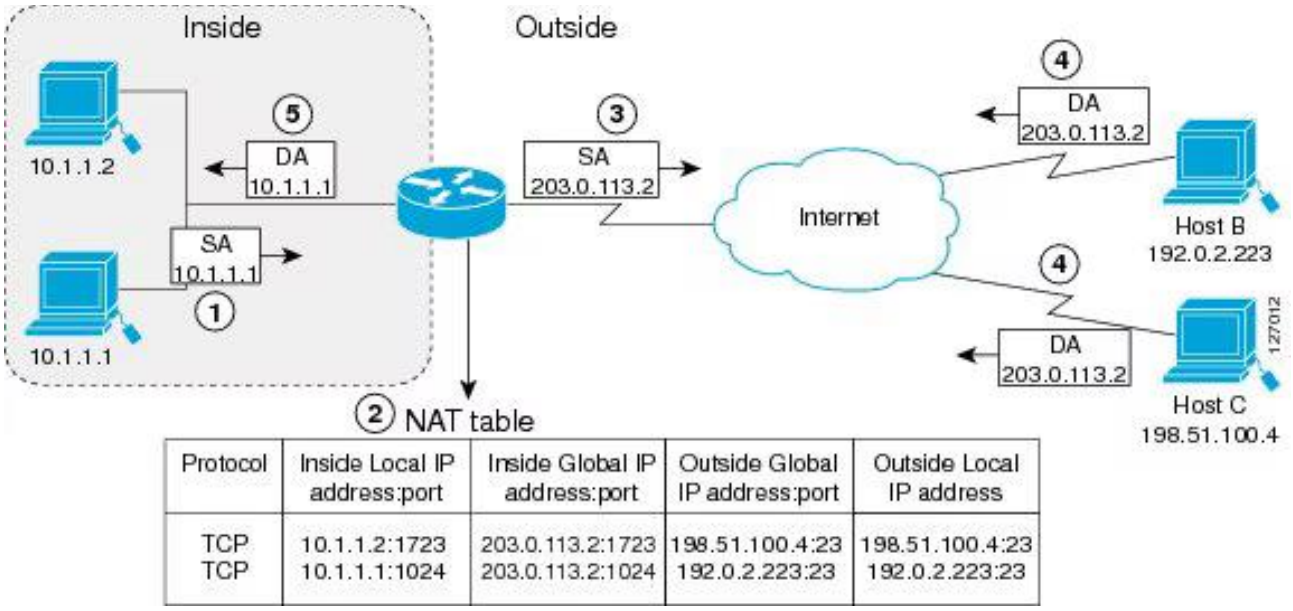


Рис. 3.1 – Принцип дії статичної NAT-трансляції між локальними та глобальними адресами

Розвитком цього підходу є динамічна трансляція з урахуванням портів, відома як PAT (Port Address Translation). На рис. 3.2 наведено приклад, коли декілька внутрішніх клієнтів використовують одну глобальну адресу, а

розрізнення сесій відбувається за портами TCP. Такий підхід дає змогу одночасно обслуговувати велику кількість з'єднань, не витрачаючи додаткові зовнішні IP-адреси, що особливо актуально для IoT-середовищ і корпоративних VPN-шлюзів.



Protocol	Inside Local IP address:port	Inside Global IP address:port	Outside Global IP address:port	Outside Local IP address
TCP	10.1.1.2:1723	203.0.113.2:1723	198.51.100.4:23	198.51.100.4:23
TCP	10.1.1.1:1024	203.0.113.2:1024	192.0.2.223:23	192.0.2.223:23

Рис. 3.2 – Механізм PAT із розподілом портів для кількох внутрішніх клієнтів

У сучасних інфраструктурах NAT часто працює у зв'язці з DNS-сервісом для резолвінгу імен у зовнішні адреси. На рис. 3.3 наведено приклад взаємодії, де внутрішній клієнт 10.1.1.1 через NAT-маршрутизатор виконує DNS-запит і отримує відповідь із глобальною адресою віддаленого вузла. Така схема забезпечує прозорість для користувача, водночас підтримуючи ізоляцію між внутрішнім і зовнішнім середовищем.

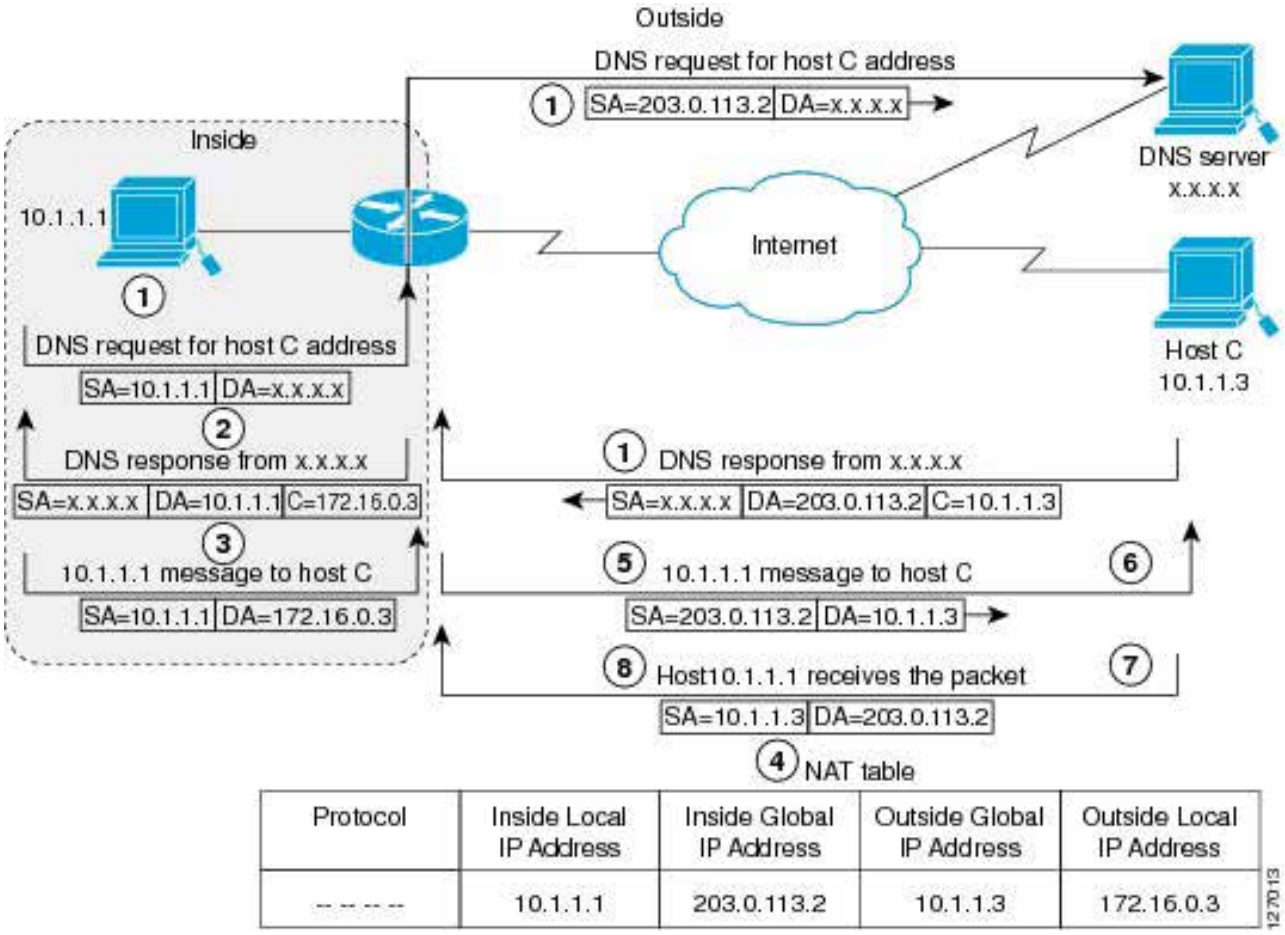


Рис. 3.3 – Інтеграція NAT і DNS-сервісу для резолвінгу зовнішніх адрес

Для тестування поведінки NAT-механізмів під навантаженням у межах лабораторного середовища розроблено сценарій з множиною реальних і віртуальних хостів. На рис. 3.4 подано приклад багатокористувацької NAT-таблиці, що містить відповідності між внутрішніми клієнтами та зовнішніми портами. Це дозволяє досліджувати продуктивність емулятора, час оновлення таблиць трансляцій і стійкість системи до великої кількості одночасних з’єднань.

