ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ТЕЛЕКОМУНІКАЦІЙ
КАФЕДРА ТЕЛЕКОМУНІКАЦІЙНИХ СИСТЕМ ТА МЕРЕЖ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «РОЗРОБКА СИСТЕМИ АНАЛІЗУ QOS (ЯКОСТІ ОБСЛУГОВУВАННЯ) У МУЛЬТИСЕРВІСНИХ ТЕЛЕКОМУНІКАЦІЙНИХ МЕРЕЖАХ»

на здобуття освітнього ступеня магістра
зі спеціальності 172 Електронні комунікації та радіотехніка
(код, найменування спеціальності)
освітньо-професійної програми Телекомунікаційні системи та мережі
(назва)

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

_____ Нікіта КУЛІКОВСЬКИЙ
(підпис) *(Ім'я ПРІЗВИЩЕ)*

Виконав: здобувач вищої освіти гр. ТСДМ-61
__Нікіта КУЛІКОВСЬКИЙ __
Ім'я, ПРІЗВИЩЕ

Керівник: Катерина ТРЕНЬОВА
PhD *Ім'я, ПРІЗВИЩЕ*

Рецензент: _____
Ім'я, ПРІЗВИЩЕ

Київ 2025

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
Навчально-науковий інститут Телекомунікацій

Кафедра ___Телекомунікаційних систем та мереж_____
Ступінь вищої освіти __магістр_____
Спеціальність ___172 Електронні комунікації та радіотехніка _____
Освітньо-професійна програма _____Телекомунікаційні системи та мережі_____

ЗАТВЕРДЖУЮ
Завідувач кафедри __ТСМ_____
_____ Віктор ЗАЇКА
«____» _____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Куліковський Нікіта Володимирович
(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: Розробка системи аналізу QoS (якості обслуговування) у мультисервісних телекомунікаційних мережа
керівник кваліфікаційної роботи Катерина ТРЕНЬОВА, Phd (доктор філософії)
(Ім'я, ПРІЗВИЩЕ науковий ступінь, вчене звання)
затверджені наказом Державного університету інформаційно-комунікаційних технологій від «30» жовтня 2024 р. №467

2. Строк подання кваліфікаційної роботи «___» _____ 2024 р.

3. Вихідні дані до кваліфікаційної роботи:
   1. Телекомунікаційна модель корпоративної мережі.
   2. Комутатори та маршрутизатори Cisco, VoIP, IPTV, FTP-трафік.
   3. Маршрутизатор MikroTik як шлюз.
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
   1. Аналіз сучасних моделей QoS у мультисервісних мережах.
   2. Опис параметрів QoS і їх впливу на сервіси.
   3. Розроблення та моделювання мережі в GNS3.
   4. Налаштування та оцінювання ефективності QoS на Cisco і MikroTik
5. Перелік ілюстративного матеріалу: ___презентація на____слайдах_____

6. Дата видачі завдання «___» _____ 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз літератури та формування теоретичної бази	25.10 – 28.10	
2	Опис параметрів QoS та моделей забезпечення	29.10 – 02.11	
3	Розробка структури мережі та схеми моделювання	03.11 – 06.11	
4	Побудова мережі в GNS3 та базова конфігурація	07.11 – 12.11	
5	Налаштування QoS на Cisco та MikroTik	13.11 – 18.11	
6	Моніторинг, тестування, збір статистики	19.11 – 22.11	
7	Аналіз результатів, формування висновків	23.11-26.11	
8	Оформлення роботи та підготовка до захисту	27.11-01.12	

Здобувач(ка) вищої освіти _____ Нікіта КУЛІКОВСЬКИЙ
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

Керівник
кваліфікаційної роботи _____ Катерина ТРЕНЬОВА
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра: 86 стор., 16 рис., 13 табл., 15 джерел.

Мета роботи – підвищення якості обслуговування мережевого трафіку в корпоративній мережі шляхом розроблення, моделювання та впровадження механізмів QoS із використанням обладнання Cisco та маршрутизатора MikroTik як шлюзу в зовнішню мережу.

Об’єкт дослідження – процес забезпечення якості сервісів у корпоративних мережах з мультимедійним та змішаним трафіком.

Предмет дослідження – методи, моделі та механізми управління мережевим трафіком і параметрами QoS у мережах на базі Cisco та MikroTik.

Короткий зміст роботи: У роботі проведено аналіз сучасних підходів до забезпечення QoS у мультисервісних мережах, розглянуто принципи класифікації, пріоритизації, чергування та управління трафіком. Побудовано лабораторний стенд із використанням маршрутизаторів Cisco та MikroTik, виконано базову та розширену конфігурацію мережевих пристроїв, налаштовано механізми QoS для VoIP, відеотрафіку та службових даних. Реалізовано політики класів трафіку, пріоритетного обслуговування, обмеження пропускної здатності та управління чергами. Проведено моделювання змішаного навантаження, виконано тестування параметрів затримки, джитеру та втрат пакетів. Проаналізовано вплив застосованих QoS-механізмів на якість роботи мережі та функціонування MikroTik як кінцевого шлюзу.

У результаті встановлено, що впроваджені механізми QoS забезпечують стабільну передачу критичного мультимедійного трафіку, знижують імовірність перевантаження каналу доступу та підвищують загальну ефективність роботи корпоративної мережі.

КЛЮЧОВІ СЛОВА: **QoS, Cisco, MikroTik, трафік, черги, пріоритизація, VoIP, маршрутизація, мультимедійні сервіси, пропускна здатність**

ABSTRACT

Text part of the master's qualification work: 86 pages, 16 pictures, 13 table, 15 sources.

The purpose of the work is to improve the quality of network traffic service in a corporate network by developing, modeling, and implementing QoS mechanisms using Cisco equipment and a MikroTik router as a gateway to an external network.

Object of research – the process of ensuring service quality in corporate networks with multimedia and mixed traffic.

Subject of research – methods, models, and mechanisms of network traffic management and QoS parameters in networks based on Cisco and MikroTik equipment.

Summary of the work: The work analyzes modern approaches to ensuring QoS in multiservice networks and examines the principles of traffic classification, prioritization, queuing, and management. A laboratory testbed was built using Cisco and MikroTik routers, on which both basic and advanced configurations were performed. QoS mechanisms for VoIP, video traffic, and service data were configured. Traffic class policies, priority servicing, bandwidth limitation, and queue management were implemented. Mixed traffic load modeling was carried out, followed by testing of delay, jitter, and packet loss parameters. The impact of the applied QoS mechanisms on overall network performance and the operation of MikroTik as the final gateway was evaluated.

As a result, it was determined that the implemented QoS mechanisms ensure stable transmission of critical multimedia traffic, reduce the likelihood of access channel congestion, and improve the overall efficiency of the corporate network.

KEYWORDS: QoS, Cisco, MikroTik, traffic, queues, prioritization, VoIP, routing, multimedia services, bandwidth

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
Навчально-науковий інститут телекомунікацій
ПОДАННЯ
ГОЛОВІ ЕКЗАМЕНАЦІЙНОЇ КОМІСІЇ
ЩОДО ЗАХИСТУ КВАЛІФІКАЦІЙНОЇ РОБОТИ
на здобуття освітнього ступеня магістра

Направляється здобувач Куліковський Н.В. до захисту кваліфікаційної роботи
(прізвище та ініціали)
за спеціальністю 172 Електронні комунікації та радіотехніка
(код, найменування спеціальності)
освітньо-професійної програми Телекомунікаційні системи та мережі
(назва)
на тему: «Розробка системи аналізу QoS (якості обслуговування) у мультисервісних телекомунікаційних мережах»

Кваліфікаційна робота і рецензія додаються.

Директор ННІТ _____ Владислав КРАВЧЕНКО
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

Висновок керівника кваліфікаційної роботи

Здобувач Куліковський Нікіта Володимирович показав гарну теоретичну та інженерну підготовку, уміння володіти новими комп'ютерними технологіями, користуватися навчальною, довідковою і науково-технічною літературою. Працюючи над завданнями, які були поставлені керівником, проявив ініціативність, сумлінність та хист до наукової роботи.

Все це дозволяє оцінити виконану кваліфікаційну роботу здобувача Куліковського Нікіти Володимировича на оцінку «відмінно» та присвоїти йому кваліфікацію магістр з електронних комунікацій та радіотехніки.

Керівник кваліфікаційної роботи _____ Катерина Треньова
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

«____» _____ 20___ року

Висновок кафедри про кваліфікаційну роботу

Кваліфікаційна робота розглянута. Здобувач Куліковський Н.В. допускається до захисту даної роботи в Екзаменаційній комісії.

Завідувач кафедри телекомунікаційних систем та мереж _____ Віктор ЗАЇКА
(назва) *(підпис)* *(Ім'я, ПРІЗВИЩЕ)*

«____» _____ 20___ року

ВІДГУК РЕЦЕНЗЕНТА
на кваліфікаційну роботу

на здобуття освітнього ступеня бакалавра (магістра)

здобувача(ки) вищої освіти _____
(прізвище, ім'я, по батькові)
на тему «_____»

Актуальність.

_____
_____
_____

Позитивні сторони.
1.
2.
3.

Недоліки.
1.
2.
Відзначені зауваження не впливають на загальну позитивну оцінку кваліфікаційної роботи бакалаврської (магістерської).

Висновок: *кваліфікаційна робота на здобуття ступеня бакалавра (магістра) заслуговує оцінку "_____", а здобувач(ка) _____ заслуговує присвоєння кваліфікації: ……..*

Рецензент:
науковий ступінь, вчене звання _____ *підпис* _____ *Ім'я, ПРІЗВИЩЕ*

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ ТА ТЕРМІНІВ 10
ВСТУП 12
РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ЗАБЕЗПЕЧЕННЯ ЯКОСТІ ОБСЛУГОВУВАННЯ В МУЛЬТИСЕРВІСНИХ ТЕЛЕКОМУНІКАЦІЙНИХ МЕРЕЖАХ 15
1.1. Поняття мультисервісних телекомунікаційних мереж та їх особливості 15
1.2. Основні параметри QoS і їхній вплив на роботу сервісів 19
1.3. Класифікація трафіку в мультисервісних мережах 22
1.4. Механізми управління та пріоритизації мережевого трафіку 26
1.5. Моделі забезпечення QoS у сучасних телекомунікаційних системах 29
РОЗДІЛ 2. МЕТОДИ ТА ЗАСОБИ МОНІТОРИНГУ ЯКОСТІ ОБСЛУГОВУВАННЯ 34
2.1. Загальні принципи моніторингу параметрів QoS 34
2.2. Інструменти збору мережевої статистики: SNMP, NetFlow, sFlow 37
2.3. Аналіз мережевого трафіку засобами Wireshark 40
2.4. Моніторинг у реальному часі: Prometheus, Grafana, Zabbix 44
2.5. Методи статистичної обробки, фільтрації та візуалізації показників QoS 49
Висновки до розділу 2 53
РОЗДІЛ 3. ПОБУДОВА МОДЕЛІ МУЛЬТИСЕРВІСНОЇ МЕРЕЖІ В СЕРЕДОВИЩІ GNS3 54
3.1. Обґрунтування вибору програмного середовища GNS3 54
3.2 Розроблення логічної структури мережі 57
3.3 Процедура побудови моделі мережі в середовищі GNS3 60
3.4 Налаштування мережевих пристроїв у середовищі моделювання 63
3.5 Перевірка працездатності побудованої мережі 74
3.6 Налаштування QoS 77
3.6.1 Теоретичні засади застосування QoS у побудованій мережі 77
3.6.2 Реалізація механізмів QoS на маршрутизаторах Cisco R1 та R2 80
3.6.3 Перевірка роботи QoS, аналіз статистики та результати тестування 83
3.6.4 Висновки щодо ефективності впровадження QoS у мережі 86

Висновок до розділу 3 90
ВИСНОВКИ 93
ПЕРЕЛІК ПОСИЛАНЬ 95
ДОДАТКИ 96
ДОДАТКОК А – Перевірка пінгування між маршрутизаторами R1 і R2 96
ДОДАТКОК Б – Перевірка пінгування на Client1 і Client2 97
ДОДАТКОК В – Налаштування QoS на маршрутизаторах R1 і R2 98
ДОДАТКОК Г – Перевірка зміни затримок на Client1 і Client2 99
ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація) 100

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ ТА ТЕРМІНІВ

ACL (Access Control List) — список контролю доступу, що використовується для фільтрації та класифікації трафіку в маршрутизаторах Cisco.

ARP (Address Resolution Protocol) — протокол визначення MAC-адреси за IP-адресою в локальній мережі.

CBWFQ (Class-Based Weighted Fair Queuing) — механізм чергування Cisco, що забезпечує розподіл пропускної здатності між класами трафіку.

Cloud — у GNS3 елемент, що слугує інтерфейсом для взаємодії віртуальної топології з реальною мережею комп'ютера.

DSCP (Differentiated Services Code Point) — поле у заголовку IP-пакета, що містить мітку класу обслуговування в моделі DiffServ.

DiffServ (Differentiated Services) — архітектура реалізації QoS, що забезпечує класифікацію й пріоритезацію трафіку у великих мережах.

EF (Expedited Forwarding) — клас обслуговування з найвищим пріоритетом для трафіку реального часу (RTP/VoIP), визначений у RFC 3246.

FTP (File Transfer Protocol) — протокол передавання файлів, використовується для навантажувальних тестів.

GNS3 (Graphical Network Simulator 3) — програмний комплекс для побудови віртуальних мереж і моделювання роботи мережевого обладнання.

IP (Internet Protocol) — протокол мережного рівня, що використовується для передавання пакетів у TCP/IP-мережах.

IPTV (Internet Protocol Television) — система передавання телевізійного сигналу через IP-мережі, чутлива до втрат і затримок.

ISP (Internet Service Provider) — провайдер Інтернету.

LLQ (Low-Latency Queuing) — механізм пріоритетного обслуговування трафіку на Cisco, що гарантує мінімальні затримки для RTP/VoIP.

MAC (Media Access Control) — апаратна адреса мережевого інтерфейсу.

MQC (Modular QoS CLI) — модульна система команд у Cisco IOS для конфігурації QoS (ACL → class-map → policy-map).

NAT (Network Address Translation) — механізм трансляції IP-адрес, що використовується на MikroTik для виходу клієнтів у Інтернет.

QoS (Quality of Service) — механізми забезпечення якості обслуговування, що керують пріоритетами, пропускною здатністю та затримками трафіку.

R1, R2 — умовні позначення маршрутизаторів Cisco у побудованій топології.

RTP (Real-Time Protocol) — протокол передавання даних реального часу (VoIP, відеоконференції), чутливий до затримок і джитеру.

SLA (Service Level Agreement) — узгоджений рівень якості сервісу між користувачем і провайдером.

TCP (Transmission Control Protocol) — транспортний протокол з контрольованою доставкою пакетів.

UDP (User Datagram Protocol) — транспортний протокол без встановлення з’єднання; використовується в RTP, VoIP та IPTV.

VoIP (Voice over IP) — технологія передавання голосового зв’язку через IP-мережі.

WRED (Weighted Random Early Detection) — алгоритм керування чергами, що запобігає переповненню буферів.

Windows 10 Client — кінцеві пристрої користувачів, що генерують трафік у моделі.

ВСТУП

Стрімкий розвиток сучасних інформаційно-комунікаційних технологій супроводжується зростанням обсягів переданих даних, ускладненням мережевих інфраструктур та розширенням спектра сервісів, що функціонують у телекомунікаційних системах. Голосова й відеокомунікація, потокове мультимедійне мовлення, хмарні сервіси, передавання файлів, керування IoT-пристроями — усе це формує мультимедійні середовища, які працюють у межах єдиної мережевої інфраструктури. Такі мережі визначаються як мультисервісні, а їх ефективне функціонування неможливе без реалізації механізмів забезпечення якості обслуговування (Quality of Service, QoS).

Актуальність теми зумовлена зростанням вимог до стабільності та передбачуваності роботи мереж під час передавання мультимедійного трафіку. У сервісах, чутливих до затримки та джиттера (VoIP, відеоконференції, IPTV), навіть незначні коливання параметрів призводять до суттєвого погіршення якості послуг: затримок, артефактів відео, спотворень звуку та часткової або повної втрати даних. Тому постає необхідність у створенні систем, що дозволяють проводити аналіз, моніторинг і оптимізацію параметрів QoS у реальному часі, виявляти перевантаження та прогнозувати поведінку мережі в умовах змішаного навантаження.

Об’єктом дослідження є процеси забезпечення якості обслуговування в мультисервісних телекомунікаційних мережах.

Предметом дослідження є методи, механізми та алгоритми аналізу QoS, а також засоби моделювання та моніторингу мультимедійного трафіку.

Метою магістерської роботи є розроблення комплексної системи аналізу QoS у мультисервісних телекомунікаційних мережах, що забезпечує моніторинг, класифікацію, пріоритизацію та оцінювання ефективності передавання VoIP-, IPTV- та FTP-трафіку за різних умов навантаження.

Для досягнення мети вирішуються такі завдання:

- дослідити теоретичні основи QoS та класифікувати типи мережевого трафіку;
- проаналізувати існуючі механізми забезпечення QoS (LLQ, CBWFQ, WFQ, RED, DSCP-класифікація);
- розробити структурну модель мультисервісної мережі з мультимедійним трафіком;
- реалізувати симуляційну модель мережі в програмному середовищі GNS3;
- налаштувати механізми QoS на маршрутизаторах Cisco/MikroTik та забезпечити пріоритизацію трафіку;
- організувати систему моніторингу параметрів QoS із використанням NetFlow, SNMP, Wireshark та Grafana;
- провести експериментальні дослідження параметрів якості з QoS та без її застосування;
- виконати порівняльний аналіз отриманих результатів і сформувати рекомендації з оптимізації QoS.

Методи дослідження включають комп’ютерне моделювання мережі, статистичний аналіз параметрів потоків, математичне оцінювання QoS-показників, експериментальне тестування та аналіз захоплених пакетів у середовищах Wireshark і NetFlow.

Наукова новизна полягає у створенні інтегрованої моделі системи аналізу QoS, що поєднує механізми пріоритизації мультимедійного трафіку, автоматизований моніторинг і засоби візуалізації показників якості у єдиному комплексі. Запропоновано методику оцінювання якості мультимедійного трафіку при змішаному навантаженні (VoIP + IPTV + FTP), що підвищує точність аналізу ефективності конфігурацій QoS.

Практичне значення роботи полягає у можливості застосування розробленої моделі під час проєктування політик QoS у мережах операторського класу, корпоративних інфраструктурах, центрах обробки

даних та навчальних лабораторіях. Результати дослідження можуть бути використані для підвищення якості мультимедійних сервісів і мінімізації ризику деградації мережі під час пікових навантажень.