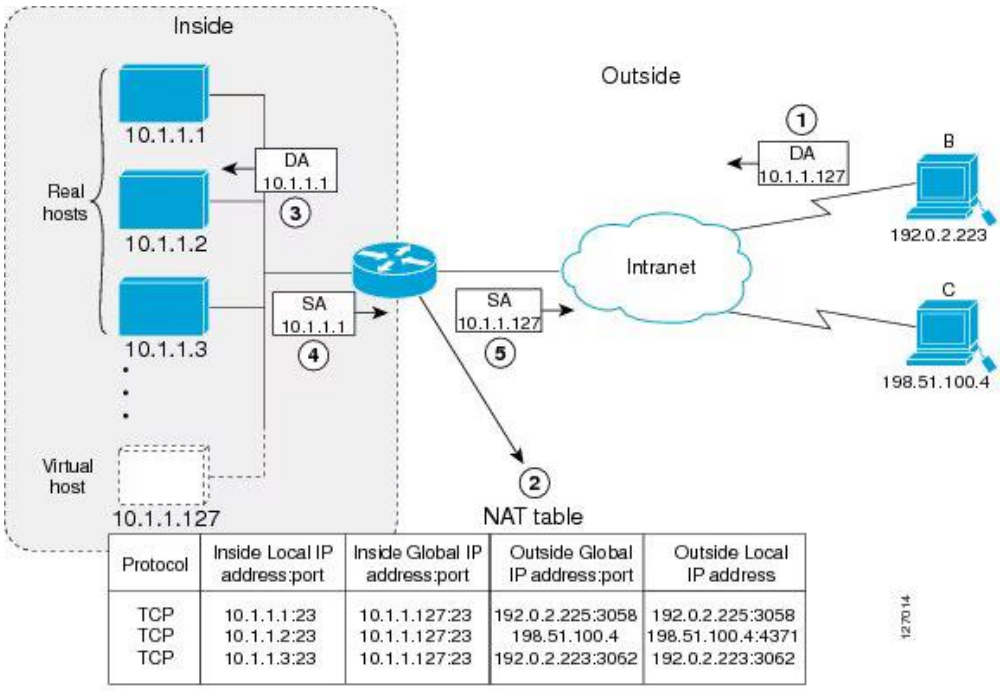


Рис. 3.4 – Приклад багатокористувацької NAT-таблиці з віртуальними вузлами

З метою підвищення безпеки трафік у дослідному стенді поділено на логічні зони, між якими діють чіткі політики доступу. На рис. 3.5 показано модель зональної маршрутизації, де NAT-ядро ізолює зони Z1 і Z2, а комунікація між ними можлива лише за визначеними політиками.

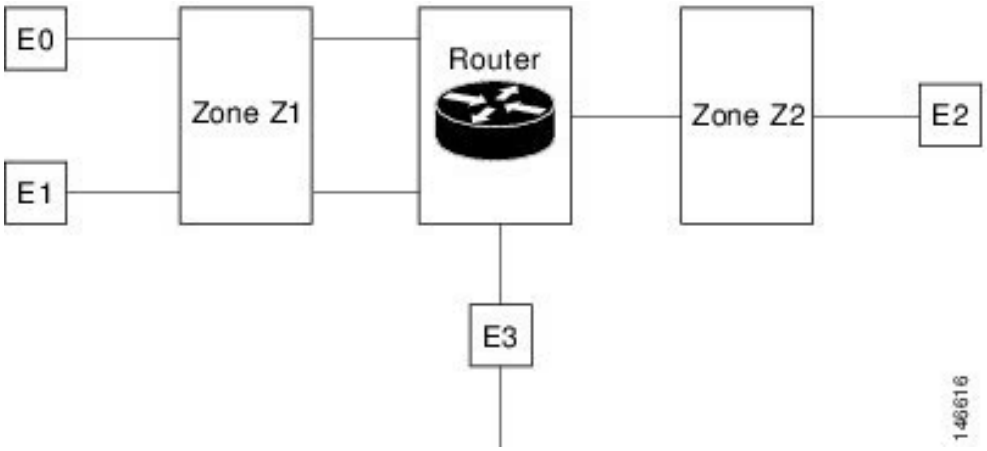


Рис. 3.5 – Модель зональної маршрутизації з ізоляцією Z1–Z2 через NAT-ядро

Таке зонування доповнюється реалізацією Zone-Based Policy Firewall, що дозволяє фільтрувати трафік залежно від напрямку потоку, протоколу та

ролі інтерфейсу. Як видно з рис. 3.6, політика між зонами реалізується через пари Zone Pair, у яких для кожного напрямку визначено окремі ACL-правила.

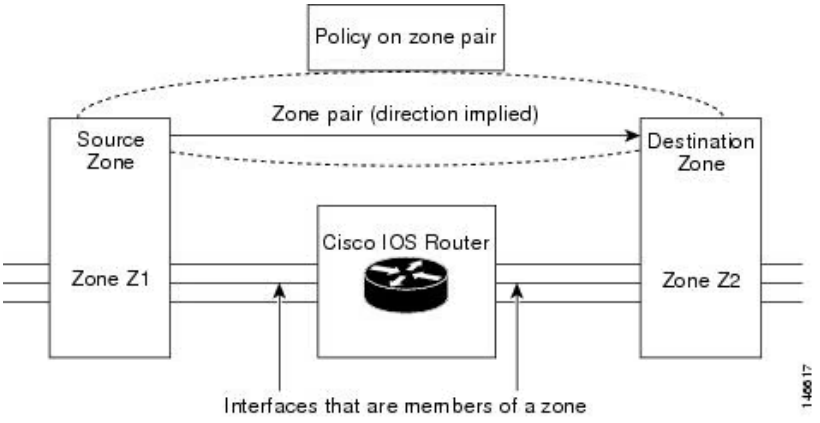


Рис. 3.6 – Застосування політик доступу між зонами через NAT-ядро (Zone Pair Policy)

Для забезпечення безпечного міжфілійного обміну даними NAT-механізми інтегруються з транспортом IPSec VPN, що створює зашифровані тунелі між периферійними вузлами та центром обробки даних. На рис. 3.7 подано типову топологію такої взаємодії між філією та Data Center, де NAT-модуль поєднує внутрішню адресацію з зовнішнім IPSec-каналом.

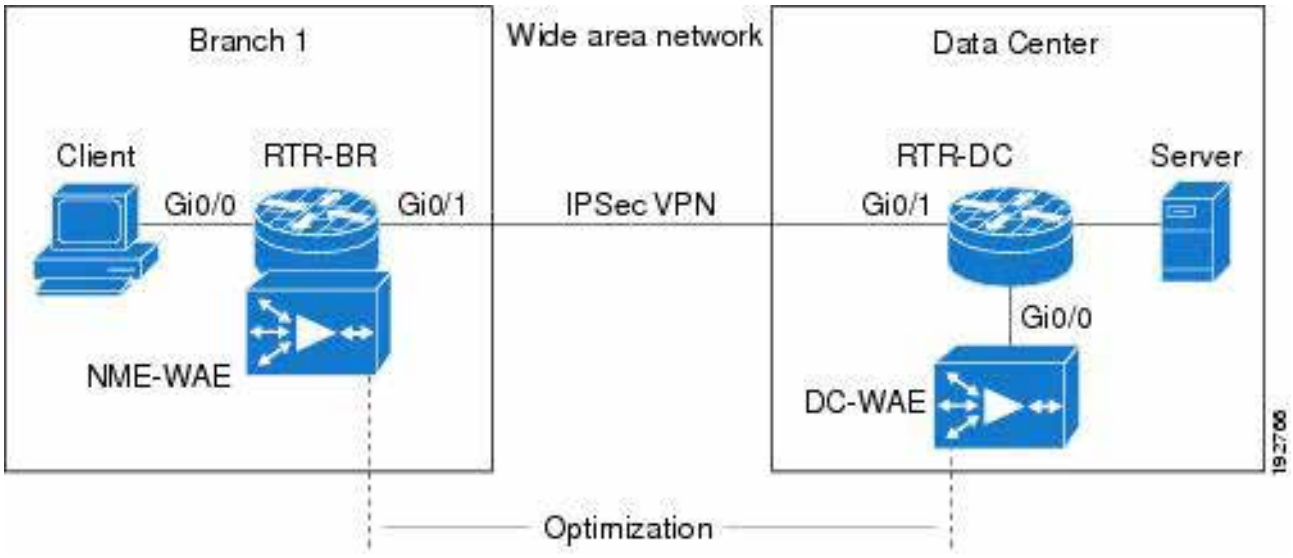


Рис. 3.7 – Інтеграція NAT і IPSec VPN у топології міжфілійного з’єднання

Моделювання механізмів трансляції адрес демонструє ключову роль NAT у побудові гібридних і багатосегментних мереж. Використання PAT, DNS-резолвінгу, зональної маршрутизації та VPN-транспорту дозволяє

досягти високого рівня безпеки, оптимізувати адресний простір і забезпечити стабільну взаємодію між компонентами автономної системи.

3.2 Логічна архітектура захищеної автономної мережі з технологією NAT

Логічна архітектура дослідного стенду побудована за принципом багаторівневого поділу мережевих доменів із чітко визначеними зонами безпеки та функціональними ролями. В основі системи лежить NAT/Firewall-ядро, яке забезпечує реалізацію трансляції адрес (SNAT, DNAT, PAT) і контроль доступу за допомогою політик ACL та станів з'єднань. Такий підхід дозволяє створити централізований вузол маршрутизації, що одночасно виконує функції безпеки, моніторингу та трафік-інспекції. На рис. 3.2 представлено структурну модель взаємодії між VLAN-сегментами, DMZ-зоною, VPN-шлюзом і зовнішніми мережами.

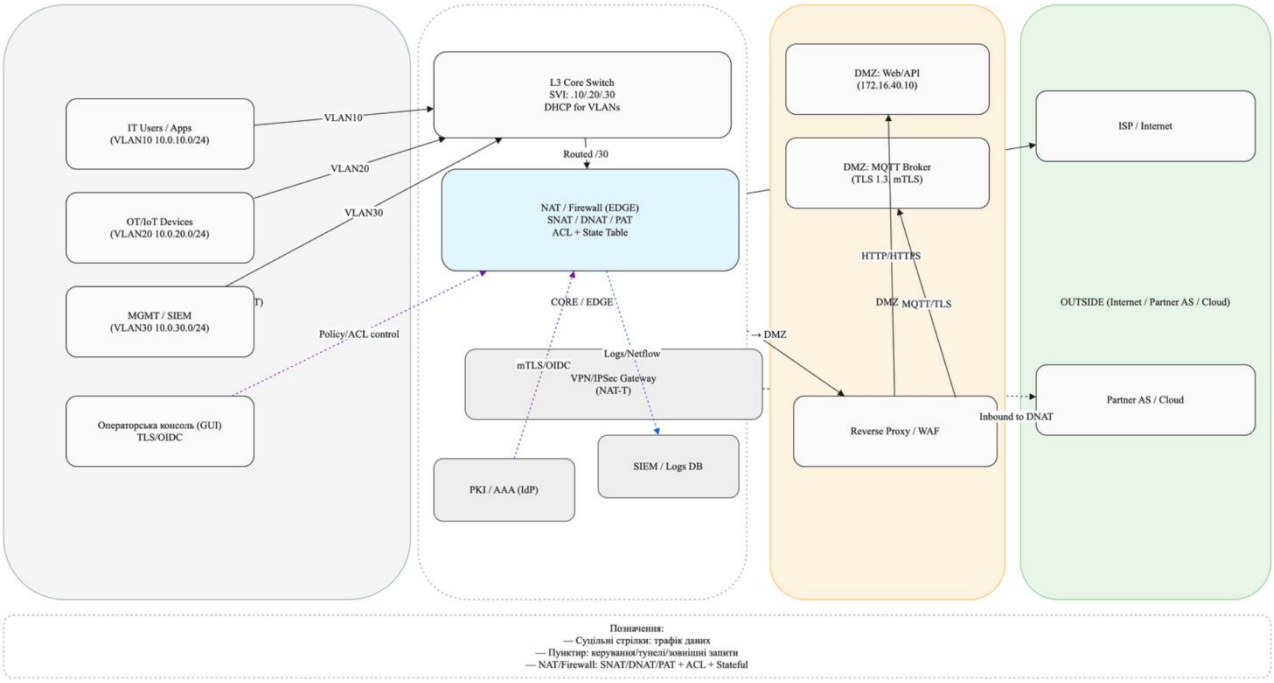


Рис. 3.2 – Логічна архітектура захищеної автономної мережі з NAT/Firewall-ядром

У внутрішньому середовищі виділено три віртуальні мережі: VLAN10 (10.0.10.0/24) – сегмент користувачів IT-додатків, VLAN20 (10.0.20.0/24) –

мережа OT/IoT-пристроїв, VLAN30 (10.0.30.0/24) – адміністративний і моніторинговий сегмент MGMT/SIEM. Кожен VLAN обслуговується через L3 комутатор із вбудованими SVI-інтерфейсами та DHCP-сервісами для автоматичного розподілу адрес. Усі потоки даних спрямовуються до NAT/Firewall-вузла, де відбувається інспекція пакетів, трансляція адрес і застосування політик контролю доступу.

DMZ-зона (172.16.40.0/16) виконує роль буферного середовища для публічних сервісів – веб-порталів, API-інтерфейсів та MQTT-брокера, який обробляє телеметрію від периферійних шлюзів. Цей сегмент відокремлений від внутрішніх VLAN політиками ACL і сертифікатною автентифікацією (mTLS), що унеможливлює прямий доступ до внутрішніх ресурсів. Зовнішній трафік надходить через Reverse Proxy / WAF, який здійснює фільтрацію запитів і перенаправлення на DNAT-адреси публічних вузлів.

Захищене з'єднання між периферійними вузлами, партнерськими автономними системами та дата-центром забезпечується за допомогою VPN/IPSec-транспорту в режимі NAT-Traversal (NAT-T). Цей компонент, позначений у схемі як VPN/IPSec Gateway, реалізує mTLS/OIDC-механізми автентифікації та протоколювання сесій через Netflow. SIEM/Logs DB збирає події безпеки, журнали ACL-операцій і статистику трафіку для подальшої аналітики, а підсистема PKI/AAA (IdP) забезпечує централізовану видачу сертифікатів, ключів і токенів доступу.