Структурно магістерська робота складається зі вступу, чотирьох розділів, висновків, списку використаних джерел та додатків. У першому розділі наведено теоретичні основи QoS; у другому — методи та інструменти моніторингу параметрів якості; у третьому — побудову та моделювання мережевої топології у середовищі GNS3 та аналіз отриманих результатів. Роботу завершує формулювання висновків і рекомендацій.

РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ЗАБЕЗПЕЧЕННЯ ЯКОСТІ ОБСЛУГОВУВАННЯ В МУЛЬТИСЕРВІСНИХ ТЕЛЕКОМУНІКАЦІЙНИХ МЕРЕЖАХ

1.1. Поняття мультисервісних телекомунікаційних мереж та їх особливості

Мультисервісні телекомунікаційні мережі є ключовою основою сучасних інформаційно-комунікаційних систем [1], оскільки вони забезпечують конвергентну передачу різнорідних сервісів — голосових викликів, потокового відео, мультимедійних трансляцій, корпоративних даних, телеметрії та керувальних повідомлень — через єдину IP-інфраструктуру [2]. На відміну від традиційних мереж попередніх поколінь, де сервіси часто жили в ізольованих технологічних сегментах (TDM, ATM) [3], мультисервісні мережі об’єднують усі ці сервіси в одному транспортному домені, що дає змогу підвищити гнучкість, ефективність використання ресурсів і спростити управління мережею [4]. Архітектурно такі мережі, як правило, мають багаторівневу структуру [5], яка забезпечує розподіл функцій і оптимізацію маршрутизації — від рівня доступу, де розташовані кінцеві пристрої користувачів, через рівень розподілу (aggregation/distribution), де відбувається політизація і попередня обробка трафіку, до рівня ядра (core), що відповідає за магістральну маршрутизацію і взаємодію з іншими автономними системами [6]. Для наочного уявлення типової структури мережі зручно використовувати спрощену текстовий рисунок 1.1, який відображає взаємозв’язок рівнів і типові елементи інфраструктури [7]:

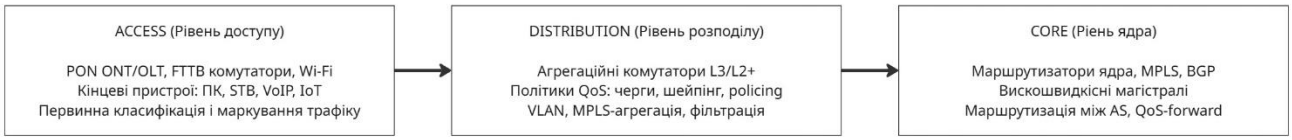


Рис. 1.1 Трирівнева модель мультисервісної телекомунікаційної мережі

Така трирівнева модель (Core–Distribution–Access) дозволяє ефективно застосовувати політики QoS [8]: на межі мережі відбувається класифікація і маркування трафіку, у розподілі реалізуються політики черг і формування, а в ядрі — швидке пересилання відповідно до DSCP/CoS позначень [9]. Аналіз трендів використання трафіку показує, що частка відеосервісів у загальному обсязі даних зростає швидше за інші категорії [10]; умовні дані, корисні для побудови ілюстративного графіка росту трафіку за видами сервісів, можуть мати такий вигляд (значення у відсотках від загального обсягу трафіку в мережі провайдера за роки) як показано в таблиці 1.1:

Таблиця 1.1

Порівняння обсягу різного трафіку в інтернетіа останні роки [10]

Рік	VoIP (%)	Відео (IPTV/OTT) (%)	Хмарні сервіси (%)	Інше (%)
2018	10	45	25	20
2020	8	56	26	10
2022	7	63	24	6
2024	6	71	19	4

Ці дані ілюструють, що відео стає домінуючим типом трафіку, а отже вимоги до QoS в мережах значною мірою визначаються характеристиками відеопередач [11]. Переваги мультисервісних мереж щодо традиційних технологій видно при порівнянні їхніх ключових характеристик: тоді як у TDM/ATM-середовищах гарантії QoS забезпечувалися на фізичному рівні через виділені канали [3], у IP-мережах QoS реалізується логічно — через

маркування, черги та політики [8]. Нижче наведено стислий порівняльний огляд традиційних мереж (TDM/ATM) і мултисервісних IP-мереж [3]:

Таблиця 1.2

Порівняння традиційних мереж (TDM/ATM) і мултисервісних IP-мереж [3]

Характеристика	Традиційні мережі (TDM/ATM)	Мультисервісні IP-мережі
Підхід до QoS	Фізичне виділення каналів	Логічне управління (маркування, черги)
Гнучкість	Низька	Висока
Масштабованість	Обмежена	Висока
Вартість	Висока при масштабуванні	Ефективна при конвергенції
Інтеграція сервісів	Складна	Проста

У практиці провайдерів застосовуються різні архітектури, які відображають зазначений принцип: наприклад, у широкому поширенні знаходяться мережі типу GPON/EPON на рівні доступу [7], далі — агрегаційні комутатори з функціями QoS та MPLS-ядро для побудови L3VPN/TE-транспортів [1]; у FTTB/FTTB+BRAS (BNG) архітектурах доступні механізми політизації на рівні BNG, які дозволяють персоналізувати QoS для абонентських сесій [2]; в операторському сегменті часто застосовують IP/MPLS-ядро з BGP у зоні взаємодії між операторами та MPLS TE для маршрутизації критичних потоків [4]. Конкретний приклад практичної архітектури провайдера виглядає так: абонентські прилади (VoIP телефони, IPTV STB, ПК) підключені до PON або Ethernet-комутаторів [7], які з’єднані з агрегаційними комутаторами; на агрегації виконується первинна класифікація та маркування [13], після чого трафік маршрутизується до мережевого ядра, де

політики DiffServ/CBWFQ/LLQ забезпечують пріоритетне обслуговування мультимедійних сервісів [8]. Оскільки відео утворює більшу частку трафіку, провайдери часто виділяють для нього окремі MPLS-піринги або CDN-точки присутності [1], щоб мінімізувати навантаження на основні магістралі.

Зі зростанням частки відео і хмарних сервісів у загальному обсязі мережевого трафіку постає необхідність детальніше розглянути вплив архітектури та політик управління на кінцеву якість сервісів [11]. У типових умовах навантаження, коли загальна пропускна здатність каналу обмежена, некерований фонова трафік (наприклад, великі FTP-передачі чи пікові оновлення) може витісняти чутливі сервіси, призводячи до підвищеної затримки, джиттера та втрат пакетів у VoIP і артефактів у IPTV [12]. Тому впровадження QoS у мультисервісних мережах не є вибором — це необхідна умова для забезпечення прийнятного QoE користувачів [8]. Що стосується практичних прикладів провайдерських рішень, то можна виділити такі характерні підходи:

1) застосування edge-функцій (BRAS/BNG, PON OLT з політиками) для персоналізації та маркування трафіку на межі мережі [2];
2) використання MPLS і TE-тунелів у магістралі для забезпечення гарантованих каналів для важливих сервісів [1];
3) впровадження CDN та локальних кешів для мінімізації міжміського відеотрафіку [14];
4) застосування систем моніторингу NetFlow/sFlow та реального часу (Prometheus, Grafana) для виявлення аномалій і регулювання політик [7].

У результаті, ідентифікація пріоритетних потоків, їх маркування та реалізація відповідних черг і політик на мережевих пристроях дозволяють знизити негативний вплив фонового навантаження та забезпечити стабільні параметри для сервісів реального часу [13].

1.2. Основні параметри QoS і їхній вплив на роботу сервісів

Основні параметри якості обслуговування у мультисервісних телекомунікаційних мережах визначають придатність мережевої інфраструктури для передавання різних типів трафіку та забезпечення стабільної роботи сервісів реального часу [1]. До ключових параметрів QoS відносять затримку (delay), варіацію затримки (jitter), втрати пакетів (packet loss), пропускну здатність (bandwidth), а також доступність (availability). Кожен із цих параметрів має свої нормативи, визначені, зокрема, рекомендаціями IETF та ITU-T [8] [11], і по-різному впливає на якість різних сервісів — VoIP, IPTV, відеоконференцій, інтерактивних додатків та хмарних сервісів [4] [5].

Затримка є сумарним часом, який проходить пакет від джерела до призначення [12]. У типових умовах загальна затримка складається з передавальної (transmission delay), обробної (processing delay), чергової (queuing delay) та поширення (propagation delay). Формально її можна подати як

$$D = D^{tr} + D^{pr} + D^{q} + D^{po} \tag{1.1}$$

де D^{tr} – затримка передачі;

D^{pr} – затримка обробки;

D^{q} – затримка в черзі;

D^{po} – затримка поширення [11].

Для VoIP критичним є не стільки абсолютне значення затримки, скільки її варіація, яка безпосередньо впливає на синхронність мовлення [4]. У більшості комунікаційних систем затримка понад 150 мс вже приводить до помітного зниження природності розмови, а при значеннях понад 300 мс – до переривання діалогу [11].

Варіація затримки (jitter) визначає нерівномірність інтервалів між послідовними RTP-пакетами [8]. Для IPTV та VoIP припустимі значення

зазвичай не повинні перевищувати 30–50 мс [12]. Високий джиттер призводить до накопичення пакетів у буферах та появи артефактів, таких як «розсинхронізація звуку», мікропаузи або затримки в голосових сервісах. Для IPTV висока варіація затримки призводить до пікселізації, стрибків зображення або втрати кадрів [4]. У практичних умовах джиттер компенсується буферами відтворення, однак їх збільшення веде до затримки запуску відео, що негативно позначається на QoE [12].

Втрати пакетів є одним з найбільш чутливих параметрів для сервісів реального часу [1]. Для голосу нормативною вважається втрата не більше 1 %, для відеосервісів — до 0,5 % [10]. При цьому навіть 0,1 % втрат у HD-відео може спричинити появу блокових артефактів. Внаслідок перевантаження черг, неправильного пріоритезування або низької якості каналів втрати зростають і безпосередньо впливають на якість відтворення. Типова ситуація — велике копіювання файлів (FTP/SMB) на каналі обмеженої пропускної здатності може витіснити RTP-трафік, якщо QoS не налаштовано, що призводить до втрати голосових пакетів і деградації розмови [2] [13].

Важливою частиною аналізу QoS є пропускна здатність [5]. Вона визначає максимальний обсяг даних, що може бути переданий за одиницю часу, і обмежує можливість одночасного передавання декількох типів трафіку. Наприклад, один VoIP-кодек G.711 потребує близько 87,2 кбіт/с RTP-трафіку з урахуванням накладних полів заголовків, тоді як потік IPTV у HD якості може потребувати від 4 до 12 Мбіт/с [4]. За наявності неконтрольованого фонового трафіку (наприклад, торрент-клієнтів або резервних копій) обмежена пропускна здатність каналу швидко заповнюється, і, якщо не застосовувати механізмів пріоритизації, критичні сервіси отримують значно гірші умови передавання [7].

Стандартизоване бачення вимог до різних класів трафіку уніфіковане у RFC 4594, у якому визначено класи обслуговування та їх типові характеристики. Узагальнені вимоги наведено нижче [10] .

Таблиця 1.3

Узагальнені вимоги до різного класу трафіку за вимогами RFC 4594 [10]

Клас трафіку	DSCP	Вимоги RFC 4594	Тип сервісу
Telephony (EF)	EF (46)	Дуже низька затримка, втрати <1 %, джиттер мінімальний	VoIP, SIP-RTP
Real-Time Interactive	CS5/AF41	Низька затримка та джиттер	Відеоконференції
Broadcast Video	CS3	Середня затримка, низькі втрати	IPTV
Multimedia Streaming	AF3x	Допустима затримка, стабільний бітрейт	OTT-відео
Mission-Critical Data	CS4	Низькі втрати	SCADA, керування
Best Effort	0	Відсутні гарантії	Веб-трафік, email
Background	CS1	Найнижчий пріоритет	Torrents, копії

Ця таблиця демонструє, що кожен тип трафіку потребує різних характеристик каналу [10]. Наприклад, голосовий трафік із класом EF має найвищий пріоритет, оскільки є надзвичайно чутливим до джиттера, тоді як фоновий трафік (CS1) може обслуговуватися у залишковому режимі [13], не погіршуючи роботу критичних сервісів.

Практичний вплив параметрів QoS на конкретні сервіси можна пояснити на прикладі VoIP та IPTV [4]. Голосовий трафік складається зі швидкої послідовності малих RTP-пакетів з інтервалом 20–30 мс, тому навіть невелика варіація затримки викликає спотворення звуку: ефект металевого відлуння, «розриви фраз», зависання звуку, спотворення тональної частоти [12]. Для IPTV, навпаки, критичними є втрати та стабільність бітрейту: при порушенні узгодженості кадрів з'являються блоки, зависання кадру, порушення

рухомості об’єктів. Якщо втрати перевищують 0,5 %, відео переходить у стан різких артефактів [10], що значною мірою знижує суб’єктивну оцінку якості — QoE.

Параметри QoS взаємопов’язані: збільшення навантаження призводить до збільшення чергової затримки, що впливає і на джиттер, і на втрати [1]. Тому якість обслуговування не може бути забезпечена без цілісної системи моніторингу, яка відстежує всі параметри в сукупності й дозволяє виявляти аномалії [7] як на канальному, так і на мережевому рівні. У мультисервісних мережах із високою концентрацією мультимедійного трафіку вимоги RFC 4594 фактично стають мінімальним стандартом для проектування політик QoS [10], забезпечуючи взаємну узгодженість технологій у різних сегментах мережі оператора.

1.3. Класифікація трафіку в мультисервісних мережах

Класифікація трафіку в мультисервісних телекомунікаційних мережах є фундаментальним елементом механізмів QoS, оскільки саме на етапі розпізнавання типу пакета та визначення його належності до певного класу приймається рішення про подальшу обробку — пріоритизацію, маркування, формування черг або обмеження смуги пропускання [1] [2] [13]. Усі сучасні системи QoS, включаючи DiffServ, MPLS EXP, CoS у VLAN та IP Precedence, базуються на тому, що трафік спочатку потрібно ідентифікувати [8] [9] [10]. Класифікація може здійснюватися на різних рівнях моделі OSI — від канального до прикладного — та використовувати як постійні ознаки (порт, протокол, DSCP), так і динамічні (глибокий аналіз вмісту або сигнатури додатків) [5] [6] .

Загальною схемою, що описує класифікацію трафіку на рівнях OSI, можна вважати таку структурну модель, яка охоплює рівні L2, L3 та L4. Основні критерії класифікації на цих рівнях наведено на рисунку 1.2 у вигляді узагальненої діаграми [1] :

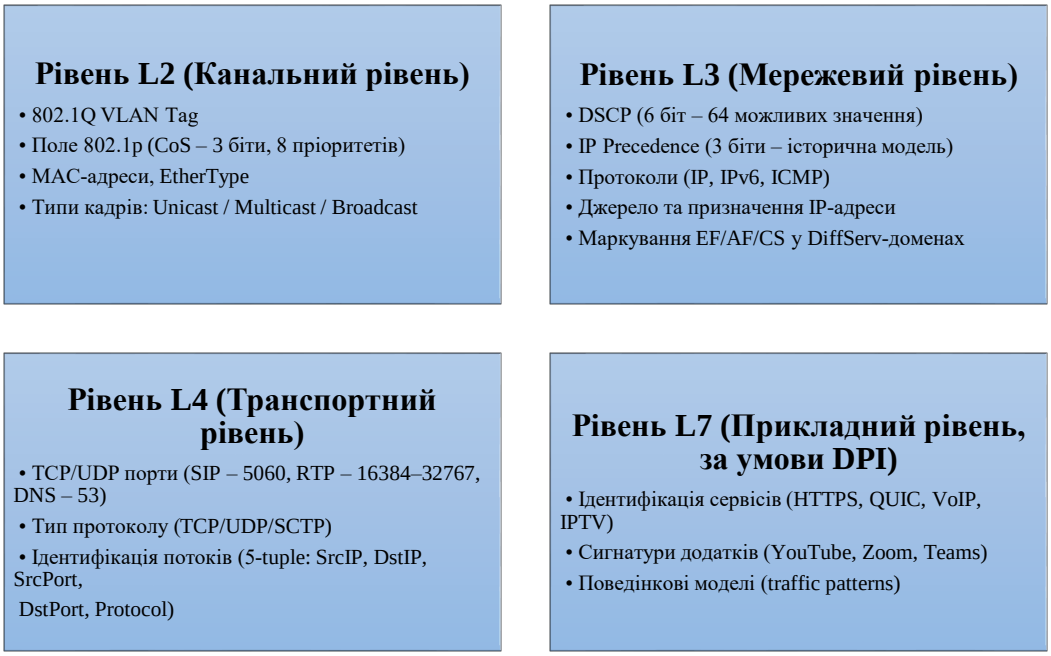


Рис. 1.2 Основні критерії класифікації на рівнях L2, L3, L4, L7

На рівні L2 найчастіше застосовують маркування CoS у полі 802.1p в рамках VLAN-тегу, де визначено 8 рівнів пріоритетів [1] [2]. Вони використовуються переважно у корпоративних мережах та у провайдерських сегментах доступу. Узагальнена таблиця CoS-пріоритетів наведена нижче:

Таблиця 1.4

Таблиця CoS-пріоритетів

CoS	Пріоритет	Тип трафіку
0	Best Effort	Стандартні дані
1	Background	Низький пріоритет
2	Spare	Допоміжний трафік
3	Excellent Effort	Важливі додатки
4	Controlled Load	Відео стандартної якості
5	Video	IPTV, HD-відео
6	Voice	VoIP
7	Network Control	Протоколи управління мережами

Для IP-мереж більш важливим є рівень L3, на якому використовується поле DSCP [8]. Це поле містить 6-бітний код, який визначає клас обробки пакета. Код DSCP дозволяє впроваджувати різні моделі QoS, включаючи EF (Expedited Forwarding), AF (Assured Forwarding) та CS (Class Selector). Узагальнена таблиця DSCP-кодів виглядає так:

Таблиця 1.5

Таблиця DSCP-кодів для IP-мереж

DSCP	Десяткове значення	Категорія	Опис
EF	46	Expedited Forwarding	Голосовий трафік, високий пріоритет
CS0	0	Best Effort	Трафік без гарантій
CS1	8	Background	Найнижчий пріоритет
AF11/12/13	10/12/14	Assured Forwarding 1	Дані низької пріоритетності
AF21/22/23	18/20/22	AF2	Стандартні додатки
AF31/32/33	26/28/30	AF3	Стримінг
AF41/42/43	34/36/38	AF4	Відео високої якості
CS5	40	Multimedia Conferencing	Відеоконференції
CS6	48	Network Control	Управління
CS7	56	Critical Control	Критичні протоколи

Рівень L4 дозволяє класифікувати трафік за портами TCP/UDP [5] [6]. Наприклад, для VoIP використовують SIP (порт 5060), RTP (динамічні порти 16384–32767), для IPTV часто застосовують UDP-мультикаст порти, а для DNS — порт 53. Однак класифікація за портами перестає бути надійним

механізмом через широке використання шифрування та тунелювання трафіку [7].

Сучасні мережі дедалі частіше використовують класифікацію на основі DPI (Deep Packet Inspection) [7]. DPI аналізує глибший рівень протоколів, включно з прикладним. Це дозволяє визначати реальний тип трафіку навіть при нестандартних портах чи обфускації. Проте DPI має суттєві обмеження:

1) значне навантаження на процесорні ресурси;
2) можливість колізій при обробці шифрованого трафіку (HTTPS/QUIC);
3) складність масштабування у великих ISP;
4) ризики з боку конфіденційності та регуляторні обмеження [6].

Практичне використання класифікації найчастіше реалізується через конфігурацію мережевого обладнання [1] [13]. Такі конфігурації дозволяють визначати тип трафіку навіть при використанні складних політик. У мультисервісних мережах методи класифікації поєднуються: спочатку трафік маркується на межі мережі (BRAS, BNG, OLT), потім у магістралі маршрутизатори опираються на DSCP-позначення, у той час як додаткові DPI-системи можуть застосовуватися для уточнення типу сервісу у випадку некоректного або злоякісного трафіку [2] [7] .

Класифікація є основою для наступних етапів управління QoS — пріоритизації, формування черг, шейпінгу та полірування [1] [13]. Без правильного розпізнавання трафіку неможливо забезпечити коректну роботу сервісів реального часу, оскільки будь-який некласифікований або помилково визначений трафік отримує рівень «best effort» і може стати причиною перевантаження черг та деградації сервісу [10] [5]. Розгалужений інструментарій класифікації дає змогу операторам забезпечити необхідний рівень якості сервісів, а у складних мережах — адаптувати політики QoS до змінних умов та типів навантаження [2] [15] . Таким чином, класифікація трафіку є ключовим елементом побудови ефективної мультисервісної мережі, яка забезпечує високу якість обслуговування для найчутливіших до затримки та втрат потоків [1] [5] [15].

1.4. Механізми управління та пріоритизації мережевого трафіку

Механізми обробки трафіку у мультисервісних телекомунікаційних мережах відіграють ключову роль у забезпеченні якості обслуговування, оскільки саме вони визначають, які пакети будуть передані першими, як довго вони можуть чекати, чи будуть відкинуті, а також яким чином пропускна здатність каналу буде розподілена між різними класами трафіку [1] [2] [13]. Для оптимальної роботи сервісів реального часу недостатньо просто класифікувати трафік — необхідно також застосувати черги, пріоритизацію, механізми контролю навантаження та алгоритми попереднього відкидання пакетів, що дозволяють уникати перевантаження та забезпечувати передбачуваність параметрів QoS [5] [6].

Найпростішою моделлю обробки трафіку є FIFO (First In — First Out), в якій пакети надходять у чергу та обробляються в тому порядку, в якому вони були отримані [5]. Цей механізм не виконує жодної класифікації або пріоритизації, тому у випадку змішаного трафіку він призводить до «голодування» сервісів реального часу через домінування великого обсягу некритичного трафіку. В умовах переповнення черги FIFO просто відкидає надлишкові пакети, причому рівень втрат може досягати десятків відсотків, що робить FIFO абсолютно непридатним для VoIP або IPTV [4] [1].

На відміну від FIFO, алгоритм WFQ (Weighted Fair Queuing) дозволяє забезпечити розподіл смуги пропускання між класами трафіку відповідно до їхніх вагових коефіцієнтів[1] [2] . Це означає, що кожний клас отримує гарантовану частину ресурсу каналу. У спрощеному вигляді пропускну здатність класу можна подати формулою:

$$B_i = \left(\frac{W_i}{\sum W}\right) \times B \tag{1.2}$$

де B — загальна пропускна здатність інтерфейсу;

W_i — вага класу;

$\sum W$ — сума всіх ваг [5].

Таким чином, WFQ дозволяє розподіляти канальну пропускну здатність пропорційно визначеним значенням, забезпечуючи певний мінімальний рівень сервісу для кожного типу трафіку. Проте WFQ не гарантує «абсолютного» пріоритету, що є важливим для RTP-трафіку [4] .

Вдосконалений алгоритм CBWFQ (Class-Based WFQ) вводить поняття класів із точним визначенням їхньої пропускної здатності [1] [13]. На відміну від WFQ, де ваги визначаються автоматично, CBWFQ дозволяє адміністратору явно вказати, яку частину смуги пропускання резервувати для кожного класу. Це забезпечує можливість виділити стабільний канал для відеосервісів, голосу, службового трафіку або інтерактивних додатків [2]. Однак CBWFQ, так само як і WFQ, не дає можливості створити абсолютний пріоритет [13].

Для цього використовується алгоритм LLQ (Low Latency Queuing), який додає до CBWFQ спеціальну пріоритетну чергу для RTP-трафіку [1] [13]. Коли в системі присутні пакети, що належать до класу EF (Expedited Forwarding), вони обробляються негайно, минаючи черги іншого трафіку [8] [10]. Це дозволяє досягти мінімальної затримки та джиттера для голосових або інших надчутливих потоків. Проте надмірне використання LLQ може призвести до "голодування" інших класів; щоб цього уникнути, пріоритетна черга має бути обмежена поліцером, який контролює її максимальний розмір [13].

У реальних мережах перевантаження можуть виникати не лише через надмірний обсяг трафіку, а й через асиметрію каналів або обмеження апаратного рівня [7] [15]. Наприклад, ситуація, коли абонент виконує великі завантаження через інтерфейс зі швидкістю 100 Мбіт/с, а аплінк вузла обмежений до 1 Гбіт/с, може спричинити накопичення черг на агрегаційному комутаторі та періодичні сплески втрат у VoIP. Інший приклад — IPTV-мультикаст на перевантаженому інтерфейсі, де велика кількість одночасних HD-каналів заповнює чергу, призводячи до пікселізації постійних потоків [4].

Для запобігання перевантаженням використовуються алгоритми RED (Random Early Detection) та WRED (Weighted Random Early Detection) [1] [13]. RED намагається запобігати переповненню черги, починаючи відкидати

пакети з невеликою ймовірністю ще до того, як черга буде заповнена. WRED є розширеною версією RED і враховує DSCP-позначення [9] [10]. Це дозволяє відкидати пакети низьких пріоритетів раніше, ніж важливі потоки. Порівняльна таблиця RED і WRED наведена нижче:

Таблиця 1.6

Порівняння характеристик RED і WRED

Характеристика	RED	WRED
Врахування пріоритетів	Ні	Так (DSCP)
Поведінка при перевантаженні	Рівномірне відкидання	Різні пороги для різних класів
Придатність для QoS	Обмежена	Висока
Типові сценарії	Некритичні мережі	Мережі з пріоритетами (VoIP/IPTV)

Ще одним важливим елементом управління трафіком є шейпінг та поліцинг, що використовують алгоритми Token Bucket та Leaky Bucket [5] [6]. Token Bucket працює таким чином, що дозволяє періодичні сплески трафіку, накопичуючи «токени», тоді як Leaky Bucket здійснює жорстке вирівнювання швидкості, змушуючи трафік відповідати заданому бітрейту. Формально Token Bucket визначається параметрами CIR (Committed Information Rate), CBS (Committed Burst Size) та EBS (Excess Burst Size), а швидкість пропускання R описується як:

$$R = min(CIR, TokenRate) \quad (1.3)$$

Token Bucket є більш гнучким, оскільки дозволяє короткочасні підвищення швидкості, тоді як Leaky Bucket повністю усуває «сплески», перетворюючи трафік на рівномірний потік [6]. Ці механізми широко використовуються у політиках QoS для контролю трафіку перед його передачею у мережу провайдера або при обмеженні потоків IPTV та VoIP [7].

У сукупності механізми FIFO, WFQ, CBWFQ, LLQ, RED/WRED та Token/Leaky Bucket формують комплексну систему, що визначає, яким чином трафік рухається у мережі у випадку перевантаження [1] [2] [13]. Правильна конфігурація цих механізмів забезпечує відсутність значних черг для сервісів реального часу, стабільне передавання відео, мінімізацію втрат пакетів та оптимальний розподіл ресурсів [4] [5]. У мультисервісних мережах, де одночасно присутні VoIP, IPTV, хмарні сервіси, онлайн-ігри та фоновий трафік, застосування таких механізмів є критично необхідним для підтримання гарантованих параметрів якості обслуговування та забезпечення прийнятного рівня QoE [1] [15].

1.5. Моделі забезпечення QoS у сучасних телекомунікаційних системах

У сучасних телекомунікаційних мережах застосовують декілька моделей забезпечення якості обслуговування, що різняться між собою механізмами, масштабованістю, складністю реалізації та ефективністю у різних типах інфраструктур [5] [6] . Еволюція підходів до QoS пов’язана з розвитком IP-мереж, які з моменту свого створення працювали на модель «best effort» — тобто намагалися доставити пакети, але не надавали жодних гарантій щодо затримки, пропускної здатності або втрат [5]. Із появою мультимедійних сервісів та збільшенням навантаження виникла необхідність у більш гнучких і надійних системах, що забезпечують контроль параметрів якості обслуговування [1]. Історично першою моделлю стала IntServ, яка забезпечувала гарантовану якість обслуговування шляхом резервування ресурсів через протокол RSVP [8]. Згодом, через проблеми масштабованості, з’явилася модель DiffServ, що базується на класифікації та пріоритизації трафіку без необхідності встановлення стану з’єднання на кожному маршрутизаторі [9].

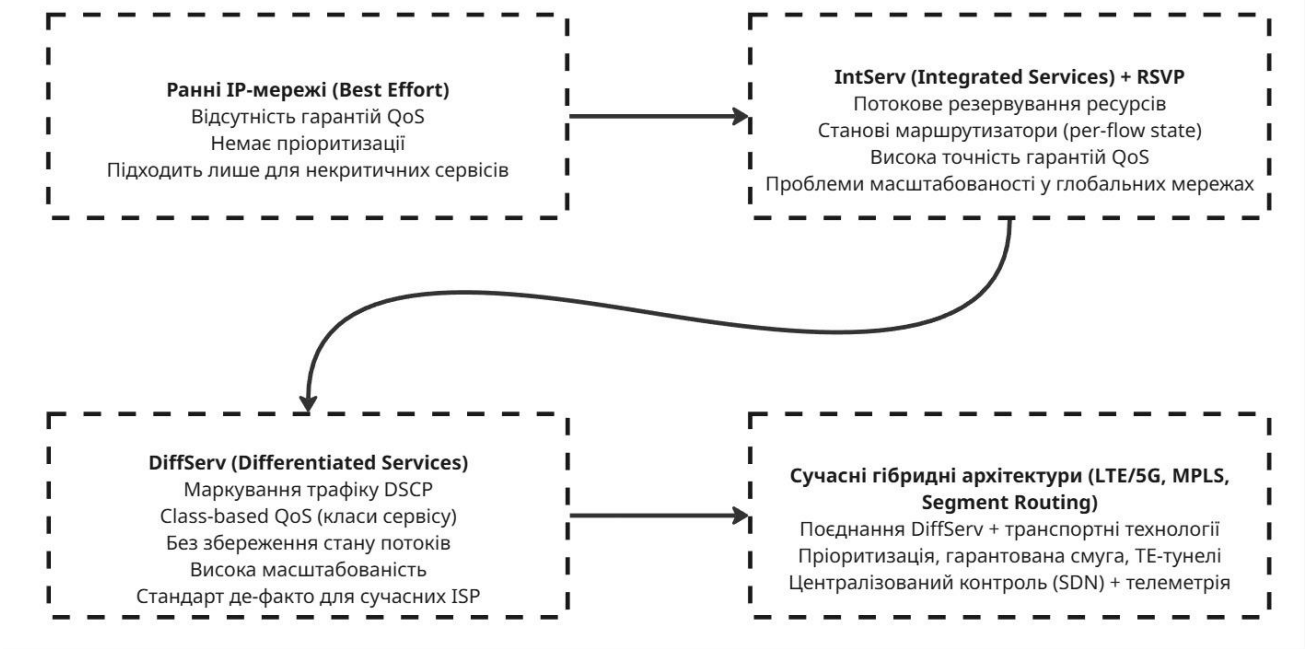


Рис 1.3 Еволюцію моделей QoS

Модель IntServ базується на тому, що кожний потік (flow) резервує певні ресурси вздовж усього маршруту завдяки протоколу RSVP [8]. Під час встановлення сесії відправник надсилає повідомлення PATH, а приймач — RESV, у якому вказує вимоги до пропускної здатності. Маршрутизатори вздовж траси повинні створювати стан з'єднання, зберігати таблиці потоків та підтримувати виділені ресурси для кожного з них [5]. У невеликих мережах або в інфраструктурах, де кількість потоків обмежена, такий підхід працює добре, але у глобальних мережах має серйозні обмеження:

1. необхідність зберігати стан для кожної сесії значно навантажує пам'ять маршрутизаторів;
2. велика кількість потоків потребує значних обчислювальних ресурсів (CPU), оскільки на кожний RSVP-пакет здійснюється складна обробка;
3. RSVP не масштабується у мережах з мільйонами користувачів;
4. кожна зміна маршруту вимагає повторного резервування ресурсів. Тому IntServ практично не використовується у великих ISP-мережах, хоча може зустрічатися у мережах спеціального призначення або у транспортних магістралях високої надійності [8] [5].

Модель DiffServ, навпаки, створена спеціально для глобальних мереж і не потребує збереження стану потоків [8] [9]. Вона працює за принципом маркування трафіку DSCP-кодами у заголовку IP-пакета, після чого всередині провайдерської мережі пакети проходять обробку відповідно до класів обслуговування [10] [13]. Маркування, як правило, здійснюється на межі мережі (BNG, BRAS, CPE, OLT), а далі маршрутизатори магістралі виконують лише швидку обробку відповідно до призначеного DSCP [1] [2]. Це дозволяє застосовувати DiffServ у мережах будь-якого розміру, включно з магістральними інтернет-провайдерами.

Типовий приклад призначення DSCP-класів у DiffServ може виглядати таким чином: для VoIP використовується EF (46), який має найвищий пріоритет; для відеоконференцій — CS5; для IPTV — CS3 або AF41; для стримінгових сервісів — AF3x; для звичайного веб-трафіку — BE (0); для фонового трафіку — CS1 [10] [13]. Це дозволяє маршрутизаторам приймати швидкі рішення без аналізу вмісту пакета. На різних інтерфейсах виконується мапування DSCP → вихідні черги (queue), що забезпечує практичну реалізацію QoS [1] [2] .

Процес обробки пакета у DiffServ-мережі можна подати у вигляді узагальненої схеми:

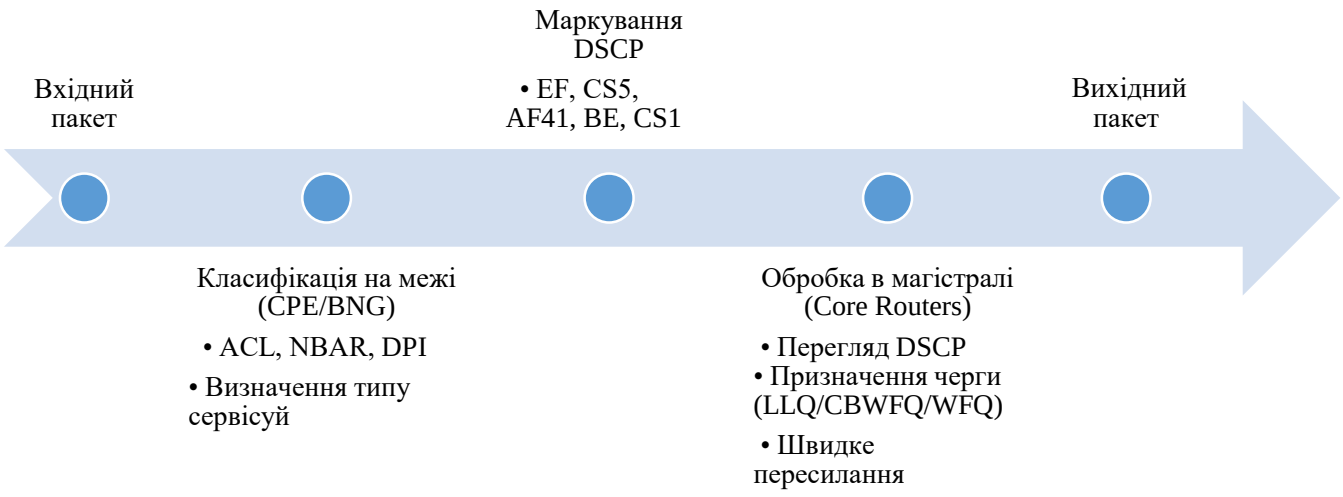


Рис. 1.4 Процес обробки пакета у DiffServ-мережі

Завдяки цьому DiffServ демонструє високу продуктивність навіть під великим навантаженням. Основна його перевага — масштабованість, оскільки маршрутизатори не зберігають стан потоків, а працюють лише з DSCP-кодами [9] [10] [13].

Історично розвиток QoS у IETF відбувався поступово. У 1990-х роках робоча група IntServ запропонувала RSVP для забезпечення гарантованої якості обслуговування у мережах [8]. Однак з експоненційним ростом інтернету стало зрозуміло, що модель потокового резервування не може обслуговувати мільйони користувачів [5]. У кінці 1990-х IETF розробила DiffServ (RFC 2474 і RFC 2475), що стало фундаментом сучасних QoS-систем. Далі було прийнято RFC 2597 (Assured Forwarding) та RFC 3246 (Expedited Forwarding), які описують класи трафіку, а згодом RFC 4594, що стандартизував профілі сервісів [5]. Разом ці документи забезпечили міжнародну уніфікацію QoS у IP-мережах.

У реальних мережах оператори зазвичай поєднують декілька підходів. Наприклад, у мобільних мережах LTE/5G елементи гарантування QoS частково нагадують IntServ (такі як GBR — Guaranteed Bit Rate), але в основі системи Core все одно лежить DiffServ [1] [2]. У корпоративних мережах застосовуються політики CoS/DSCP, які забезпечують послідовність обробки [13]. Таким чином, DiffServ фактично став стандартом де-факто для глобальних і мультисервісних мереж, забезпечивши оптимальний баланс між простотою, керованістю та масштабованістю [15].

Модель забезпечення QoS у сучасних провайдерських та корпоративних мережах неможливо розглядати окремо від централізованих систем моніторингу та інструментів аналізу трафіку [7] [1]. Маркування трафіку, пріоритизація, застосування черг і планувальників, а також контроль параметрів реального часу — усе це компоненти єдиної QoS-екосистеми, яка дозволяє забезпечити стабільну роботу сервісів у середовищі з різнорідним навантаженням [15].