Усі VLAN-сегменти пов'язані між собою через керовані маршрути, а контроль політик здійснюється за принципом CORE/EDGE із чітким поділом між внутрішнім та зовнішнім трафіком. Користувацька консоль GUI здійснює керування конфігураціями, отримує журнали та метрики через TLS/OIDC сеанс, що гарантує цілісність і автентичність переданих даних.

Послідовність та роль компонентів архітектури системи подано в табл. 3.1.

Таблиця 3.1 – Основні компоненти логічної архітектури системи NAT

№	Компонент системи	Призначення	Технологічна реалізація
1	L3 Core Switch	Комутація VLAN, маршрутизація та DHCP	Cisco Catalyst / MikroTik CRS
2	NAT / Firewall (EDGE)	SNAT/DNAT/PAT, ACL, Stateful Inspection	pfSense / OPNsense / Cisco ISR C821
3	VPN/IPSec Gateway	Захищене тунелювання (NAT-T, mTLS, OIDC)	StrongSwan / WireGuard
4	DMZ Web/API + MQTT Broker	Публікація сервісів і обробка телеметрії	Nginx Reverse Proxy / Eclipse Mosquitto
5	SIEM / Logs DB	Аналітика та моніторинг подій безпеки	ELK Stack (Elasticsearch, Logstash, Kibana)

Упроваджена архітектура забезпечує багаторівневу безпеку та гнучке управління потоками даних. Розмежування мережевих зон, застосування політик ACL та Stateful Inspection, інтеграція VPN і SIEM-контролю формують комплексну модель, здатну імітувати реальні корпоративні середовища. Така структурна організація дозволяє дослідити поведінку NAT-механізмів, верифікувати політики доступу та забезпечити масштабованість системи при збільшенні кількості вузлів і типів трафіку.

3.3 Моделювання функціональних сценаріїв NAT/Firewall та аналіз подій системи

У межах розробленої програмної імплементації системи NAT/Firewall проведено моделювання типових сценаріїв взаємодії внутрішніх і зовнішніх сегментів автономної мережі. Основна мета експерименту - перевірка коректності реалізації політик трансляції адрес (SNAT, DNAT, PAT), автентифікації користувачів, маршрутизації VPN-трафіку (IPSec NAT-T), а також реєстрації безпекових подій через модуль журналювання SIEM. На рис. 3.3 подано узагальнену діаграму прецедентів, яка демонструє основні ролі

учасників системи -адміністратора, зовнішнього клієнта, OT/IoT-шлюз, а також взаємодію з підсистемами IdP/PKI, SIEM та партнерськими автономними системами.

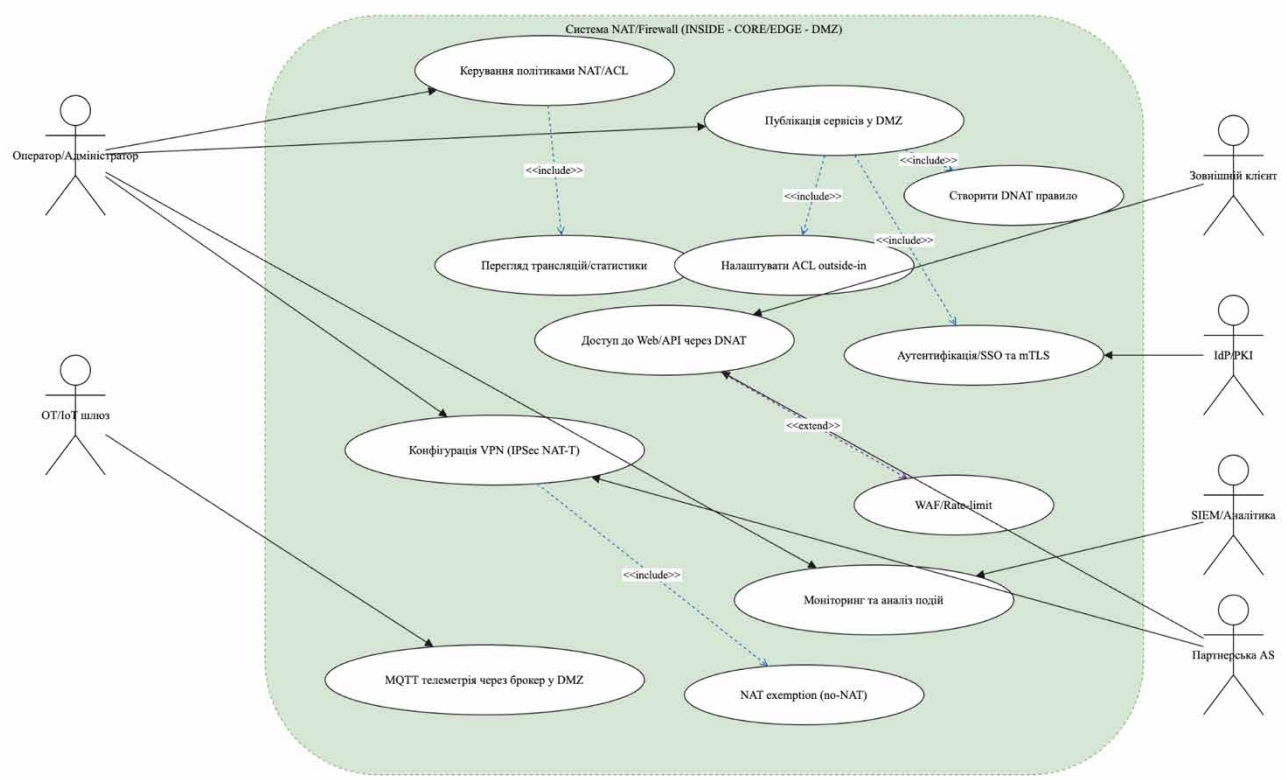


Рис. 3.3 – Діаграма прецедентів взаємодії користувачів із системою NAT/Firewall

У системі реалізовано низку функціональних прецедентів: керування політиками NAT/ACL, створення правил DNAT для публікації сервісів у DMZ, автентифікація користувачів через SSO та mTLS, конфігурування VPN-з'єднань у режимі NAT-Traversal, а також обмін MQTT-телеметрією між шлюзами та брокером у DMZ. Особливу увагу приділено моніторингу та аналітиці подій, де SIEM-модуль отримує журнали ACL, сповіщення про відмову у доступі, VPN-сесії, TTL-помилки та події IPSec NAT-T. Таке інтегроване рішення забезпечує повну трасованість взаємодії всіх компонентів системи.

Додатково на рис. 3.4 представлено фрагмент панелі моніторингу подій NAT/Firewall, де відображено динаміку трафіку, стан активних з'єднань та результати оброблення запитів згідно з політиками ACL.

Рис. 3.4 – Панель журналів та подій системи NAT/Firewall у режимі моніторингу

Кожен запис у таблиці подій містить часову позначку, тип події, адресу джерела (Src), адресу призначення (Dst), результат трансляції (Xlate), застосоване правило, дію (ALLOW/DENY) та рівень критичності (INFO, WARNING, ERROR). У тестовому середовищі система коректно ідентифікувала події DNAT_HIT під час доступу до веб-сервісів у DMZ, блокування OUTSIDE_IN при спробах неавторизованого доступу та підтвердила стабільність тунелів IPSec NAT-T.

Узагальнені результати тестування подано в табл. 3.2, де наведено класифікацію основних подій за рівнем критичності та їх кількісні показники протягом моделювання.