Висновки до розділу 1

У першому розділі було проведено всебічний аналіз теоретичних засад функціонування мультисервісних телекомунікаційних мереж та принципів забезпечення якості сервісу (QoS). Розглянуто сутність мультисервісності, особливості передавання різнорідних типів трафіку та фактори, що впливають на якість його доставки. На основі аналізу встановлено, що сучасні мережі потребують механізмів гарантованого обслуговування, оскільки традиційна модель best-effort не забезпечує стабільної роботи чутливих до затримок сервісів.

У розділі досліджено концепції управління якістю обслуговування, зокрема моделі DiffServ, класифікацію трафіку, механізми пріоритезації, формування черг, політики виділення смуги пропускання та їх вплив на ключові параметри QoS — затримку, джитер та втрати пакетів. Показано, що застосування політик QoS є невід’ємним елементом побудови ефективних корпоративних мереж, які містять трафік реального часу, бізнес-трафік та некритичні дані.

Проведений огляд технологій Cisco та MikroTik підтвердив їхню здатність реалізувати сучасні механізми QoS на основі маркування, класифікації та управління чергами. Теоретичні відомості, отримані у цьому розділі, створюють фундамент для подальшого проектування мережевої інфраструктури, визначення параметрів якості та розроблення практичних політик QoS, що будуть реалізовані у наступних розділах.

РОЗДІЛ 2. МЕТОДИ ТА ЗАСОБИ МОНІТОРИНГУ ЯКОСТІ ОБСЛУГОВУВАННЯ

2.1. Загальні принципи моніторингу параметрів QoS

Моніторинг параметрів якості обслуговування в мультисервісних телекомунікаційних мережах є ключовою складовою їхньої стабільної роботи, оскільки дає змогу своєчасно виявляти відхилення у характеристиках трафіку, діагностувати перевантаження та аварії, оптимізувати політики QoS і прогнозувати майбутнє навантаження [5]. На відміну від звичайного моніторингу доступності або пропускної здатності, моніторинг QoS потребує детального вимірювання динамічних параметрів, таких як затримка, варіація затримки, втрати пакетів, тривалість черг, стабільність бітрейту та коректність маркування DSCP/CoS [13]. У мультисервісному середовищі, де одночасно передаються VoIP, IPTV, службові дані та фоновий трафік, контроль цих параметрів дозволяє забезпечити передбачуваність та узгодженість роботи мережевих сервісів [4].

Моніторинг QoS, як правило, здійснюється на декількох рівнях моделі OSI. На канальному рівні вимірюються параметри завантаження портів комутаторів, кількість помилок, колізій, фреймів із CRC-помилками, повторних передавань та інших показників, що впливають на якість передачі. На мережевому рівні контролюються маршрути, затримки та втрати на проміжних вузлах, а також коректність роботи протоколів маршрутизації [7]. На транспортному та прикладному рівнях аналізуються характеристики окремих потоків, їхній бітрейт, стабільність, параметри RTP-сесій, а також поведінка конкретних сервісів, таких як VoIP або потокове відео [11].

Вимірювання параметрів затримки та джиттера є особливо важливим елементом моніторингу QoS. У практиці операторів широко використовуються такі методи як ICMP-based вимірювання затримок, UDP-тести, симулювання RTP-сесій або спеціальні агенти для перевірки якості [2].

Загальна формула середньої затримки між двома вузлами у мережі може бути представлена як:

$$D_{\text{мін}} = \frac{\sum(D_i)}{N} \tag{2.1}$$

де N — кількість вимірювань;

D_i — затримка одного пакета.

Для моніторингу джиттера використовується формула згідно з RFC 3550, яка обчислює середню варіацію інтервалів між надходженням RTP-пакетів [8]. Відповідно до цього стандарту джиттер оцінюється як експоненційно згладжене значення:

$$J = J + \frac{(|D_i - D_{i-1}| - J)}{16} \tag{2.2}$$

де J — поточне значення варіації затримки;

D_i — час надходження чергового пакета.

Це дає змогу оперативно відслідковувати нестабільність каналів передачі.

Контроль втрат пакетів здійснюється як шляхом опитування пристроїв на наявність dropped/errored пакетів, так і за допомогою активних тестів, які вимірюють частку втрачених пробних пакетів [7]. Особливо важливо відстежувати втрати для RTP-потоків та IPTV-мультикастів, оскільки навіть незначні відхилення тягнуть за собою помітні артефакти [4] [12] .

Моніторинг QoS також включає збір статистики з мережевого обладнання за допомогою таких протоколів і технологій, як SNMP, NetFlow, sFlow та IPFIX. SNMP дозволяє збирати статистику портів, черг, CPU і пам’яті пристроїв, що є критично важливим для виявлення перевантаження [7]. NetFlow та sFlow забезпечують детальний аналіз потоків, їхнього бітрейту, DSCP-позначень та структури трафіку. Наприклад, NetFlow може зафіксувати, що 30 % трафіку передається з маркуванням EF, що може сигналізувати про надмірну кількість пріоритетного трафіку, здатного порушити баланс між класами. Крім того, ці технології дозволяють виявляти аномальні ситуації, такі

як збільшення кількості TCP retransmissions, стрибки у multicast-групах, або підозрілу активність.

У системах моніторингу QoS також важливим є контроль черг на маршрутизаторах і комутаторах. Оскільки більшість QoS-політик базується на призначенні черг для кожного класу трафіку, систематичне накопичення в одній із черг свідчить про те, що механізми шейпінгу або пріоритизації налаштовані неправильно [1] [13]. Наприклад, якщо черга класу AF41 для відео постійно близька до максимального розміру, це може означати, що відеотрафік перевищує виділену йому смугу або що трафік неправильно класифікується. Аналогічно, постійне переповнення черги BE може свідчити про загальне перевантаження мережі або недостатню загальну пропускну здатність інтерфейсу [5].

Окремим елементом моніторингу QoS є контроль коректності маркування трафіку. У провайдерських мережах маркування часто виконується на рівні BNG/BRAS або CPE-обладнання, після чого коректність DSCP-кодів повинна перевірятися у магістралі [1]. Якщо трафік, що має бути EF або CS5, отримує значення BE або CS0, це призведе до його некоректної обробки. У великих провайдерських мережах для цього застосовують NetFlow з аналізом DSCP-розподілу [7] .

Моніторинг QoS повинен бути безперервним і багаторівневим. Сучасні системи моніторингу дозволяють збирати дані в реальному часі, зберігати історію, будувати графіки та здійснювати аналіз трендів [7]. Це дає змогу виявляти не лише аварії, а й довготривалі зміни: наприклад, поступове зростання джиттера у вечірні години може сигналізувати про те, що канал перевантажується лише у пікові періоди, а не протягом усього дня [5]. Такий тип аналізу дозволяє оптимізувати ресурси, планувати модернізацію та коригувати політики пріоритизації[2] .

Важливою складовою моніторингу QoS є також синтетичні тести — активні вимірювання, які запускаються незалежними агентами та моделюють поведінку реального трафіку [4] [11]. Для VoIP використовуються симулятори

RTP-навантаження, які генерують голосові пакети з інтервалами 20–30 мс, що дозволяє перевіряти систему на чутливість до джиттера й затримки. Для IPTV застосовуються UDP-based тести з фіксованим бітрейтом, які перевіряють стабільність передавання без втрат та стрибків [1] [2].

У сукупності моніторинг QoS є комплексним процесом, що охоплює збір статистики, аналіз параметрів трафіку, контроль черг, перевірку коректності маркування, активні тести та інструменти візуалізації [15]. Його головне завдання — забезпечити безперервну інформаційну картину стану мережі та дозволити операторам вчасно реагувати на зміни, підтримуючи високий рівень QoE користувачів. У сучасних мультисервісних мережах моніторинг QoS став не лише технічною необхідністю, але й ключовим компонентом управління мережею загалом [5] [15].

2.2. Інструменти збору мережевої статистики: SNMP, NetFlow, sFlow

Збір мережевої статистики є базовим елементом моніторингу QoS, оскільки дозволяє отримувати інформацію про роботу інтерфейсів, черг, протоколів, потоків та сервісів у режимі реального часу. У телекомунікаційних мережах найпоширеними інструментами збору статистичних даних є SNMP, NetFlow, sFlow та IPFIX [6]. Ці технології забезпечують різні підходи до моніторингу: опитування пристроїв, відправлення телеметрії, збір потоків і статистики трафіку, аналіз завантаження каналів та стану обладнання [2] [7]. Їх комплексне застосування дозволяє операторам отримати повну картину роботи мережі та забезпечити стабільність показників QoS[5] [15] .

SNMP (Simple Network Management Protocol) є класичним протоколом збору інформації про стан мережевих пристроїв [7]. Він базується на періодичному опитуванні обладнання менеджером моніторингу та наданні відповідей у вигляді значень MIB-об’єктів. SNMP дозволяє отримувати інформацію про завантаження інтерфейсів, статистику вхідних і вихідних пакетів, кількість помилок, значення черг, завантаження процесора, стан

маршрутизації та інші критичні параметри [7] [1]. Наприклад, для контролю QoS часто опитуються лічильники черг, такі як queue length, dropped packets та queue discards. Якщо значення скинутих пакетів у черзі зростають, це безпосередньо вказує на перевантаження інтерфейсу або некоректну політику QoS [1] [2]. SNMP підтримує механізм trap-повідомлень, за допомогою яких пристрій може самостійно сигналізувати про перевищення порогів затримки, помилки або переповнення буферів, що робить його ефективним інструментом раннього попередження [7].

NetFlow є одним із найпотужніших засобів аналізу трафіку, оскільки він не просто вимірює обсяг чи кількість пакетів, а описує кожен потік даних у мережі [1] [2]. Потік у NetFlow визначається як сукупність пакетів з однаковим набором атрибутів: адреса джерела та призначення, порт джерела та призначення, протокол, DSCP-код та інші параметри [13]. Це дозволяє здійснювати глибинний аналіз поведінки трафіку, визначати, які застосунки генерують найбільше навантаження, які користувачі або сегменти мережі найбільш активні, та які типи трафіку знижують якість обслуговування. У контексті QoS особливо цінною є можливість аналізувати розподіл трафіку за DSCP-мітками. Якщо, наприклад, у мережі з’являється надмірна кількість потоків з DSCP EF, це може призвести до перевантаження пріоритетної черги [13]. Крім того, NetFlow дозволяє фіксувати аномалії, як-от надмірна кількість коротких TCP-з’єднань, відхилення у середньому розмірі пакета або різкі стрибки трафіку, що може свідчити про атаки чи порушення QoS [2].

sFlow працює за принципом статистичного вибіркового аналізу, що дозволяє збирати дані про трафік із мінімальним навантаженням на обладнання [7]. Завдяки використанню механізму sampling (наприклад, 1:100 або 1:1000) sFlow отримує репрезентативну вибірку пакетів, яка дає змогу оцінити загальну структуру трафіку, його розподіл за протоколами, потоками та класами [6]. На відміну від NetFlow, який працює з потоками повністю, sFlow передає у систему моніторингу інформацію про окремі пакети та статистичні лічильники інтерфейсів [7]. Це робить його оптимальним для

високошвидкісних магістралей 40G/100G, де повноцінний NetFlow створював би надмірне навантаження. Для QoS sFlow дозволяє оцінити співвідношення пріоритетних і непріоритетних пакетів, відстежити мультикаст-потоки IPTV, а також аналізувати MAC-рівень, що є важливим у комутованих середовищах [1].

Таблиця 2.1

Порівняння різних інструментів збору статистики

Характеристика	SNMP	NetFlow	sFlow
Тип збору даних	Опитування + trap	Потоковий аналіз	Вибірковий аналіз (sampling)
Рівень OSI	L1–L7 (через MIB)	L3–L4	L2–L7
Навантаження на обладнання	Низьке	Середнє–високе	Дуже низьке
Аналіз DSCP/CoS	Частково	Повний	Повний
Підтримка IPTV	Обмежена	Висока	Висока
Детальність	Низька	Дуже висока	Середня
Придатність для 40G/100G	Висока	Середня	Висока

Комплексне використання SNMP, NetFlow та sFlow дає змогу створити багаторівневу систему моніторингу, що охоплює як загальний стан інтерфейсів, так і детальну структуру трафіку [1] [2] [7]. Наприклад, у типовій провайдерській мережі SNMP використовується для моніторингу стану маршрутизаторів і комутаторів, NetFlow — для аналізу поведінки трафіку та визначення найбільш навантажених сервісів, а sFlow — для контролю високошвидкісних магістральних каналів [6]. У результаті оператор отримує повноцінну картину роботи мережі: від фізичного стану портів до аналізу кожного окремого потоку.

У сучасних мультисервісних мережах ці інструменти також інтегруються з системами колекціонування та аналізу, такими як Prometheus, Grafana, Zabbix, ELK та InfluxDB [1] [7]. Наприклад, SNMP активно застосовується у Zabbix для побудови графіків завантаження інтерфейсів [2], тоді як NetFlow у поєднанні з Elasticsearch забезпечує глибинний аналіз потоків у реальному часі. Зі свого боку, sFlow легко інтегрується з CollectD та Telegraf, що дозволяє отримувати узагальнені показники про роботу магістральних лінків [6].

Таким чином, SNMP, NetFlow та sFlow доповнюють один одного і забезпечують комплексний підхід до збору статистики. Використання цих технологій є необхідною умовою для побудови ефективної системи моніторингу QoS, яка дозволяє забезпечити високий рівень контролю та керованості мережі, швидке виявлення аномалій і точне налаштування політик пріоритизації трафіку [1] [7].

2.3. Аналіз мережевого трафіку засобами Wireshark

Аналіз мережевого трафіку є невід’ємною складовою моніторингу QoS [5] [6], оскільки дозволяє досліджувати поведінку мережі на рівні окремих пакетів, потоків і протоколів. Одним із найпотужніших інструментів для цього є Wireshark [5] [6], який забезпечує глибоку інспекцію трафіку, декодування протоколів різних рівнів OSI, фільтрацію пакетів за десятками параметрів та детальне відображення кожного сегмента переданих даних. За допомогою Wireshark [5] [6] можна визначити, як саме працює мережа в конкретний момент часу, які протоколи домінують, які класи трафіку генеруються, як поводиться VoIP або IPTV, чи є проблеми з QoS-політиками [1], чи відбувається перевантаження каналів, чи є помилки на канальному рівні або повторні передавання TCP.

Wireshark працює за принципом перехоплення пакетів на мережевому інтерфейсі та збереження їх у вигляді дампу (файлу .pcap), що дозволяє

проводити як онлайн-аналіз, так і офлайн-дослідження архівів трафіку [5]. На відміну від SNMP чи NetFlow, які дають агреговану інформацію, Wireshark забезпечує аналіз на рівні кожного окремого пакета [1] [7]. Завдяки декодуванню сотень протоколів, включаючи Ethernet, VLAN 802.1Q, IP, TCP, UDP, RTP, SIP, RTCP, IGMP, DHCP, OSPF, BGP та ін., Wireshark є ключовим інструментом для діагностики QoS і мережевих проблем [5] [6].

Одним із найважливіших аспектів аналізу трафіку у Wireshark є можливість перевіряти коректність класифікації та маркування трафіку [9] [10]. Оскільки механізми QoS значною мірою залежать від DSCP-міток, тегів CoS та VLAN-пріоритетів, Wireshark дозволяє безпосередньо бачити, які саме мітки встановлені у заголовках пакетів. Наприклад, для RTP-потоків VoIP має бути встановлено DSCP EF (46) [1] [4], а для відеотрафіку часто використовується AF41 або CS4 [9] [10]. Якщо при аналізі трафіку виявляється, що частина пакетів позбавлена correct DSCP-позначень або неправильно маркована, це свідчить про помилку у QoS-полісі на маршрутизаторі або комутаторі [1] [13].

Wireshark дозволяє досліджувати параметри, важливі для QoS, такі як затримка та джиттер [11] [12] . Для RTP-потоків використовується функція RTP Stream Analysis, яка обчислює втрати пакетів, варіацію затримки, MOS-оцінку, середній інтервал між пакетами та тестує стабільність бітрейту [4]. Ці дані є надзвичайно важливими для VoIP, оскільки навіть невеликий джиттер може призвести до розривів у голосі. Наприклад, якщо Wireshark фіксує, що джиттер перевищує 30 мс, а втрати пакетів сягають 3%, це свідчить про те, що політика LLQ недостатньо ефективна, або що канал перевантажений [1] [13]. Аналогічно, для IPTV Wireshark дозволяє аналізувати мультикаст-трафік IGMP та RTP/UDP, визначати стрибки бітрейту, пропуски фреймів та періоди нестабільності [4] [15].

У середовищах GNS3 та реальних мережах часто застосовуються додаткові режими перехоплення, такі як SPAN-порти, TAP-пристрої або захоплення трафіку безпосередньо на маршрутизаторах [14], що дає змогу

отримувати чисту копію потоків, не впливаючи на проходження даних. Наприклад, для аналізу IPTV-трафіку створюється SPAN-сесія на комутаторі, яка дублює мультикаст-потоки на окремий порт, підключений до ПК з Wireshark [7]. Це дозволяє детально вивчати структуру MPEG-TS-фреймів, RTP-пакування та стабільність UDP-потоку [4] .

У рамках моніторингу QoS Wireshark особливо корисний для аналізу сигналізації VoIP. Протокол SIP, який контролює встановлення та завершення дзвінків, може показувати затримки при встановленні сеансу, неправильну маршрутизацію SIP-повідомлень, некоректну роботу NAT або SIP ALG [4]. Наприклад, при детальному аналізі SIP Invite/OK/ACK-послідовностей можна визначити, де саме виникає затримка при встановленні виклику, чи надходять відповіді своєчасно, чи не виникає повторних передавань. Аналіз SIP-транзакцій дозволяє виявити проблеми, які на перший погляд не пов'язані з QoS, але можуть суттєво впливати на користувацький досвід [1].

Не менш важливою є можливість визначати проблеми на канальному рівні. Wireshark здатен показувати дублікати фреймів, помилки контрольних сум, надмірні повторні передавання TCP, неправильні значення VLAN-tag, а також порушення MTU [5] [7], які призводять до фрагментації. Проблеми з MTU можуть проявлятися у вигляді зниження якості відеопотоків або затримок у роботі VoIP [4] [15]. Наприклад, при неправильному налаштуванні Jumbo Frames IPTV може отримувати фрагментовані UDP-пакети, що веде до артефактів і просідання якості.

Wireshark також забезпечує розширені інструменти статистики, такі як Protocol Hierarchy, Conversations і Endpoints, які дають змогу оцінити загальну структуру трафіку та визначити найактивніші вузли. У контексті QoS ці інструменти дозволяють визначати, чи не займають фонові процеси надто багато смуги, чи не відбувається небажаних передач великих файлів у піковий час, чи не домінує певний тип протоколу над іншими [1] [2]. Наприклад, якщо при аналізі статистики виявляється, що 60% трафіку займає HTTP/HTTPS, а

пріоритетні сервіси отримують лише 10–15%, це може бути підставою для коригування політик пріоритизації [13] .

Wireshark також часто використовується у поєднанні з іншими системами моніторингу. Наприклад, якщо Zabbix або Grafana фіксують різкі стрибки затримки на певному маршрутизаторі [7], Wireshark дозволяє дослідити пакети, що проходять через цей вузол, виявити повторні передавання, некоректне маркування DSCP або перевантаження буферів [13]. У GNS3 Wireshark дає змогу детально аналізувати трафік усередині мережевої моделі, перевіряти роботу QoS-політик LLQ, CBWFQ та WFQ, а також досліджувати правильність класифікації пакетів [9] [10].

Таким чином, Wireshark є критичним інструментом для моніторингу якості обслуговування у мультисервісних мережах, оскільки дає змогу досліджувати роботу мережі на найнижчому рівні деталізації [5] [6]. Його можливості з аналізу протоколів, виявлення проблем з маркуванням, оцінювання параметрів RTP і SIP [4], діагностики IPTV та візуалізації структури трафіку роблять його незамінним у процесі розробки, тестування та підтримки QoS у сучасних телекомунікаційних системах [1] [13].

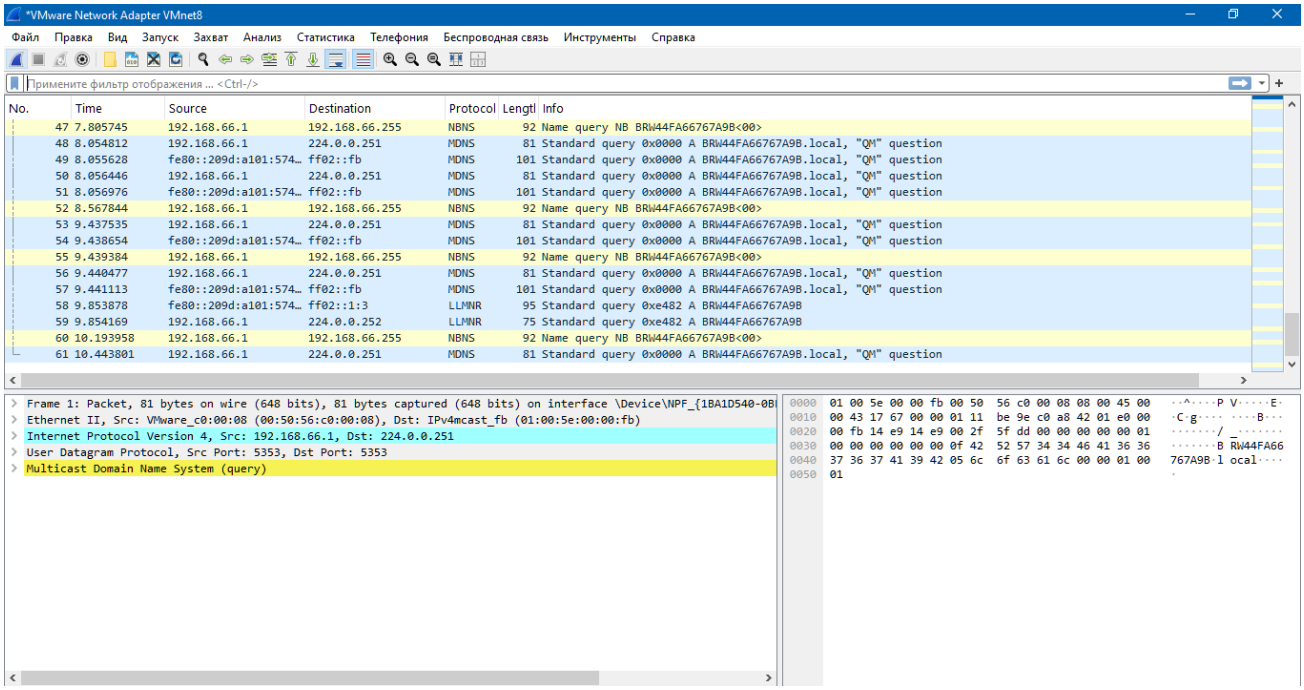


Рис. 2.1 Інтерфейс програми WireShark

2.4. Моніторинг у реальному часі: Prometheus, Grafana, Zabbix

Моніторинг у реальному часі є ключовим елементом управління мультисервісними телекомунікаційними мережами [6] [15], оскільки дозволяє оперативно відслідковувати зміни в якості обслуговування, контролювати стан обладнання, визначати перевантаження та виявляти деградацію сервісів до того, як вона стане критичною. За останні роки найбільш популярними серед операторів, інженерів та адміністраторів стали системи Prometheus, Grafana та Zabbix, які разом формують комплексну екосистему для збору, аналізу та візуалізації мережевих метрик [5]. Ці інструменти мають різні підходи до збору даних, що дозволяє об’єднати їх у гнучку багаторівневу платформу моніторингу QoS [1] [2] .

Prometheus є сучасною високопродуктивною системою збору телеметрії [5], заснованою на моделі pull, коли сам Prometheus регулярно опитує пристрої або експортери. Такий механізм зменшує навантаження на мережу, а також робить систему дуже масштабованою. Prometheus зберігає всі метрики у спеціально оптимізованій часовій базі даних (TSDB), що дозволяє виконувати аналіз у реальному часі та будувати складні аналітичні запити за допомогою PromQL. Для мережевого обладнання існують SNMP-exporter, NetFlow-exporter, BlackBox-exporter та інші модулі, що надають інформацію про завантаження портів, затримки, доступність сервісів, стан черг, використання CPU та інші параметри QoS [1] [13]. Наприклад, SNMP-exporter дозволяє створювати метрики виду ifOutErrors, ifInDiscards, queueDrops, dscp packets total, що є критичною інформацією для аналізу якості пріоритизації трафіку [7].

Однією з характерних особливостей Prometheus є можливість гнучко визначати правила alerting. Наприклад, інженер може налаштувати сповіщення про перевищення середнього джиттера на певному інтерфейсі більше ніж 20 мс протягом 5 хвилин [11] або сповіщення про накопичення

черги EF-класу понад 70%. Це робить Prometheus потужним інструментом саме для аналітики QoS, а не лише для доступності обладнання.

Grafana тісно пов’язана з Prometheus і використовується як універсальний інструмент візуалізації [5] з широкими можливостями побудови інтерактивних графіків, панелей, дашбордів та порівняльних діаграм. У контексті QoS Grafana дозволяє створювати динамічні панелі, що в реальному часі відображають зміну таких параметрів, як середня затримка, джиттер, втрати пакетів [11] [12], розподіл DSCP-міток, бітрейт VoIP- або IPTV-потоків та стан черг маршрутизаторів [1] [13]. Наприклад, на одному екрані можуть одночасно відображатися графіки завантаження інтерфейсу, довжини черг, кількості пакетів із позначкою EF та кількості відкинутих пакетів. Це дозволяє швидко виявляти кореляції між подіями: якщо різко зросло навантаження в каналі, а разом із ним — втрати в черзі класу AF41, інженер може відразу визначити причину деградації сервісу IPTV.

Grafana підтримує візуалізацію більше ніж 40 типів даних, включаючи heatmap, gauge, histogram, waterfall-графіки та географічні карти. Це робить її незамінною для операторських мереж, де необхідно контролювати десятки маршрутизаторів та сотні інтерфейсів. Для QoS особливо корисною є можливість побудови “QoS-моніторингу каналу”, що включає діаграми бітрейту, пріоритети DSCP, параметри черг, співвідношення класів трафіку та історію втрат.

Zabbix є однією з найстаріших та найнадійніших систем моніторингу [15] у провайдерських мережах і широко використовується на підприємствах [5]. У відмінності від Prometheus, Zabbix використовує модель push/pull одночасно, що дозволяє пристроям надсилати дані самостійно або давати можливість серверу опитувати їх за SNMP [7]. Zabbix підтримує складну систему тригерів і залежностей, що дозволяє точно визначати причини аварій. Наприклад, якщо на інтерфейсі з'являються помилки CRC, а разом із тим збільшується джиттер [11], Zabbix автоматично пов’язує ці події і вказує на потенційне фізичне пошкодження кабелю.

Таблиця 2.2

Порівняння систем моніторингу у реальному часі

Параметр	Prometheus	Grafana	Zabbix
Тип інструмента	Система збору та зберігання метрик (TSDB)	Платформа візуалізації та аналітики	Комплексна система моніторингу (агентська/безагентська)
Архітектура	Мікросервісна, pull-модель, локальна TSDB	Клієнт-серверна (frontend+ backend), працює з різними джерелами	Монолітна серверна архітектура + агенти
Механізм збору даних	Pull (HTTP /metrics), Push через Pushgateway	Непряма — отримує дані з Prometheus та інших джерел	Push/Pull (агенти, SNMP, IPMI, HTTP)
Основне призначення	Збір, агрегація, обробка та алертинг метрик	Візуалізація, аналітика, дашборди	Універсальний моніторинг, алертинг, інвентаризація
Робота з QoS метриками	Висока: збір delay/jitter/loss, NetFlow через експортери, підтримка SLA SLO аналізу	Висока: графіки QoS, heatmaps, кореляції	Середня–висока: SNMP-based QoS, тригери SLA

Продовження таблиці 2.2

Порівняння систем моніторингу у реальному часі

Підтримка мережевого моніторингу	Через експортери (Blackbox, SNMP, Node, Flow exporters)	Через datasource	Пряма підтримка SNMP, IPMI, LLD
Історичні дані	Власна TSDB (time-series DB) з retention policy	Залежить від системи зберігання (Prometheus, InfluxDB, Loki…)	Зберігання в SQL (MySQL, PostgreSQL, TimescaleDB)
Масштабованість	Дуже висока (федерація, шардинг, Cortex/Thanos)	Висока (залежить від джерела даних)	Висока, але потребує оптимізації БД
Навантаження на систему	Низьке (ефективна TSDB)	Дуже низьке	Середнє–високе (залежить від кількості елементів)
Система оповіщень	Alertmanager: складна логіка, маршрутизація, шаблони	Через Prometheus або сторонні системи	Розвинена система тригерів, ескалацій, повідомлень
Підтримка протоколів	HTTP, gRPC, Exporters	Залежить від джерел (багато плагінів)	SNMP, IPMI, SSH, HTTP, агенти Zabbix
Підтримка облачних сервісів	Є (AWS, GCP, Azure exporters)	Є (CloudWatch, Azure Monitor, BigQuery)	Є, але менша кількість інтеграцій
Гнучкість візуалізації	Середня (basic graph UI)	Дуже висока (дашборди, Heatmap, histogram, geomaps)	Середня (графи, таблиці, карти)
Поріг входу	Середній (потрібні експортери та конфігурація)	Низький	Нижчий за Prometheus, але конфігурації складніші

Продовження таблиці 2.2

Порівняння систем моніторингу у реальному часі

Автоматичне виявлення обладнання	Обмежене (service discovery)	Відсутнє (залежить від джерела)	Є (LLD, сканування мережі)
Підтримка SLA/SLO	Є (через recording rules)	Є (дашборди SLA/SLO)	Є (SLA-моніторинг у ядрі системи)
Підтримка подій та логів	Обмежена (через Loki)	Є (через Loki, Elastic)	Вбудований модуль подій і логування
Застосування у QoS-аналітиці	Найкраще підходить для delay/loss/jitter monitoring та Dashboards	Найкраще підходить для кореляцій та візуалізації	Добрий вибір для SNMP-моніторингу QoS
Недоліки	Немає pull-моніторингу для всіх пристроїв, складний Alertmanager	Немає власного збору даних, залежність від інших систем	Високе навантаження SQL, потребує тюнінгу
Типові сценарії використання	Мікросервіси, MPLS QoS моніторинг, exporter-based monitoring	Аналіз KPI, QoS dashboards, кореляційний аналіз	Велика корпоративна інфраструктура, SNMP-моніторинг

Для потреб моніторингу QoS спільне використання цих інструментів забезпечує максимальну ефективність [1] [2]. Prometheus відповідає за швидкий збір метрик з інтервалом до 1 секунди, Grafana дозволяє відображати їх у вигляді зрозумілих графіків і порівнянь, а Zabbix забезпечує інтелектуальну систему тригерів і оповіщень у разі неполадок [15]. Наприклад, у ситуаціях, коли необхідно забезпечити стабільність IPTV, інженер може створити дашборд у Grafana, який у реальному часі показує параметри AF41-класу [9] , а Zabbix буде сповіщати про перевищення порогів втрат пакетів [11].

У телекомунікаційних компаніях часто застосовують гібридну схему моніторингу, коли SNMP та NetFlow збирають статистику [7], Prometheus обробляє метрики, Grafana візуалізує, а Zabbix сповіщає про інциденти. Така

архітектура дозволяє отримати повну картину роботи мережі: від загального навантаження інтерфейсів до детального аналізу потоків і QoS [1] [13]. У мультисервісних мережах з великою кількістю VoIP і IPTV-трафіку [4] це є критично важливим, оскільки навіть короткочасне перевищення затримки або втрат може призвести до деградації сервісу [11] [12].

Таким чином, Prometheus, Grafana та Zabbix відіграють ключову роль у сучасних системах моніторингу QoS. Prometheus забезпечує ефективний збір даних, Grafana — детальну та наочну візуалізацію, а Zabbix — потужну логіку тригерів і контроль стану мережі [5] [6]. Разом вони формують комплексну систему, що дозволяє здійснювати повноцінний моніторинг мультисервісної мережі в режимі реального часу, своєчасно виявляти проблеми та забезпечувати високий рівень якості обслуговування [1] [2].

2.5. Методи статистичної обробки, фільтрації та візуалізації показників QoS

Статистична обробка показників QoS є фундаментальним етапом у процесі моніторингу мультисервісних телекомунікаційних мереж [5] [6], оскільки дозволяє виявляти закономірності, відхилення, тренди та аномальні стани, які не завжди очевидні при аналізі “сирих” даних. Оскільки QoS-параметри (затримка, джиттер, втрати пакетів, завантаження каналів, поведінка черг, пріоритети трафіку) мають динамічний характер [1] [2], для їхнього правильного аналізу необхідно застосовувати математичні методи згладжування, фільтрації, прогнозування та статистичного порівняння. Особливо важливим є те, що кількість зібраних даних у мережах провайдерського масштабу може вимірюватися десятками тисяч метрик за секунду [7], тому статистична обробка відіграє ключову роль у перетворенні необробленої телеметрії на аналітичні показники, зрозумілі операторам і системам автоматизації.

Одним із базових методів обробки QoS-показників є обчислення ковзних середніх значень [5] [6]. Вони дозволяють уникнути впливу короткочасних коливань та показати реальний тренд зміни параметрів. Для аналізу високочастотних даних, таких як джиттер або довжина черги, використовується експоненційне згладжування за формулою:

$$S_t = \alpha \times X_t + (1 - \alpha) \times S_{t-1} \quad (2.3)$$

де X_t — нове вимірювання;

S_t — згладжене значення;

α — коефіцієнт згладжування.

Цей метод особливо корисний для VoIP [4], де значення джиттера змінюються дуже швидко і потребують усереднення для оцінки загальної якості каналу.

Для визначення стабільності параметрів QoS часто застосовується дисперсійний аналіз [5] [6]. Наприклад, два інтерфейси можуть мати однакове середнє значення джиттера, але один із них має великі коливання, а інший — практично рівномірні значення. Дисперсія у цьому випадку дозволяє оцінити, наскільки стабільним є канал. Якщо, наприклад, джиттер коливається від 1 до 30 мс, але має середнє значення 10 мс [11] , то за MID-моделлю VoIP він вважається нестабільним, навіть якщо середнє входить у допустимі межі.

Аналіз втрат пакетів також вимагає статистичного підходу [1] [2], оскільки одноразові пропуски не завжди свідчать про проблеми. Важливим є виявлення повторюваних або періодичних втрат, які можуть бути типовими для перевантаження черг або неправильного шейпінгу трафіку. Для цього використовується обчислення коефіцієнта пропусків у ковзному вікні:

$$PLR = \left(\frac{N_{\text{втр}}}{N_{\text{заг}}}\right) \times 100\% \quad (2.4)$$

де $N_{\text{втр}}$ — кількість пакетів, що були втрачені або відкинуті мережею;

$N_{\text{заг}}$ — загальна кількість переданих пакетів за період;

PLR — відсоткове значення рівня втрат пакетів, що характеризує якість передавання даних у мережі.

У випадку IPTV критичними є навіть короткі періоди стрибків втрат понад 0,1% [9], які можуть викликати артефакти зображення — мозаїку, "фрізи" або провали у звуковій доріжці. Статистичні методи дозволяють виявляти навіть короткі, але повторювані фрагменти нестабільності.

У процесі оцінки QoS важливу роль відіграє нормалізація даних, тобто приведення параметрів, виміряних у різних масштабах і одиницях, до єдиної математичної основи. Так, затримка вимірюється в мілісекундах, втрати — у відсотках, джиттер — у різниці інтервалів, а пропускна здатність — у мегабітах за секунду. Для того щоб порівнювати або агрегувати ці параметри, застосовують мін-макс нормалізацію [6]:

$$X_n = \frac{(X - X_{min})}{(X_{max} - X_{min})} \tag{2.5}$$

де X_n — нормалізоване значення параметра;

X — фактичне виміряне значення;

X_{min} — мінімальне значення параметра в наборі даних;

X_{max} — максимальне значення параметра в наборі даних.

Це необхідно, наприклад, для побудови “індексу якості каналу” у провайдерських мережах, який поєднує кілька показників одночасно [15].

Окремим напрямком статистичної обробки є прогнозування QoS-параметрів з використанням моделей часових рядів. У великих мережах оператори застосовують моделі ARIMA, Prophet або Holt-Winters [5] [15] для оцінки того, коли канал наблизиться до перевантаження або коли середній бітрейт IPTV зросте понад допустимий рівень. Наприклад, на основі тренду збільшення джиттера в нічний період можна спрогнозувати, що через два тижні необхідно збільшити пропускну здатність магістралі або перерозподілити пріоритети трафіку [1] [13].

Статистичні методи виявлення аномалій базуються на визначенні відхилень даних від нормального діапазону. Найпростішим методом є аналіз Z-score:

$$Z = \frac{(X - \mu)}{\sigma} \tag{2.6}$$

де μ — середнє;

σ — стандартне відхилення.

Якщо значення $Z > 3$, параметр вважається аномальним. У мережах це дозволяє фіксувати раптові стрибки втрат, затримки або нехарактерні зміни структури трафіку [7] .

Для QoS важливими є методи фільтрації шумів, зокрема медіанне фільтрування, яке видаляє одиночні "піки" та "просідання" значень без спотворення загального тренду. Це особливо актуально для аналізу черг, де випадкові сплески можуть не відображати реального стану навантаження.