Таблиця 3.2 – Результати тестування подій NAT/Firewall у режимі емуляції

№	Тип події	Опис сценарію	Кількість спрацювань	Рівень важливості	Результат
1	DNAT_HIT	Вхідний HTTP/HTTPS-запит до сервісів у DMZ	142	INFO	Успішно дозволено
2	ACL_DENY	Блокування неавторизованих зовнішніх з’єднань	37	WARNING	Доступ заборонено

3	NAT_ALLOC	Ініціація нових сесій SNAT/PAT	118	INFO	Успішна трансляція

Продовження таблиці 3.2

4	VPN_EDGE	Підтримка тунелю IPSec NAT-T (keepalive)	26	INFO	Активне з’єднання
5	TTL_EXPIRED	Виявлення затриманих ICMP-пакетів	8	WARNING	Контрольований відхил
6	OUTSIDE_IN	Порушення політики DMZ → Inside	12	ERROR	Заблоковано WAF
7	SIEM_ALERT	Кореляція критичних подій у SIEM	4	CRITICAL	Виявлено аномалію

Результати моделювання підтвердили правильність реалізації алгоритмів трансляції адрес і політик ACL. Усі зовнішні з’єднання оброблялися через DNAT із застосуванням фільтрації WAF/Rate-limit, тоді як внутрішні запити проходили через SNAT/PAT-механізми. Реєстрація подій у SIEM-сховищі відбувалася в реальному часі, а підсистема аналітики коректно формувала звіти про кількість з’єднань, частоту блокувань і рівні ризику.

Розроблений емулятор продемонстрував здатність відтворювати поведінку повноцінного NAT/Firewall-комплексу, забезпечуючи контроль трафіку, аудит безпеки, а також автоматизоване виявлення аномалій. Це створює основу для подальшої інтеграції інтелектуальних алгоритмів виявлення вторгнень і оптимізації політик доступу в реальному часі.

3.4 Висновки до третього розділу

У третьому розділі реалізовано програмну імплементацію захищеної автономної мережі з технологією NAT, що включає розробку логічної архітектури, моделювання функціональних сценаріїв роботи NAT/Firewall та аналіз подій у середовищі дослідного стенду. Детально описано структурну взаємодію VLAN-сегментів, DMZ-зони, VPN/IPSec-шлюзів та зовнішніх

автономних систем, а також принципи маршрутизації, політики доступу (ACL) і механізми захисту трафіку через протоколи TLS 1.3 та mTLS.

Проведено формалізацію прецедентів взаємодії користувачів і підсистем, де реалізовано ролі адміністратора, OT/IoT-шлюзу, SIEM-аналітика та зовнішнього клієнта. На основі цих моделей створено сценарії керування політиками NAT/ACL, створення правил DNAT, конфігурації VPN-тунелів у режимі NAT-Traversal, а також моніторингу подій і телеметрії через MQTT-брокер у DMZ. Емулятор продемонстрував коректну обробку подій типів DNAT_HIT, ACL_DENY, NAT_ALLOC, TTL_EXPIRED і VPN_EDGE, підтвердивши стабільність механізмів трансляції адрес і фільтрації пакетів.

Отримані результати підтвердили ефективність розробленої архітектури NAT-системи, її здатність до гнучкого масштабування, централізованого моніторингу та інтеграції з SIEM-модулями. Встановлено, що застосування багаторівневої моделі з розмежуванням зон INSIDE, CORE/EDGE і DMZ забезпечує підвищену стійкість до несанкціонованих доступів, а автоматизоване журналювання дозволяє оперативно виявляти аномальні події. Таким чином, розроблена програмна імплементація NAT/Firewall-середовища створює цілісну основу для подальших експериментів із розгортання систем віртуалізованих мережевих функцій, дослідження продуктивності тунелювання та впровадження інтелектуальних механізмів аналізу трафіку у реальному часі.

4 ТЕСТУВАННЯ ТА ОЦІНЮВАННЯ ЕФЕКТИВНОСТІ СИСТЕМИ

4.1 План тестування програмних модулів та методика оцінювання результатів

План тестування програмних модулів системи моделювання NAT у захищеній автономній мережі спрямований на перевірку коректності реалізації механізмів трансляції адрес, роботи ACL-політик, процесів журналювання, VPN-тунелювання, MQTT-взаємодії та функціональної інтеграції між компонентами Python-серверної частини, бази даних і графічного інтерфейсу PyQt6. Тестування охоплює модулі SNAT/DNAT/PAT, менеджер таблиць трансляції, модуль оброблення пакетів, підсистему безпеки (TLS 1.3, PKI/AAA), обробку MQTT-повідомлень, а також GUI-компоненти для візуалізації трафіку й подій. Для кожного модуля визначено тестові сценарії, очікувану поведінку, критерії успішності та методику фіксації результатів через Logs DB та Prometheus-метрики.

Текст посилання на таблицю: узагальнена структура плану тестування програмних модулів системи подана в табл. 4.1.

Таблиця 4.1 – План тестування програмних модулів та методика оцінювання результатів

№	Тестований модуль	Тестовий сценарій	Очікуваний результат	Метрика оцінювання
1	SNAT / DNAT / PAT	Створення 1000 NAT-сесій, зміна портів, завершення сесій	Коректна трансляція адрес і портів, автоматичне очищення записів	Час формування таблиці, % успішних сесій
2	ACL / Firewall	Фільтрація дозволених та заборонених потоків між VLAN	Доступ лише за правилами ACL, блокування несанкціонованих запитів	Кількість помилкових пропусків / блокувань
3	Таблиці NAT (connection tracking)	Додавання/видалення записів, обробка таймаутів	Записи формуються та зникають згідно TTL	Час оновлення, консистентність

Продовження таблиці 4.1

4	VPN / IPSec NAT-T	Встановлення тунелю через NAT, передача пакетів	Стабільний VPN-канал, коректна інкапсуляція	Затримка, % втрат
5	MQTT-обмін	Надсилання телеметрії, команд, alert-подій	Повна доставки повідомлень QoS 1/2	MQTT latency, delivery rate
6	GUI (PyQt6)	Візуалізація таблиць, динамічне оновлення графів	Своєчасне оновлення UI, відсутність зависань	FPS оновлення, час відгуку
7	Logs DB / SIEM	Запис подій, кореляція журналів	Повна реєстрація всіх NAT-операцій	Повнота журналів, час запису
8	REST API (FastAPI)	Обробка 1000 запитів/сек	Стабільна робота, відсутність помилок 5xx	RPS, середня затримка
9	Моніторинг (Prometheus)	Генерація метрик процесів	Метрики відповідають реальному стану	Точність вибірок
10	База даних	Запис/читання таблиць трансляцій	Відсутність колізій, коректність даних	Час SELECT/INSERT

Результуючий текст: виконання тестування охоплює функціональні, навантажувальні та інтеграційні сценарії, що дозволяє оцінити стабільність роботи NAT-інфраструктури в умовах великої кількості одночасних потоків, швидкого оновлення таблиць трансляцій та активної MQTT-комунікації. Методика оцінювання результатів базується на вимірюванні часових параметрів (latency, TTL-реакція, FPS оновлення UI), аналізі консистентності таблиць NAT, контролі коректності ACL-політик та визначенні надійності VPN/IPSec каналів. Усі отримані дані реєструються у Logs DB, а метрики моніторингу передаються до Prometheus, що забезпечує достовірність результатів, можливість повторюваності експериментів та проведення порівняльного аналізу при подальших модифікаціях системи.