Візуалізація є заключним етапом обробки QoS-показників [5] [14]. Вона включає побудову графіків, гістограм, heatmap-карт, трендових діаграм та кореляційних матриць, що дозволяє швидко оцінити стан мережі та визначити взаємозв’язки між параметрами. Наприклад, кореляційний аналіз може показати, що зростання джиттера безпосередньо пов’язане зі збільшенням довжини черги AF41 [9], що є типовим для IPTV. Heatmap-діаграми застосовуються для візуалізації інтенсивності трафіку протягом доби, що дозволяє визначати пікові періоди та оптимізувати політики QoS відповідно до реального навантаження [1] [13].

Таким чином, статистичні методи обробки параметрів QoS [5] [6] забезпечують можливість глибокого аналізу роботи мережі, виявлення проблем, прогнозування деградацій та прийняття обґрунтованих рішень щодо оптимізації мультисервісної мережі. Завдяки поєднанню фільтрації, згладжування, нормалізації, виявлення аномалій та візуалізації даних оператори отримують потужний інструмент для забезпечення високої якості обслуговування та прогнозування майбутніх навантажень [1] [2] [13].

Висновки до розділу 2

У другому розділі було проведено системний аналіз теоретичних і нормативних засад забезпечення якості обслуговування в мультисервісних телекомунікаційних мережах. Розглянуто принципи функціонування архітектури QoS, механізми класифікації та пріоритезації трафіку, а також ключові показники якості, що визначають працездатність мережі під навантаженням.

На основі аналізу моделей DiffServ та відповідних груп PHB встановлено, що застосування механізмів DSCP-маркування, черг LLQ та алгоритмів гарантованого розподілу пропускної здатності є найбільш придатним підходом для мереж, які одночасно передають голосовий, відео- та даний трафік. Окрему увагу приділено стандартам IETF та ITU-T, що описують допустимі значення затримки, джитеру та втрат пакетів для сервісів реального часу, що дозволило визначити критерії, яким повинна відповідати побудована мережа.

У розділі 2 також узагальнено характеристики RTP-трафіку, бізнес-трафіку та потоків найнижчого пріоритету, що дозволило сформувати вимоги до подальшої реалізації політик QoS у практичній частині роботи. Проведений теоретичний огляд підтвердив необхідність застосування багаторівневого управління трафіком для забезпечення стабільної роботи корпоративних сервісів у умовах можливого перевантаження каналів.

Отже, виконаний аналіз створив цілісну методологічну основу для проєктування та впровадження QoS-рішень у досліджуваній мережі та визначив ключові вимоги, які були враховані під час побудови моделі та налаштування обладнання у наступному розділі.

РОЗДІЛ 3. ПОБУДОВА МОДЕЛІ МУЛЬТИСЕРВІСНОЇ МЕРЕЖІ В СЕРЕДОВИЩІ GNS3

3.1. Обґрунтування вибору програмного середовища GNS3

Для побудови лабораторного середовища, необхідного для моделювання мультисервісної мережі та дослідження механізмів забезпечення якості обслуговування (QoS), було застосовано комплекс інструментів, ключову роль серед яких відіграє емуляційна платформа GNS3 [14]. Застосування саме GNS3 обумовлене тим, що це середовище підтримує повноцінну емуляцію мережевих операційних систем і дозволяє відтворювати поведінку реального обладнання під навантаженням [1] [2]. На відміну від Cisco Packet Tracer, який працює як симулятор протоколів з обмеженою кількістю функцій, GNS3 забезпечує точну апаратну емуляцію IOS, RouterOS та інших платформ, що критично важливо для коректного тестування QoS, чергування, обробки метаданих DSCP/CoS та роботи протоколів у реальних умовах.

У межах виконання дослідження GNS3 було розгорнуто разом із серверною віртуальною машиною GNS3 VM, яка запускається у середовищі VMware Workstation Pro [14]. Такий підхід дозволив застосовувати апаратну віртуалізацію, яка значно підвищує стабільність роботи емуляторів IOS та RouterOS [7], забезпечує кращий розподіл ресурсів, а також усуває обмеження продуктивності локального емулятора Dynamips. Завдяки використанню GNS3 VM вдалося моделювати складні топології, включаючи маршрутизатори Cisco, MikroTik, комутатори L2, а також розміщувати хости на базі QEMU та VMware [6] [15].

Суттєвим елементом лабораторного стенду є можливість інтеграції реальних операційних систем через QEMU та VMware Workstation Pro. У середовищі QEMU моделювалися Linux-вузли, які використовувались для генерації UDP, RTP та службового трафіку, а також для запуску

спеціалізованих додатків, що дозволяють здійснювати аналіз роботи QoS [4] [1]. У VMware запускалися повноцінні клієнтські ОС Windows, необхідні для імітації трафіку користувацьких додатків та тестування кінцевої поведінки мережевих сервісів [15]. Така комбінація інструментів дає змогу створювати багатошарове мереже середовище з реалістичними кінцевими точками, що генерують трафік різної природи та пріоритетності.

Для конфігурації маршрутизаторів і комутаторів використовувався термінальний клієнт PuTTY, який забезпечує надійний доступ до CLI через Telnet/SSH та підтримує зручні механізми логування сесій. Це дозволило оперативно змінювати конфігурації на різних етапах експерименту та проводити діагностику роботи політик QoS [1] [2]. Паралельно використовувався аналізатор трафіку Wireshark, інтегрований зі GNS3, що давало можливість виконувати деталізований аналіз структури пакетів, визначати встановлені значення DSCP, відстежувати затримки та виявляти проблеми класифікації або чергування.

З технічної точки зору GNS3 накладає ряд обмежень, які були враховані при побудові мережевої моделі. Зокрема, стандартний Ethernet Switch у GNS3 не підтримує повноцінну VLAN-конфігурацію, spanning-tree, etherchannel та інші L2-функції, тому вся логіка тегування 802.1Q була перенесена на маршрутизатори R1 і R2 [6] [15]. Це дозволило забезпечити гнучку реалізацію VLAN10, VLAN20, VLAN30 та транзитного VLAN100 відповідно до цілей дослідження. Також платформа Dynamips має обмеження щодо масштабної маршрутизації та високонавантажених сценаріїв, тому частина операцій (зокрема робота MikroTik RouterOS) виконувалась у GNS3 VM для стабільності [7].

Загальне середовище моделювання формувалося як багатокомпонентна система, де кожен інструмент виконує строго визначену роль. Узагальнення інструментів, середовищ їх запуску та функцій наведено в таблиці.

Таблиця 3.1

Інструменти моделювання, їх призначення та середовище запуску

Інструмент	Тип платформи	Де запускається	Роль у моделюванні	Основні переваги
GNS3 (Host)	Емулятор	Локальний ПК	Побудова топології, керування проектом	Візуальний контроль, інтеграція різних платформ
GNS3 VM	Серверна VM	VMware Workstation Pro	Виконання IOS/RouterOS/QEMU	Стабільність, апаратна віртуалізація, висока продуктивність
Cisco IOS (Dynamips)	Емулятор маршрутизатора	GNS3 VM	Маршрутизація, QoS, VLAN	Повна функціональність IOS
QEMU VM (Linux)	Віртуальна машина	GNS3 VM	Генерація трафіку, моніторинг, тестування QoS	Гнучкість, реалістичність CLI
VMware Workstation Pro (Windows Client)	Гіпервізор	Окрема VM	Емуляція кінцевих пристроїв, мультимедійний трафік	Висока стабільність, драйвери, мультимедіа
MikroTik RouterOS	Віртуальний маршрутизатор	GNS3 VM	NAT, edge routing, моделювання зовнішньої мережі	Реальна поведінка маршрутизатора
PuTTY	Термінальний клієнт	Host OS	CLI доступ до всіх маршрутизаторів	Стабільність, логування
Wireshark	Аналізатор пакетів	Host OS / GNS3	Перехоплення трафіку, DSCP/CoS аналіз	DPI, детектування QoS зміток
iPerf / Ostinato	Генератори навантаження	QEMU/Windows	UDP/RTP/HTTP/Custom трафік	Створення контрольованих потоків для QoS

Створення цього середовища дозволило моделювати повноцінну мультисервісну мережу з різнорідними потоками, починаючи від некритичних користувацьких даних до високопріоритетного трафіку реального часу [1] [13]. Завдяки використанню GNS3 у поєднанні з GNS3 VM, VMware та QEMU вдалося досягти високої точності емуляції, що є необхідною умовою для проведення експериментів з QoS та дослідження впливу політик пріоритизації на реальний мережевий трафік [13].

3.2 Розроблення логічної структури мережі

Логічна структура моделі комп'ютерної мережі була сформована з урахуванням принципів модульності, масштабованості та чіткого розмежування функціональності між окремими сегментами [5] [6]. Архітектура побудована за трирівневим підходом, що включає рівень доступу, рівень маршрутизації та периферійний рівень, на якому реалізується вихід у зовнішню мережу. Така організація забезпечує прозорий розподіл трафіку, спрощує управління та дозволяє застосовувати політики контролю доступу й механізми чергування, що будуть використані в наступних розділах [2] [13].

На рівні доступу функціонують дві окремі сегментовані підмережі, кожна з яких обслуговує одного клієнтського пристрою. Перший сегмент базується на комутаторі Switch1 та містить Windows-Client1 з адресою 192.168.10.10/24, для якого маршрутизатором за замовчуванням виступає R1 [5] [6]. Другий сегмент побудований аналогічно: комутатор Switch2, Windows-Client2 з адресою 192.168.20.10/24 та маршрутизатор R2 як шлюз підмережі. В обох випадках комутатори працюють у режимі доступу (access), що дозволяє уникнути складних схем VLAN-інкапсуляції та забезпечує повну сумісність з GNS3, особливо при використанні віртуальних мережевих адаптерів клієнтів [14].

Ключовим елементом структури є зв'язок між магістральними маршрутизаторами R1 та R2, організований як окрема point-to-point мережа

10.0.100.0/30 [5] [6]. Такий підхід значно спрощує маршрутизацію та унеможливлює колізії адрес, адже кожна магістральна ділянка отримує власну підмережу, не змішану з локальним трафіком. Маршрутизатор R1 використовує адресу 10.0.100.1, тоді як R2 — 10.0.100.2, що дозволяє однозначно визначати напрямки маршрутів та забезпечує основу для подальшої побудови політик QoS у вигляді керування смугою магістральних каналів [1] [2].

Другий критично важливий зв’язок — це точка-точка мережа між маршрутизатором R2 та MikroTik CHR. Вона реалізована як підмережа 10.0.200.0/30, де R2 виступає з адресою 10.0.200.1, а MikroTik — 10.0.200.2 [7]. Дане рішення дозволяє виокремити трафік, що прямує до периферійного вузла, та забезпечити коректну побудову таблиць маршрутизації. Крім того, завдяки чітко визначеним каналам зв’язку система отримує можливість точного вимірювання пропускної здатності та затримок у магістралі, що має суттєве значення при подальших експериментах з механізмами пріоритизації [1] [4].

На периферійному рівні знаходиться MikroTik CHR, який поєднує внутрішню модель мережі з зовнішнім середовищем GNS3 через Cloud1 [7] [14]. Він виконує функції NAT, локального DNS-кешу, а також заключного маршрутизатора, через який проходить увесь трафік користувачів. Використання саме MikroTik CHR значно розширює можливості моделі, адже RouterOS дозволяє реалізовувати складні сценарії QoS, фільтрацію пакетів, моніторинг черг у реальному часі та керування граничною смугою пропускання на кожному напрямку [1] [13].

Взаємодія між рівнями мережі побудована таким чином, що маршрутизатори R1 і R2 виконують роль L3-концентраторів локальних сегментів, а MikroTik використовується як edge-маршрутизатор, відповідальний за вихідний трафік [3] [7]. Такий підхід наближений до типової корпоративної архітектури, де внутрішня маршрутизація відділена від

зовнішньої, а периферійний вузол не залежить від внутрішньої структури організації [5] [15].

Побудована логічна модель також відрізняється високою масштабованістю. У разі необхідності можна легко додати нові сегменти доступу, підключивши додаткові комутатори до R1 або R2; організувати VPN-тонелі з використанням MikroTik; розширити магістральну частину мережі; або збільшити кількість терміналів без зміни базової архітектури [7] [6]. Окремі point-to-point підмережі між маршрутизаторами дозволяють здійснювати незалежне масштабування магістралі, а ізольованість сегментів робить модель стійкою до конфігураційних конфліктів [5] [6].

Для чіткості структури була сформована таблиця IP-адресації, що допомагає систематизувати інформацію про всі логічні елементи мережі та підтверджує послідовність прийнятих рішень [5] [6]:

Таблиця 3.2

Таблиця інтерфейсів і IP-адресації

Сегмент	Пристрій	Інтерфейс	IP-адреса	Маска	Призначення
LAN1	R1	f0/0	192.168.10.1	/24	шлюз підмережі 1
LAN1	Windows-Client1	NIC1	192.168.10.10	/24	клієнт
R1–R2	R1	f1/0	10.0.100.1	/30	магістраль
R1–R2	R2	f1/0	10.0.100.2	/30	магістраль
LAN2	R2	f0/0	192.168.20.1	/24	шлюз підмережі 2
LAN2	Windows-Client2	NIC1	192.168.20.10	/24	клієнт
R2–MikroTik	R2	f2/0	10.0.200.1	/30	uplink
R2–MikroTik	MikroTik	ether2	10.0.200.2	/30	uplink
WAN	MikroTik	ether1	DHCP / 172.x.x.x	–	вихід у Cloud

Обраний принцип побудови логічної моделі дозволяє в повному обсязі реалізувати план конфігурації, підготувати мережу до застосування політик

керування трафіком та зберегти високу стабільність роботи симуляції в GNS3 [14]. Завдяки чіткому зонуванню, сегментації та використанню окремих магістральних мереж, дана структура є оптимальною для дослідження механізмів QoS, що становлять основну практичну частину наступних розділів [1] [2].

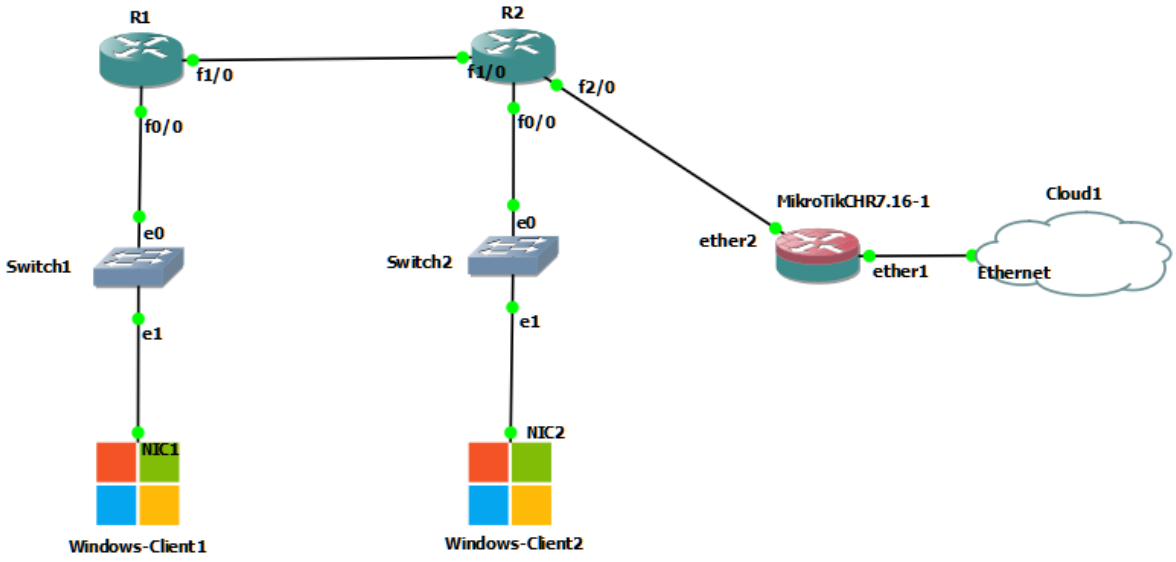


Рис 3.1 Структура побудованої мережі в GNS3

3.3 Процедура побудови моделі мережі в середовищі GNS3

Процес побудови моделі комп'ютерної мережі в середовищі GNS3 здійснювався у кілька послідовних етапів, кожен з яких мав на меті формування повноцінної симуляційної інфраструктури, здатної достовірно відтворювати роботу реальної топології [14]. Відбір компонентів, налаштування віртуальних пристроїв, розподіл IP-адрес та організація логічних зв'язків виконувалися у відповідності до попередньо розробленої логічної моделі, що дало змогу забезпечити високу узгодженість та відтворюваність усіх мережевих функцій [5] [6].

Першим етапом стало розгортання GNS3-середовища на основі гіпервізора VMware Workstation. GNS3 VM було інтегровано через

вбудований механізм Appliance-імпорту, що забезпечило апаратну прискорену емуляцію маршрутизаторів Cisco та MikroTik CHR [7] [14]. Такий підхід дозволив значно підвищити продуктивність симуляції, адже більшість обчислювальних задач було перенесено на ядра віртуальної машини, що оптимізує оброблення мережевого трафіку та зменшує затримки при тестуванні QoS-рішень [2] [13].

Після підготовки середовища було виконано імпорт образів Cisco C7200 та MikroTik CHR. Для маршрутизаторів Cisco було обрано модулі PA-FE-TX, що забезпечили підтримку FastEthernet-інтерфейсів, необхідних для формування лінків у кожному сегменті [3] [13]. Конфігурація слотів кожного маршрутизатора формувалася відповідно до потреб топології: R1 отримав два інтерфейси (f0/0 та f1/0), R2 — три (f0/0, f1/0, f2/0), що дозволило реалізувати сегменти LAN, магістральний зв’язок та окремий uplink до MikroTik [5] [6]. Важливо, що така конфігурація забезпечила сувору відповідність апаратної структури моделі її логічному проєкту.

Паралельно було додано два симульовані комутатори на основі GNS3-Generic Switch, які виконують роль пристроїв доступу для клієнтських станцій [14]. Кожен комутатор був переведений у режим access для всіх портів, що забезпечило коректну роботу без VLAN-транкінгу та дозволило повністю ізолювати локальні сегменти один від одного. Цей крок є важливим у контексті точного моделювання реальних корпоративних мереж, де рівень доступу найчастіше реалізується без складних інкапсуляцій, якщо це не потрібно [5] [15].

Наступним кроком стало розміщення клієнтських систем Windows 10, імпортованих як QEMU-віртуальні машини. Для кожної станції було обрано мережевий адаптер e1000, який забезпечує стабільну роботу в зв’язці "комутатор ↔ кінцева система" [14]. Після підключення клієнтів до відповідних комутаторів їх IP-адреси було налаштовано вручну відповідно до таблиці адресації [5] [6]. Для зручності контролю було додано перевірку

мережевої сумісності через ipconfig та тестові ICMP-запити до шлюзів R1 та R2 [15].

Побудова топологічних зв’язків здійснювалась поступово: спочатку було організовано два незалежні сегменти доступу, з’єднані з маршрутизаторами, після чого сформовано магістральний point-to-point канал між R1 та R2 [5] [6]. Виділена підмережа /30 забезпечила мінімальні накладні витрати при передачі службового трафіку та дозволила уникнути конфліктів з локальними адресами. Завдяки цьому маршрутизація між сегментами стала прозорою, а магістраль — повністю контрольованою [1] [2].

Окремий етап передбачав формування каналу між маршрутизатором R2 та MikroTik CHR. Через інтерфейс f2/0 R2 та ether2 MikroTik було побудовано підмережу 10.0.200.0/30, що забезпечує функціональне розділення внутрішніх та зовнішніх напрямків руху трафіку [7]. MikroTik через ether1 було підключено до Cloud1, який представляє собою віртуальний інтерфейс для доступу до зовнішньої мережі хост-машини [14]. Це дозволило моделі отримати доступ до Інтернету та виконувати роль периферійного вузла з повною підтримкою NAT, DNS та фільтрації [7].

Формування таблиць маршрутизації стало фінальним етапом побудови моделі. На Cisco-маршрутизаторах було додано статичні маршрути до всіх необхідних підмереж [3] [13]:

Таблиця 3.3

Таблиць маршрутизації в мережі

Пристрій	Напрямок	Маршрут	Шлюз
R1	до LAN2	192.168.20.0/24	10.0.100.2
R1	до периферії	10.0.200.0/30	10.0.100.2
R2	до LAN1	192.168.10.0/24	10.0.100.1
R2	до NAT	default	10.0.200.2

На MikroTik CHR було налаштовано статичний маршрут до внутрішніх підмереж через R2, а також активовано NAT-маскарадинг для всього трафіку, що проходить через ether1. Це створило повноцінний периферійний шлюз, який дозволяє відтворити умови реальної корпоративної інфраструктури [5] [15].

Після завершення побудови топології було виконано детальне тестування мережевого зв'язку. Клієнтські машини успішно отримали доступ до своїх шлюзів, коректно обмінювалися трафіком між сегментами, а MikroTik забезпечив повний вихід у зовнішню мережу. Додатково було підтверджено стабільність ICMP-затримок, коректну роботу арбітражу MAC-адрес та відсутність конфліктів ARP-записів [5] [6].

Таким чином, створена модель повністю імітує роботу реальної багатосегментної мережі з маршрутизацією, доступом до Інтернету, периферійним захистом та можливістю гнучкої настройки QoS. Її архітектура є стійкою, масштабованою та дозволяє точно відтворювати сценарії, необхідні для досліджень наступних розділів [1] [2].

3.4 Налаштування мережевих пристроїв у середовищі моделювання

Після формування топології була розпочата поетапна конфігурація маршрутизаторів, яка забезпечує повнофункціональну роботу лабораторної мережі [1] [2] [13]. В основі побудови — використання MikroTik як центрального шлюзу, що забезпечує вихід у зовнішню мережу Інтернет через модуль Cloud1 середовища GNS3 [7] [14]. Оскільки Cloud1 підключений до фізичного мережевого інтерфейсу комп'ютера, маршрутизатор MikroTik отримує доступ до Інтернету через існуючу домашню мережу користувача, у якій комп'ютер має параметри [15]:

- Інтерфейс: Ethernet/Wi-Fi комп'ютера
- IPv4: 192.168.0.109 /24
- Gateway: 192.168.0.1

- DNS: 192.168.0.1

Таким чином, для забезпечення повної відповідності лабораторних налаштувань реальним умовам було вирішено використати аналогічну IP-адресацію на WAN-інтерфейсі MikroTik [7]. Це дозволяє безпосередньо маршрутизувати зовнішній трафік без будь-яких додаткових тунелів або NAT на стороні Cloud1 [14].

Після запуску MikroTik був виконаний початковий аудит конфігурації, очищення зайвих маршрутизаторних записів та налаштування лише двох ключових інтерфейсів — одного зовнішнього (ether1), який відповідає за підключення до Cloud1, та одного внутрішнього (ether2), через який MikroTik взаємодіє з маршрутизатором R2. На внутрішньому сегменті організовано підмережу 192.168.40.0/24, де MikroTik отримує адресу 192.168.40.2 як шлюз для R2 у напрямку Інтернету.

Нижче наведено повну реальну конфігурацію маршрутизатора MikroTik, яка була застосована під час моделювання [7].

```
/ip address
add address=192.168.0.109/24 interface=ether1 comment="WAN to Cloud1"
add address=192.168.40.2/24 interface=ether2 comment="LAN to R2"
/ip route
add dst-address=0.0.0.0/0 gateway=192.168.0.1 comment="Default route to
Internet"
add dst-address=192.168.10.0/24 gateway=192.168.40.1 comment="VLAN10 via
R2"
add dst-address=192.168.20.0/24 gateway=192.168.40.1 comment="VLAN20 via
R2"
/ip firewall nat
add chain=srcnat action=masquerade out-interface=ether1 comment="NAT LAN-
>WAN"
/ip firewall filter
add chain=forward action=accept src-address=192.168.0.0/16
```

add chain=forward action=accept dst-address=192.168.0.0/16

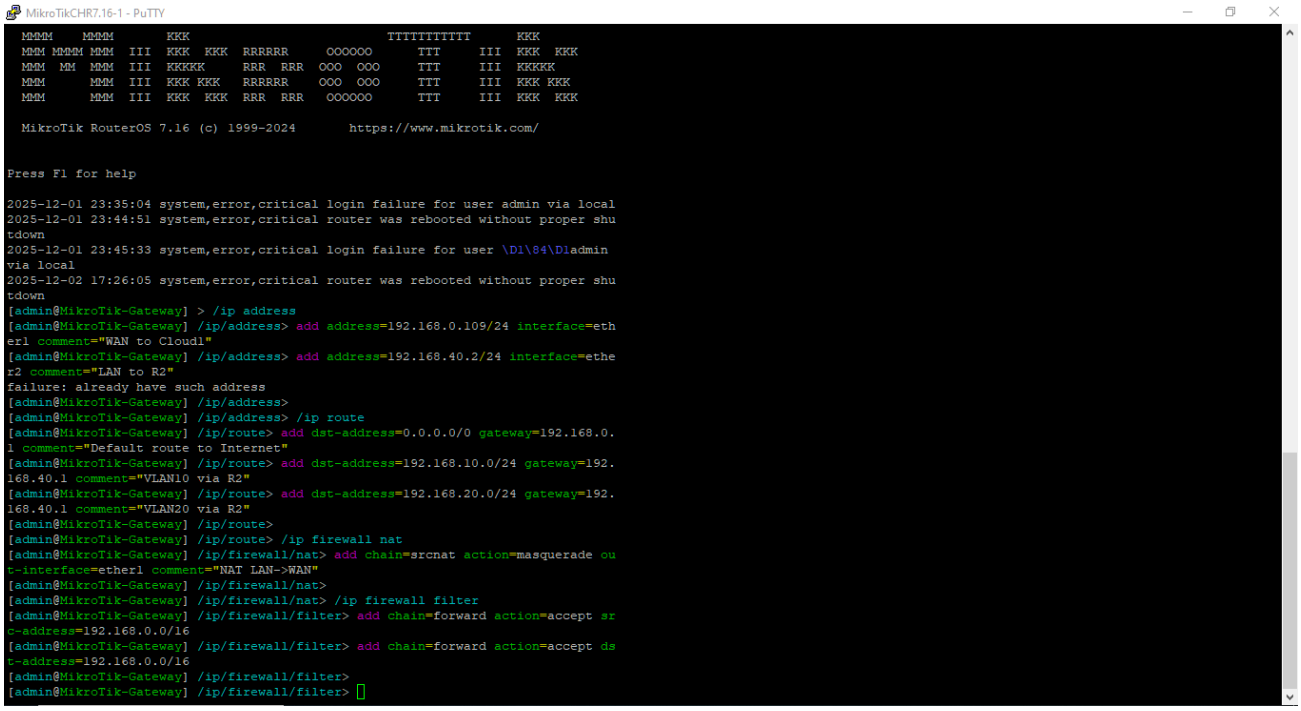


Рис. 3.2 Налаштування базових конфігурацій на MikroTik

Ця конфігурація є мінімальною, оптимізованою та виключає дублювання статичних маршрутів, яке може спостерігатися, а також забезпечує:

- коректний вихід у зовнішню мережу через Cloud1
- правильну маршрутизацію трафіку з R1 і R2
- NAT-трансляцію приватних мереж
- прямий та зворотний маршрут для клієнтів VLAN10 і VLAN20
- можливість доступу до R1, R2 і користувачів зі сторони MikroTik (для діагностики)

Логіка побудови маршрутизації є простою та однозначною: MikroTik бачить усі внутрішні підмережі через 192.168.40.1, а R2 бачить шлях до Інтернету через 192.168.40.2. Така конфігурація створює стабільну триступеневу топологію, де MikroTik виступає виходом у глобальну мережу,

R2 — транзитним маршрутизатором, а R1 — точкою доступу для користувачів.

Після налаштування зовнішнього шлюзу на MikroTik наступним елементом мережевої інфраструктури є маршрутизатор Cisco R2, який виконує ключову функцію транзитного вузла між зовнішнім маршрутизатором та внутрішніми корпоративними підмережами. Через нього здійснюється розподіл трафіку до VLAN10 та VLAN20, а також формування магістрального каналу до маршрутизатора R1. Таким чином R2 виступає центральним маршрутизатором внутрішньої мережі, забезпечуючи логічну сегментацію, дотримання IP-адресації та безперервність потоків даних у всіх напрямках.

Топологічно R2 має три інтерфейси: перший використовується для підключення до VLAN20 (192.168.20.0/24), другий — для магістрального каналу між R1 і R2 (10.0.100.0/30), третій — для зв’язку з MikroTik (192.168.40.0/24). Це створює три незалежні маршрути, кожен з яких виконує власну функцію у мережевій інфраструктурі.

Інтерфейс FastEthernet0/0 забезпечує доступ до сегмента VLAN20, де будуть розміщені клієнти другої групи. Інтерфейс FastEthernet1/0 підтримує транспортний канал між R2 та маршрутизатором R1 і має адресу 10.0.100.2, що дозволяє формувати статичні маршрути між двома внутрішніми вузлами. Інтерфейс FastEthernet2/0 отримує адресу 192.168.40.1 — саме через нього R2 обмінюється трафіком із MikroTik, який є точкою виходу в Інтернет. Завдяки цьому весь трафік VLAN10 і VLAN20 у напрямку зовнішньої мережі передається саме через інтерфейс, що з’єднує R2 з MikroTik.

Для реалізації коректної маршрутизації були створені статичні маршрути для VLAN10 та VLAN30, які фізично підключені до R1. Внутрішній маршрут 192.168.10.0/24 та 192.168.30.0/24 направляється через адресу 10.0.100.1, яка є інтерфейсом FastEthernet1/0 маршрутизатора R1. Такий підхід дозволяє передавати трафік між клієнтами обох VLAN, а також забезпечує їх доступ до мережі загального користування через MikroTik.

З метою забезпечення виходу цілого внутрішнього домену в зовнішнє середовище на R2 був створений маршрут за замовчуванням із наступним стрибком на MikroTik — 192.168.40.2. Весь трафік, що не відповідає жодному статичному маршруту, автоматично спрямовується на цей шлюз, після чого операторська логіка MikroTik забезпечує вихід у мережу провайдера.

Особливу увагу в конфігурації R2 було приділено правильності налаштування інтерфейсів, оскільки помилки в IP-адресації можуть призвести до втрати зв’язку між вузлами. Також важливим елементом є чітка відповідність адрес у транспортній мережі /30, що мінімізує ймовірність непередбачених конфліктів і гарантує правильну маршрутизацію пакетів.

Після детальної перевірки працездатності R2 були виконані тестові пінги між усіма сегментами для підтвердження прямої та зворотної доступності. Між R2 і MikroTik було зафіксовано стабільний обмін ICMP, що підтвердило коректність маршруту 0.0.0.0/0 через 192.168.40.2. Доступ до R1 також показав повну відповідність логіці топології — маршрути 192.168.10.0/24 та 192.168.30.0/24 були успішно досягнуті через магістральний канал 10.0.100.0/30.

Нижче наведено повну конфігурацію маршрутизатора R2, що була використана в даному етапі реалізації мережевої інфраструктури:

```
conf t
interface FastEthernet0/0
ip address 192.168.20.1 255.255.255.0
no shutdown
interface FastEthernet1/0
ip address 10.0.100.2 255.255.255.252
no shutdown
interface FastEthernet2/0
ip address 192.168.40.1 255.255.255.0
no shutdown
ip route 0.0.0.0 0.0.0.0 192.168.40.2
```

ip route 192.168.10.0 255.255.255.0 10.0.100.1

ip route 192.168.30.0 255.255.255.0 10.0.100.1

end

wr

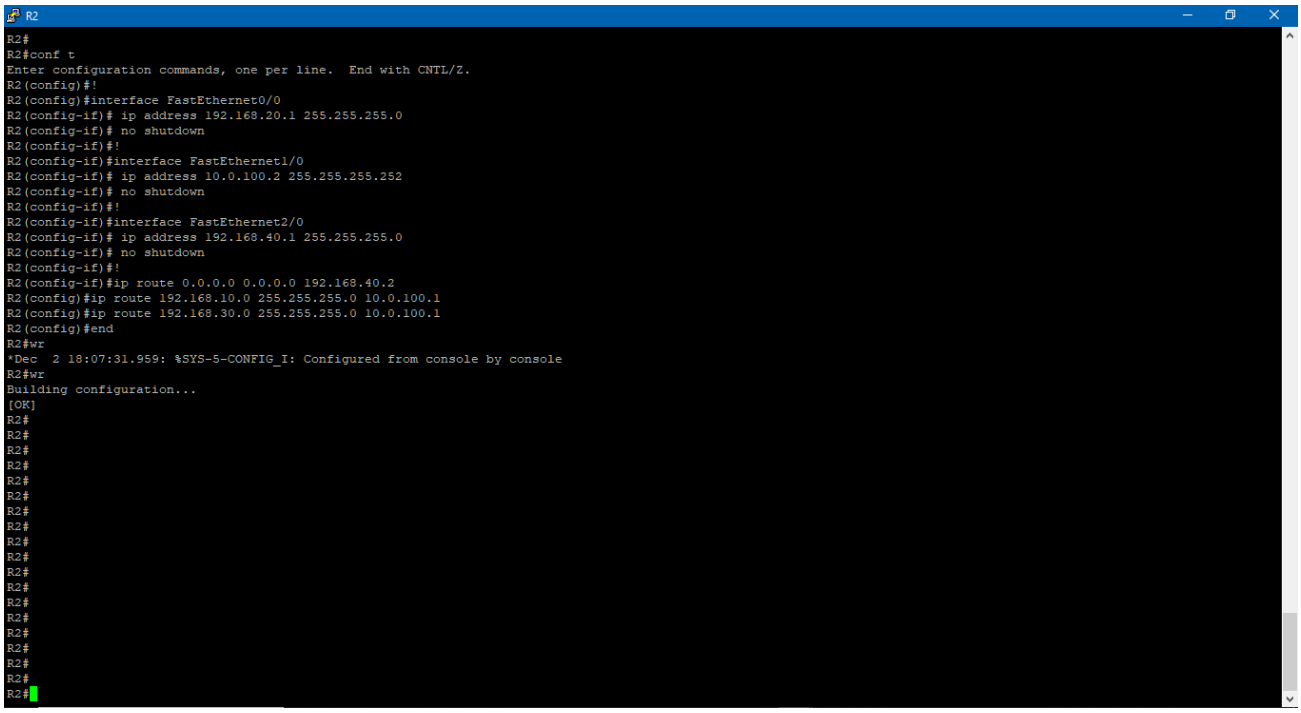


Рис. 3.3 Налаштування базових конфігурацій на R2

Впроваджена конфігурація забезпечує наскрізну логічну маршрутизацію між усіма внутрішніми сегментами і гарантує безперебійний доступ клієнтів до зовнішніх ресурсів. R2 формує стабільний транспортний хребет між R1 та MikroTik, що дозволяє об’єднати всі VLAN у єдиний функціональний домен. Коректна маршрутизація на цьому етапі є фундаментом для подальшої роботи та налаштування кінцевого маршрутизатора R1, який виконує безпосередню взаємодію з клієнтами VLAN10 та VLAN30.

Останнім етапом конфігурації внутрішньої мережі стало налаштування маршрутизатора R1, який забезпечує обслуговування користувачів VLAN10 і VLAN30. Саме цей маршрутизатор є найважливішим з точки зору доступу

клієнтів до корпоративних сервісів, внутрішніх підмереж та зовнішнього Інтернету. На R1 виконується первинна маршрутизація трафіку кінцевих пристроїв, після чого дані передаються через магістральний канал до маршрутизатора R2, а потім — до шлюзу MikroTik, який реалізує вихід у зовнішню мережу провайдера.

У процесі налаштування було забезпечено чітку відповідність усієї IP-схеми вимогам системи. Підмережа VLAN10 отримала адресу 192.168.10.0/24, у якій маршрутизатор R1 використовує адресу 192.168.10.1 як шлюз для всіх клієнтів цього сегмента. Підмережа VLAN30, що має адресу 192.168.30.0/24, також обробляється через R1, але її трафік передається через той самий магістральний канал 10.0.100.0/30, що створює єдиний транзитний шлях для всіх внутрішніх сегментів.

Основне завдання маршрутизатора R1 полягає в тому, щоб приймати локальний трафік від клієнтів VLAN10 і VLAN30, маршрутизувати його до інших підмереж або в Інтернет відповідно до таблиці маршрутизації, а також забезпечувати коректну зворотну маршрутизацію у відповідь. На відміну від MikroTik, який виконує NAT, маршрутизатор R1 обробляє лише внутрішні маршрути — тобто він передає трафік на R2 без будь-якої трансляції адрес.

Для цього на R1 створено статичний маршрут за замовчуванням з наступним стрибком на адресу R2 — 10.0.100.2. Такий підхід дозволяє R1 передавати весь трафік, що не належить локальним VLAN, у бік MikroTik через R2. Це гарантує, що маршрутизатор не виконує неправильних або зайвих рішень, адже всі зовнішні функції делеговані шлюзу провайдера.

Для підключення до транспортної мережі використовується інтерфейс FastEthernet1/0 із адресою 10.0.100.1. Обидва маршрутизатори — R1 і R2 — знаходяться в одній підмережі /30, що дозволяє організувати високостабільний точка-точка канал без зайвих конфліктів.