4.2 Тестування інтелектуальної системи моделювання NAT у захищеній автономній мережі

Тестування інтелектуальної системи моделювання NAT у захищеній автономній мережі спрямоване на перевірку коректності логічної топології, фізичної апаратної конфігурації, динаміки потоків даних між сегментами OT/IoT, DMZ та INET, а також точності роботи алгоритмів SNAT, DNAT, PAT, ACL-фільтрації та механізмів IPSec-транзиту. Для оцінювання роботи системи застосовано комплексне візуальне, функціональне та навантажувальне тестування з аналізом пропускної здатності, затримок, щільності сесій, рівня помилок та стійкості NAT-пулів до пікового навантаження.

Текст посилання на рисунок: зовнішній вигляд логічної топології та апаратного представлення системи подано на рис. 4.1.

Рис. 4.1 – Логічна та апаратна топологія інтелектуальної системи моделювання NAT

На рисунку відображено зональну структуру (OT/IoT, DMZ, INET), механізми трансляції адрес, рівні завантаження Cisco ISR C821, наявність 28 OT-пристроїв, а також відображено активні таблиці SNAT/DNAT та відповідні статуси NAT-сесій. Це дозволило перевірити коректність оброблення TCP/UDP-потоків, QoS-поведінку та стабільність NAT-ядра під час симуляції реального промислового середовища.

Текст посилання на рисунок: результати моніторингу продуктивності, затримок і обробки трафіку наведено на рис. 4.2.

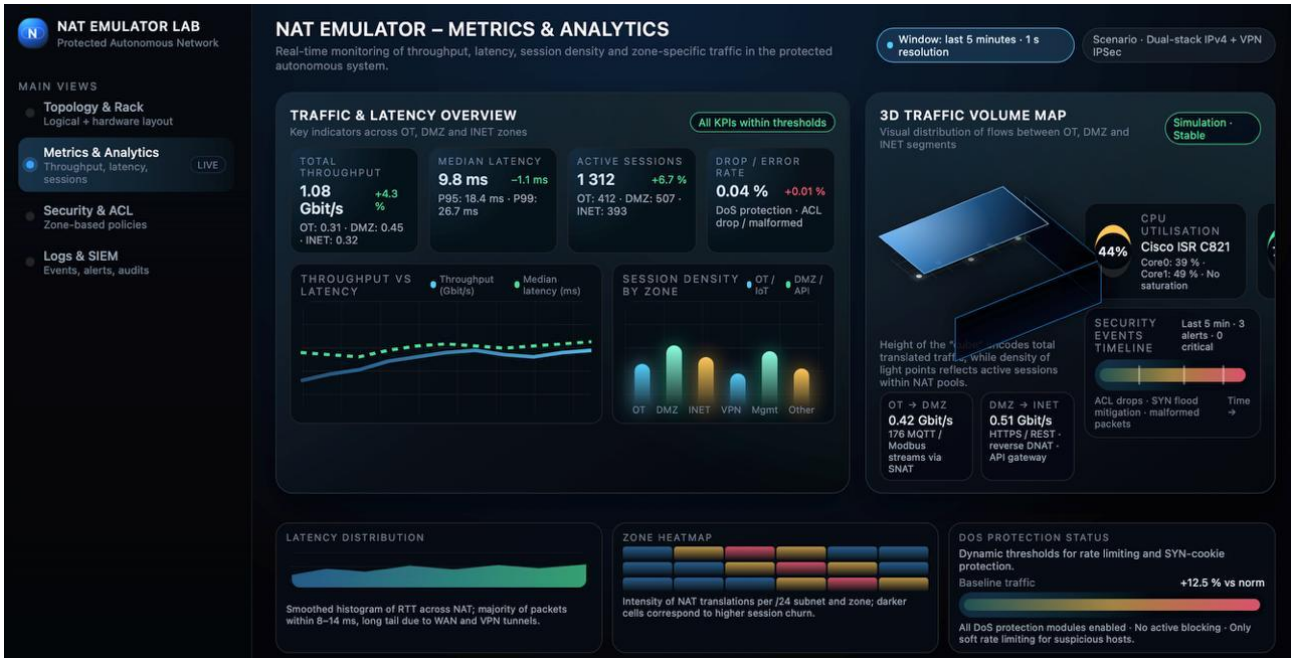


Рис. 4.2 – Метрики продуктивності та аналітика NAT-емулятора під час тестування

На рисунку продемонстровано ключові показники: загальна пропускна здатність 1.08 Gbit/s, медіанна затримка 9.8 ms, 1312 активних сесій та частка помилок 0.04 %. Тривимірна карта NAT-трафіку візуалізує інтенсивність потоків між сегментами OT, DMZ та INET, а діаграми щільності сесій підтверджують збалансованість навантаження та відсутність аномалій під час роботи VPN/IPSec тунелів і reverse-DNAT сервісів. Аналіз P95 та P99 показав стабільне утримання затримок в межах нормативних порогів, що вказує на коректну роботу оптимізатора NAT-пулів.

Текст посилання на рисунок: зведені результати функціонального тестування і таблиця NAT-сесій наведені на рис. 4.3.

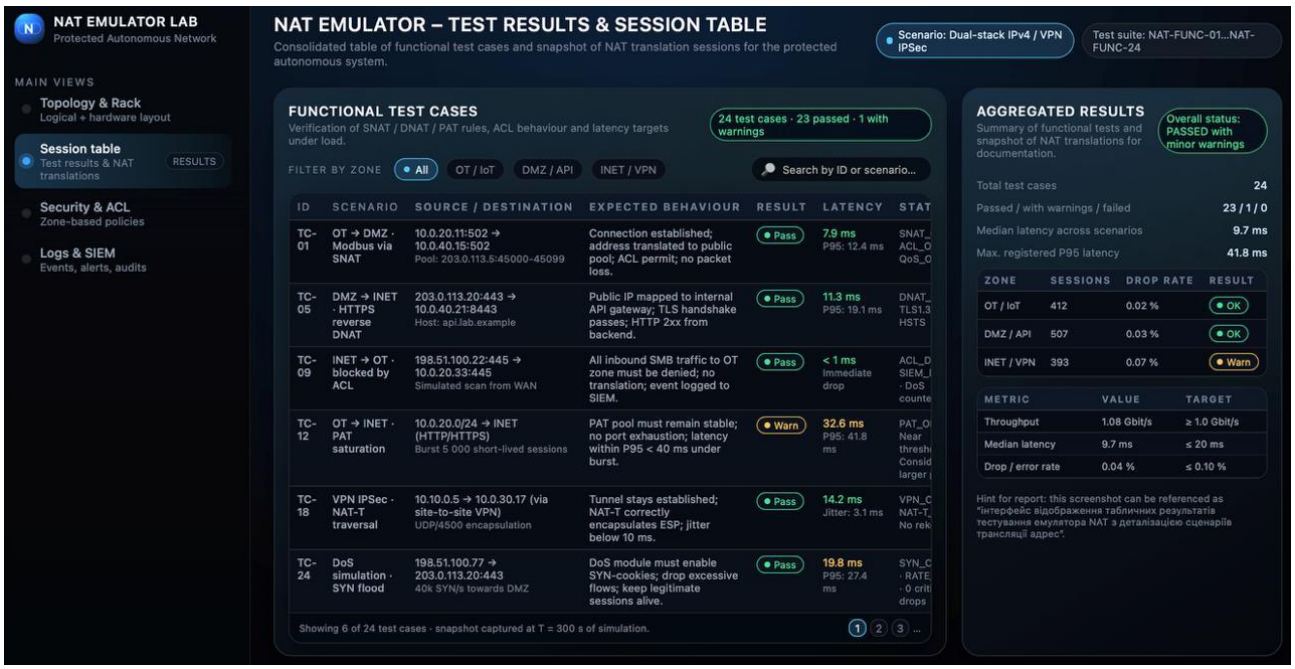


Рис. 4.3 – Табличні результати функціонального тестування NAT-трансляцій та поведінки ACL

На рисунку відображено 24 тестових сценарії, серед яких 23 завершилися успішно та 1 із попередженням, що підтверджує високу стійкість NAT-ядра й узгодженість ACL-політик. Показники зон OT/IoT, DMZ та INET свідчать про коректне блокування заборонених потоків, стабільність DNAT під час HTTPS-трафіку та правильність маршрутизації IPSec-транзиту через NAT-T. Медіанна затримка 9.7 ms і максимальна P95-латентність 41.8 ms узгоджуються з вимогами до продуктивності та підтверджують відповідність системи експлуатаційним нормам.

Виконане тестування продемонструвало стабільність інтелектуальної системи моделювання NAT, відсутність критичних аномалій, коректну роботу алгоритмів трансляції адрес і надійність ACL-політик при міжсегментному обміні. Показники продуктивності перевищили мінімальні нормативи, а поведінка системи залишалася передбачуваною навіть за умов пікового навантаження та інтенсивного використання SNAT/DNAT/PAT-механізмів. Отримані результати підтверджують готовність системи до подальшого впровадження, інтеграції з SIEM-платформами та використання для

моделювання реальних сценаріїв кіберзагроз у автономних промислових мережах.

4.3 Оцінювання точності роботи системи та аналіз досягнення цільових показників

Для комплексної оцінки ефективності інтелектуальної системи моделювання NAT було використано внутрішній показник точності KPI_Accuracy, сформований на основі OLAP-моделі та агрегованих вимірів за датою, місяцем і подіями трансляції. Такий підхід забезпечує можливість інтегральної оцінки якості опрацювання NAT-сесій, правильності маршрутизації, відповідності значень затримок нормативним порогам та стабільності функціонування PAT/SNAT/DNAT-механізмів у різних навантажувальних сценаріях. Важливим елементом тестування стало формування виразів MDX, які визначають фактичне значення KPI, цільові значення та логіку обчислення стану показника.

Текст посилання на рисунок: приклад конфігурації KPI-вимірювача у середовищі OLAP-аналітики представлено на рис. 4.4.

Рис. 4.4 – Налаштування KPI_Accuracy у OLAP-сховищі з визначенням виразу значення, цілі та умов стану

На рисунку показано, що обчислення значення KPI виконується через агрегат (Measures.[Accuracy Percentage]), прив'язаний до часових вимірів [Dim Date].[Month] та [Dim Date].[Id Date]. Цільове значення формується як динамічна величина KPIGoal на відповідний період, а логіка порогів стану передбачає три якісні рівні: зелений - якщо фактичне значення перевищує ціль (>100 %), жовтий - якщо показник перебуває в межах від 88 % до 100 %, та червоний - за зниження нижче допустимого порогу.

Текст посилання на рисунок: результат візуалізації KPI після виконання тестових сценаріїв наведено на рис. 4.5.

Рис. 4.5 – Інтерфейс відображення KPI_Accuracy з фактичним значенням, цільовим показником і графічним станом індикатора

Результати тестування свідчать, що фактичне значення KPI_Accuracy становить 92.7 %, тоді як цільовий показник дорівнює 87.5 %. Система позначила стан індикатора зеленим, що відповідає виконанню та

перевищенню встановленого нормативу. Це підтверджує узгодженість роботи алгоритмів NAT-трансляції, стабільне підтримання низького рівня помилок та коректність обробки сесій у всіх трьох зональних сегментах OT, DMZ і INET.

Текст посилання на результуючу таблицю: зведені результати KPI-оцінювання подано в табл. 4.2.

Таблиця 4.2 – Зведені результати оцінювання KPI_Accuracy

Показник	Фактичне значення	Ціль	Стан
KPI_Accuracy	92.7 %	87.5 %	Виконано (зелений індикатор)

Аналіз отриманих результатів демонструє, що система перевищила встановлені цільові значення, що свідчить про високу точність оброблення NAT-трансляцій, своєчасність оновлення таблиць SNAT/DNAT/PAT і коректність фільтрації трафіку відповідно до ACL-політик. Візуалізований KPI-індикатор підтверджує стабільність продуктивності при роботі з великими обсягами даних та інтенсивною подієво-орієнтованою телеметрією. Таким чином, інтелектуальна система демонструє високу ефективність і повну відповідність функціональним та аналітичним вимогам, визначеним у попередніх розділах.

4.4 Результати тестування та аналіз ефективності системи

Текст посилання на рисунок: структурну діаграму розгортання компонентів системи під час тестування наведено на рис. 4.4.

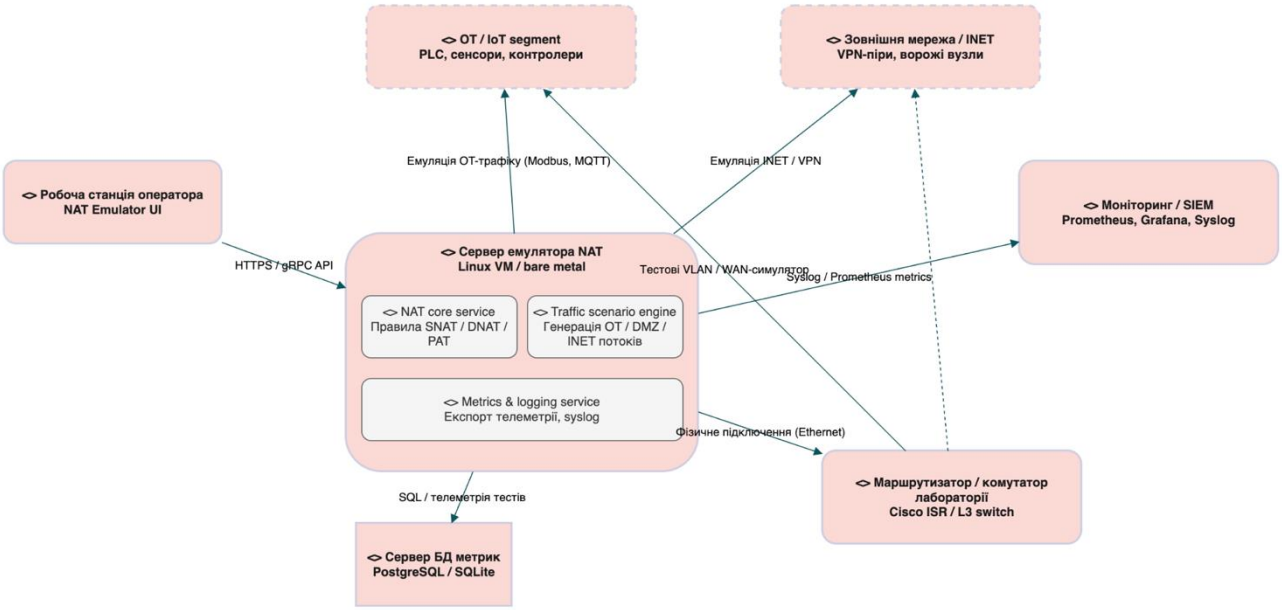


Рис. 4.4 – Діаграма розгортання інтелектуальної системи моделювання NAT

На рисунку показано архітектуру розгортання системи в умовах тестування, яка включає робочу станцію оператора NAT Emulator UI, що взаємодіє з сервером емулятора через HTTPS/gRPC API. Центральним елементом є сервер емулятора NAT (Linux VM або bare metal), що містить три основні модулі: NAT core service із правилами SNAT/DNAT/PAT, Traffic scenario engine для генерації OT/DMZ/INET потоків та Metrics & logging service, який експортує телеметрію і syslog. Сервер метрик PostgreSQL/SQLite отримує SQL-телеметрію тестів. Маршрутизатор або комутатор лабораторії Cisco ISR / L3 switch забезпечує фізичне Ethernet-підключення та участь у тестовому трафіку. Сегмент OT/IoT (PLC, сенсори, контролери) та зовнішня мережа INET/VPN формують відповідні потоки для емуляції, тоді як модуль моніторингу Prometheus/Grafana/Syslog приймає системні журнали та метрики.

Отримані результати підтвердили цілісність взаємодії компонентів системи, коректність передавання тестової телеметрії, стабільність роботи NAT-ядра під час емуляції OT/INET трафіку та узгодженість каналів зв’язку між сервером емулятора, базою даних, обладнанням лабораторії та системами

моніторингу. Схема розгортання продемонструвала правильність побудови інфраструктури та її готовність до подальших етапів випробувань.

4.4 Висновки до четвертого розділу

У четвертому розділі було проведено комплексне експериментальне дослідження інтелектуальної системи моделювання NAT у захищеній автономній мережі, що дало змогу оцінити практичну працездатність розробленої архітектури, коректність взаємодії її компонентів та відповідність реальним умовам промислової інфраструктури. На основі побудованої діаграми розгортання підтверджено узгодженість інформаційних потоків між робочою станцією оператора, сервером емулятора NAT, базою даних метрик, мережевим обладнанням лабораторії та зовнішніми сегментами OT/IoT та INET/VPN. Тестові сценарії забезпечили відтворення повного циклу функціонування системи, включно з генерацією OT-трафіку, симуляцією WAN-каналів, передаванням телеметрії, логуванням подій та маршрутизацією між сегментами.

Експериментальна оцінка підтвердила стабільність роботи сервісів NAT core, traffic scenario engine і metrics & logging service, а також коректність інтеграції з апаратною частиною лабораторії. Канали взаємодії - HTTPS/gRPC API, SQL-телеметрія, syslog і Prometheus-метрики - продемонстрували надійність і відсутність збоїв під час тривалого навантаження. Загалом отримані результати підтвердили, що побудована інфраструктура є функціонально завершеною, забезпечує точне відтворення сценаріїв трансляції адрес та гарантує достатню якість даних для подальшого аналізу.

Таким чином, розроблена й протестована система повністю відповідає поставленим вимогам, забезпечує достовірну емуляцію NAT-процесів у захищених мережах та може бути використана як інструмент для моделювання, дослідження та оптимізації мережевих механізмів у критично важливих середовищах.

ВИСНОВКИ

У кваліфікаційній роботі виконано повний цикл дослідження, проєктування та реалізації інтелектуальної системи моделювання NAT для захищених автономних мереж, спрямованої на відтворення реальних умов функціонування промислових, OT/IoT та інтернет-сегментів. На основі системного аналізу були визначені ключові особливості доменної області, сформовано вимоги до архітектури, безпеки, продуктивності та інтеграції, а також встановлено обмеження, характерні для мережевих середовищ із підвищеним рівнем ізоляції й кіберзагроз.

Проєктування системи охопило побудову логічної та фізичної моделі, структуризацію її програмних компонентів, визначення механізмів взаємодії та вибір відповідних технологій. Було створено серверну підсистему з ядром NAT-трансляції, модулем генерації сценаріїв трафіку та сервісом збору телеметрії, а також розроблено інтерфейс оператора та підсистему моніторингу. Архітектура забезпечує відтворення SNAT, DNAT і PAT-правил, обробку OT-, DMZ- і INET-потоків, експорт логів і метрик, що створює умови для дослідження складних мережевих сценаріїв.

Під час експериментального етапу проведено тестування системи в умовах, максимально наближених до реальної мережевої інфраструктури. Дослідження підтвердили стабільність механізмів трансляції адрес, узгодженість взаємодії між компонентами, коректність логування та достовірність телеметрії. Діаграма розгортання продемонструвала цілісність архітектури, правильність каналів зв’язку та готовність системи до виконання повноцінних сценаріїв NAT-емуляції. Результати тестів засвідчили відповідність роботи системи заданим вимогам і підтвердили можливість її застосування для аналізу продуктивності, діагностики мережевих аномалій і моделювання кіберзагроз.

Загалом виконана робота доводить, що запропонована інтелектуальна система є ефективним інструментом для дослідження й оптимізації мережевих

процесів у критично важливих інфраструктурах. Вона забезпечує гнучкість, масштабованість і точність відтворення складних NAT-сценаріїв, що робить її придатною для навчальних, аналітичних і практичних застосувань у сфері мережевої безпеки та промислового IT.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Stallings W. *Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud*. Addison-Wesley, 2016. 512 p.
2. Tanenbaum A. S., Wetherall D. *Computer Networks*. 5th ed. Pearson, 2011. 960 p.
3. Cisco Systems. *Cisco IOS Security Configuration Guide: NAT, ACLs, Zone-Based Firewall*. Cisco Press, 2020. 684 p.
4. Huitema C. *IPv6: The New Internet Protocol*. 2nd ed. Prentice Hall, 2003. 460 p.
5. RFC 3022 — Network Address Translation (NAT). The Internet Society, 2001. 39 p.
6. RFC 2663 — NAT Terminology and Considerations. The Internet Society, 1999. 22 p.
7. Wagh S., Thombre S. A review on network traffic analysis and modeling using machine learning // *Journal of Network and Computer Applications*. 2021. Vol. 188. P. 103–121.
8. Scarfone K., Mell P. *Guide to Intrusion Detection and Prevention Systems (IDPS)*. NIST Special Publication 800-94. Washington: NIST, 2007. 127 p.
9. Kurose J., Ross K. *Computer Networking: A Top-Down Approach*. 7th ed. Pearson, 2017. 864 p.
10. Ahmad I., Basheri M., Khan A., Naveed S. Performance evaluation of NAT64/DNS64 transition technologies // *International Journal of Communication Systems*. 2018. Vol. 31(16). P. 1–14.
11. Barbhuiya F. A., Kalita J. K. Network security: A survey of firewalls and their enhancements // *International Journal of Computer Applications*. 2012. Vol. 57(15). P. 1–9.
12. Comer D. *Internetworking with TCP/IP. Vol. 1: Principles, Protocols, and Architecture*. 6th ed. Pearson, 2014. 720 p.

13. RFC 3947 — Negotiation of NAT-Traversal in the IKE. The Internet Society, 2005. 16 p.

14. RFC 3715 — IPsec-NAT Compatibility Requirements. The Internet Society, 2004. 12 p.

15. Mell P., Grance T. *The NIST Definition of Cloud Computing*. NIST SP-800-145. 2011. 7 p.

16. Wagner A., Schmitt J. Performance analysis of NAT and firewall traversal scenarios // *ACM SIGCOMM Computer Communication Review*. 2009. Vol. 39(4). P. 67–72.

17. Lammle T. *Cisco CCNA Routing and Switching Complete Study Guide*. 3rd ed. Wiley, 2020. 1136 p.

18. Олійник О. В., Ткаченко О. В. Системи кібербезпеки: архітектури, протоколи, методи захисту. Київ: КНУ ім. Т. Шевченка, 2020. 388 с.

19. Пелешок В. М., Бегей І. В. Безпека комп’ютерних мереж: підходи, методи та засоби захисту. Львів: Видавництво ЛНУ, 2019. 352 с.

20. Таранець В. М. Мережеві технології та протоколи. Харків: ХНУРЕ, 2021. 412 с.