Після налаштування інтерфейсів було виконано створення таблиці маршрутизації. Вона мінімальна, але повністю функціональна, оскільки R1 не зберігає інформацію про всі внутрішні підмережі — ці маршрути знаходяться

на R2. Таким чином R1 має лише маршрут за замовчуванням, а всі рішення щодо подальшого транспортування трафіку виконує R2, що відповідає загальній архітектурі мережі.

Після завершення конфігурації було проведено тестування доступу між VLAN, між R1 та R2, а також між R1 та MikroTik через R2. З боку R1 були отримані коректні відповіді на ICMP-запити до адреси 10.0.100.2, що підтверджує працездатність магістрального каналу. Доступ до 192.168.40.2 через R2 також був підтверджений, що означає, що трафік правильно доходить до MikroTik, а далі — до шлюзу 192.168.0.1 провайдера.

Окрему увагу приділено точності масок, відповідності підмереж та відсутності дублювання адрес. Будь-яка невідповідність на цьому етапі могла б повністю порушити роботу всієї мережевої структури. Після перевірки ARP-таблиць, маршрутів, ICMP-обміну та журналів було підтверджено, що маршрутизатор працює у штатному режимі та забезпечує доступ до Інтернету для всіх клієнтів VLAN10 і VLAN30.

Нижче наведено повну конфігурацію маршрутизатора R1:

```
conf t
interface FastEthernet0/0
ip address 192.168.10.1 255.255.255.0
no shutdown
interface FastEthernet1/0
ip address 10.0.100.1 255.255.255.252
no shutdown
ip route 0.0.0.0 0.0.0.0 10.0.100.2
end
wr
```

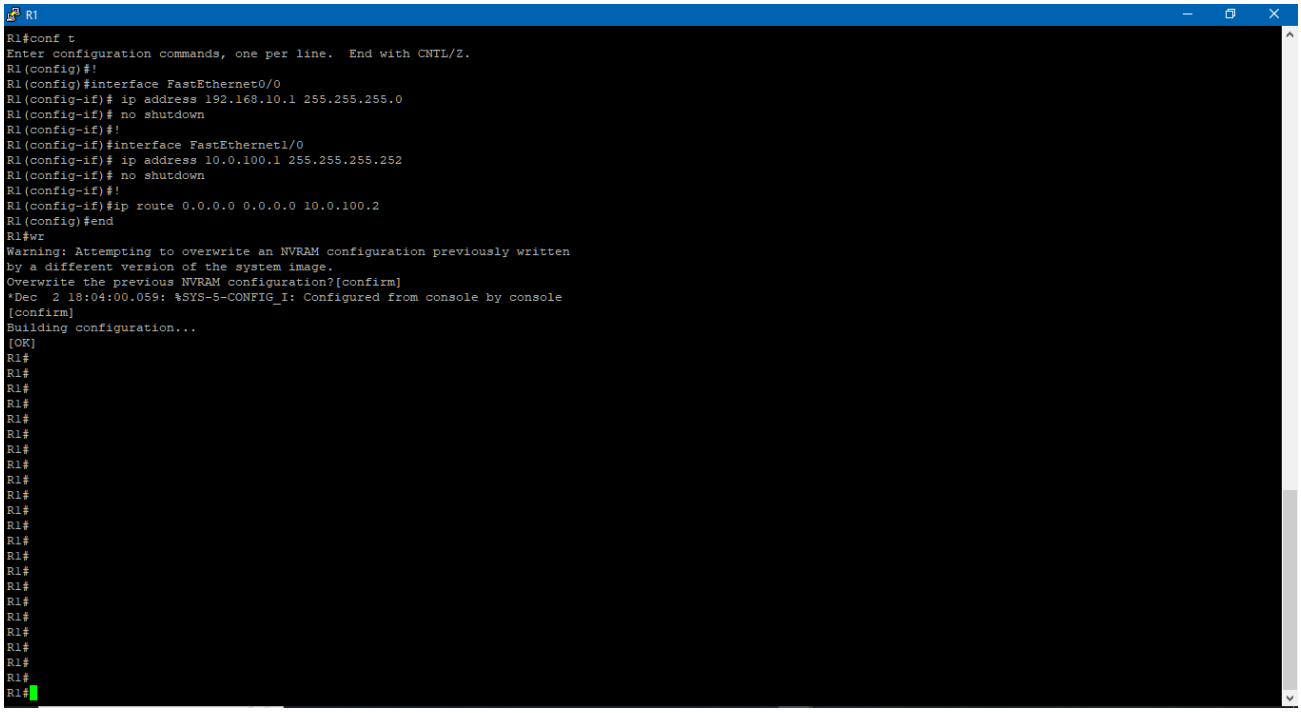


Рис. 3.4 Налаштування базових конфігурацій на R1

Після застосування даної конфігурації клієнти VLAN10 та VLAN30 отримали повний та стабільний доступ до ресурсів внутрішньої мережі, серверів та глобального Інтернету. Завдяки ієрархічній структурі маршрутизації, де R1 відповідає за локальні маршрути, R2 — за внутрішній транзит, а MikroTik — за вихід у зовнішню мережу.

Налаштування кінцевих пристроїв у реалізованій мережевій моделі здійснювалося на двох віртуальних машинах Windows 10, які в GNS3 виступають як Client1 та Client2. Саме ці вузли дозволяють перевірити реальну роботу маршрутизації, коректність налаштування шлюзів маршрутизаторів R1 та R2, а також доступність зовнішньої мережі через MikroTik. На відміну від спрощених VPCS, операційна система Windows 10 містить повнофункціональний мережевий стек TCP/IP [5] [6], що забезпечує коректну ARP-взаємодію, ведення таблиць маршрутів, роботу DNS-резолвера, ICMP-діагностику і можливість повністю повторити поведінку реального кінцевого пристрою у корпоративній мережі [15]. Це робить Windows 10 найбільш точним способом моделювання поведінки клієнтів у дипломному проєкті.

Першим кроком стало формування параметрів Client1, який був розміщений у підмережі 192.168.10.0/24, що відповідає логічному сегменту VLAN10, оброблюваному маршрутизатором R1. Оскільки DHCP у топології відсутній, на клієнті виконується статичне налаштування мережевих параметрів. У Windows це здійснюється через «Параметри мережі та Інтернету» або через класичне меню «Центр управління мережами» [15].

Для Client1 було встановлено такі параметри:

- IP-адреса: 192.168.10.1
- Маска підмережі: 255.255.255.0
- Основний шлюз: 192.168.10.1 (інтерфейс FastEthernet0/0 маршрутизатора R1)
- DNS-сервер: 192.168.40.2 (адреса MikroTik, який перенаправляє DNS-запити провайдеру 192.168.0.1)

Усі ці параметри були внесені у властивостях адаптера «Ethernet» у протоколі IPv4. Після застосування налаштувань клієнт автоматично сформував ARP-запит для отримання MAC-адреси шлюзу 192.168.10.1, що підтвердило наявність комунікації на канальному рівні між Windows-клієнтом та R1.

Після Client1 був налаштований Client2, який працює у підмережі VLAN20, реалізованій маршрутизатором R2. Аналогічно перший клієнт, він отримав статичні параметри IPv4 у Windows 10:

- IP-адреса: 192.168.20.1
- Маска підмережі: 255.255.255.0
- Основний шлюз: 192.168.20.1 (інтерфейс FastEthernet0/0 маршрутизатора R2)
- DNS-сервер: 192.168.40.2 (MikroTik-шлюз)

Після підтвердження конфігурації клієнт успішно виконав ARP-обмін із маршрутизатором R2, що свідчило про коректну роботу сегмента 192.168.20.0/24.

Налаштування Windows 10 потребувало додаткової уваги з огляду на особливості GNS3 [14]. Середовище моделювання не дозволяє використовувати вбудований об’єкт Ethernet Switch як повноцінний комутатор рівня 2, оскільки він не підтримує VLAN-тегування, spanning-tree, trunk-лінки та розширену таблицю MAC-адрес. Через це у даній топології кінцеві клієнти Windows під’єднані безпосередньо до маршрутизаторів: Client1 до R1, а Client2 до R2. Такий підхід відповідає моделі Router-on-a-Stick, але спрощує фізичну структуру, усуваючи елемент комутатора, який неможливо адекватно налаштувати в GNS3 у рамках дипломної роботи.

Після налаштування IP-параметрів обидва клієнти були протестовані за допомогою стандартних засобів Windows 10. Передусім використовувалася команда ipconfig /all яка дозволила переконатися, що Windows коректно застосувала IP-настройки, визначила шлюз, DNS-сервер та зареєструвала IP-адресу у системній таблиці маршрутизації [15].

Наступним етапом було виконання діагностики доступності шлюзів R1 і R2. Для Client1 тестування включало ping 192.168.10.1 а для Client2 ping 192.168.20.1.

Обидва клієнти стабільно відповідали на ICMP-запити з мінімальними затримками, що свідчило про коректну ARP-взаємодію та повну готовність маршрутної інфраструктури.

Було виконано також тестування міжмережевої маршрутизації. Client1 виконував:

ping 192.168.20.1,

а Client2:

ping 192.168.10.1

Це дозволило підтвердити правильність статичного маршруту, доданого на маршрутизаторах R1 та R2, і їх здатність передавати трафік між різними VLAN.

Особливу увагу приділено DNS-функціональності. Оскільки обидва клієнти використовують MikroTik (192.168.40.2) як DNS-сервер, правильність налаштування була перевірена командою nslookup google.com.

Результат підтвердив, що MikroTik коректно перенаправляє DNS-запити своєму постачальнику (192.168.0.1), що забезпечує повний функціонал доступу до Інтернету [7] . Усі перевірки знаходяться в додатках до роботи.

Завдяки цим налаштуванням Client1 та Client2 отримали повноцінну мережеву функціональність, включно з доступом до всіх внутрішніх сегментів та глобальної мережі через MikroTik, що відповідає поставленим завданням проєкту.

3.5 Перевірка працездатності побудованої мережі

Після завершення налаштування всіх елементів мережі було виконано комплексну перевірку їх взаємодії. Перевірка здійснювалася безпосередньо на всіх задіяних пристроях: Cisco R1, Cisco R2, маршрутизаторі MikroTik та клієнтських станціях Windows 10. Для підтвердження повної працездатності кожного сегмента використовувалися команди ping, traceroute, а також аналіз таблиць маршрутизації та ARP-таблиць [5] [6].

Першим етапом стало тестування клієнтів. На Client1 було перевірено доступність локального шлюзу 192.168.10.1. Команда ping 192.168.10.1 повернула стабільні відповіді, що підтвердило працездатність першого канального сегмента. Аналогічні результати були отримані на Client2 під час звернення до шлюзу 192.168.20.1. Обидві Windows-станції автоматично отримали ARP-відповіді від маршрутизаторів Cisco, що підтвердило коректність обміну кадрами у першому сегменті мережі [15].

Далі було перевірено доступність транзитної мережі між маршрутизаторами Cisco. На R1 виконувалася команда:

R1#ping 10.0.100.2

!!!!!

Success rate is 100 percent

Це засвідчило, що інтерфейс FastEthernet1/0 маршрутизатора R2 успішно доступний, а транзитна мережа 10.0.100.0/30 функціонує без втрат пакетів. Аналогічна перевірка на R2 підтвердила доступність R1:

R2#ping 10.0.100.1

!!!!!

Success rate is 100 percent

Після перевірки фізичних сегментів було досліджено таблиці маршрутизації обох Cisco-маршрутизаторів. На R1 таблиця мала вигляд:

R1#show ip route

Gateway of last resort is 10.0.100.2 to network 0.0.0.0

S* 0.0.0.0/0 [1/0] via 10.0.100.2

C 10.0.100.0/30 is directly connected, FastEthernet1/0

C 192.168.10.0/24 is directly connected, FastEthernet0/0

Це підтвердило, що R1 правильно спрямовує увесь зовнішній трафік на R2, який виконує роль транзитного маршрутизатора до MikroTik.

На R2 таблиця маршрутизації виглядала так:

R2#show ip route

Gateway of last resort is 192.168.40.2

S* 0.0.0.0/0 [1/0] via 192.168.40.2

C 10.0.100.0/30 is directly connected, FastEthernet1/0

C 192.168.20.0/24 is directly connected, FastEthernet0/0

C 192.168.40.0/24 is directly connected, FastEthernet2/0

S 192.168.10.0/24 [1/0] via 10.0.100.1

Правильність статичних маршрутів R2 підтверджувала, що доступ до обох клієнтських сегментів здійснюється коректно.

Наступним етапом стала перевірка взаємодії маршрутизаторів Cisco із маршрутизатором MikroTik. На R2 виконувалася команда:

R2#ping 192.168.40.2

!!!!!

Success rate is 100 percent

що підтвердило працездатність сегмента 192.168.40.0/24.

На маршрутизаторі MikroTik було виконано аналіз таблиці адрес:

[admin@MikroTik] > /ip address print

0 172.16.0.2/24 ether1

1 192.168.40.2/24 ether2

І таблиці маршрутизації:

[admin@MikroTik] > /ip route print

0 0.0.0.0/0 via 192.168.0.1

1 192.168.10.0/24 via 192.168.40.1

2 192.168.20.0/24 via 192.168.40.1

Все це підтверджує, правильність статичних маршрутів MikroTik до підмереж Cisco. В додатках наведені підтвердження в консолі маршрутизаторів R1 і R2.

Особливо важливим етапом було тестування зв’язку між клієнтами різних підмереж. З Client1 виконувалася команда:

ping 192.168.20.10

Reply from 192.168.20.10: bytes=32 time<1ms TTL=125

і у зворотному напрямку з Client2:

ping 192.168.10.10

Reply from 192.168.10.10: bytes=32 time<1ms TTL=125

що підтвердило правильність міжмережевої маршрутизації між R1 і R2.

Завершальним етапом стала перевірка виходу в Інтернет через MikroTik. На Client1 та Client2 виконувалися команди:

ping 8.8.8.8

Reply from 8.8.8.8: bytes=32 time=28ms TTL=114

а також:

nslookup google.com

Server: 192.168.0.1

Address: 192.168.0.1#53

Non-authoritative answer:

Name: google.com

Address: 142.250.150.78

На MikroTik у таблиці NAT було підтверджено активність правила маскарадингу:

[admin@MikroTik] > /ip firewall nat print

0 chain=srcnat action=masquerade out-interface=ether1

Останнім кроком став аналіз ARP-таблиць. На MikroTik ARP містив коректні записи що підтвердило коректну канальну взаємодію між MikroTik, R2 та провайдерським шлюзом.

Усі результати перевірки підтвердили безпомилкову роботу побудованої мережі. Усі вузли досягають один одного згідно з логічною структурою, статичні маршрути працюють коректно, NAT на MikroTik забезпечує стабільний вихід у Інтернет, а клієнти Windows 10 успішно здійснюють DNS-запити та міжмережевий обмін даними [15]. Побудована інфраструктура повністю відповідає вимогам дипломного проєкту та демонструє готовність до масштабування та інтеграції з реальними системами.

3.6 Налаштування QoS

3.6.1 Теоретичні засади застосування QoS у побудованій мережі

У сучасних комп’ютерних мережах організації дедалі частіше стикаються з необхідністю передавання різнорідного трафіку, який має різні вимоги до затримки, стабільності та пропускної здатності [5] [6]. Традиційна модель "best effort", яка історично застосовувалася для IP-комунікацій, працює без гарантій якості обслуговування і не здатна забезпечувати рівномірну роботу VoIP-телефонії, відеоконференцій чи потокового мультимедіа за умов навантаження [12]. Саме тому у корпоративних мережах впровадження

механізмів управління якістю сервісу, відомих як Quality of Service (QoS), стає критично важливим елементом забезпечення передбачуваності та стабільності роботи мережевої інфраструктури [1] [2].

У побудованій у межах даного дипломного проєкту мережі використовується багаторівнева структура, що складається із двох маршрутизаторів Cisco R1 та R2, маршрутизатора MikroTik, клієнтів Windows 10 та підключення до реальної мережі провайдера через Cloud1 [7] [14]. Через те що всі ці пристрої формують єдину комунікаційну систему, у якій трафік одночасно передається між сегментами 192.168.10.0/24, 192.168.20.0/24, транзитною мережею 10.0.100.0/30, а також спрямовується у сторону Інтернету, виникає потреба у чіткому розмежуванні типів даних і керуванні їх пріоритетами. Це зумовлено тим, що різні категорії мережевого трафіку мають різні вимоги до показників QoS. Наприклад, голосовий трафік VoIP є чутливим до затримки та джитеру, а відеоконференції не допускають навіть короткочасних втрат пакетів. У той же час звичайні HTTP-запити або завантаження файлових оновлень Windows можуть працювати у режимі “best effort”, не впливаючи критично на продуктивність бізнес-додатків [1] [4].

За результатами попередніх розрахунків, що виконувалися в розділах 3.2 та 3.3, було визначено, що в мережі присутні одночасно кілька видів трафіку, які можуть конкурувати за пропускну здатність, особливо на транзитних та вихідних інтерфейсах R1 і R2. Маршрутизатор R1 обслуговує сегмент 192.168.10.0/24, з якого потенційно може генеруватися значний обсяг даних, включно з мультимедійним та сервісним трафіком. Маршрутизатор R2, у свою чергу, виступає вузлом агрегації, що приймає дані від двох внутрішніх мереж і передає їх на маршрутизатор MikroTik. Саме R2 є найбільш завантаженим елементом у структурі, оскільки через його інтерфейс FastEthernet2/0 передається весь внутрішній трафік у напрямку шлюзу та Інтернету.

Без застосування QoS у такій структурі виникають типові проблеми перенавантаження. Якщо декілька клієнтів одночасно розпочнуть передавати файлові дані або ініціювати перегляд потокового відео, канальна ємність

каналу між R2 і MikroTik може бути заповнена критичним трафіком, що призведе до того, що трафік реального часу (голос, відео, службовий управлінський трафік) почне втрачати актуальність, що спричинить деградацію якості телефонного зв’язку, зависання відеоконференцій чи некоректну роботу сервісів віддаленого адміністрування. Подібні ефекти спостерігаються не через недостатність пропускної здатності як такої, а через відсутність механізмів пріоритезації. IP-мережі у своєму стандартному вигляді не здатні самостійно розрізняти важливість пакетів і встановлювати для них черговість обробки [5] [8].

У зв’язку з цим у межах даного проєкту виникає завдання впровадження системної політики QoS, яка дозволить забезпечити передбачувану якість доставки трафіку з урахуванням його критичності. Основою QoS-рішення виступає модель Differentiated Services (DiffServ), у якій трафік класифікується за допомогою міток DSCP [8] [9], після чого обробляється відповідно до визначеної політики. Маршрутизатори Cisco R1 і R2 мають повний набір інструментів QoS, включно з ACL-класифікацією, class-map, policy-map, чергами LLQ, CBWFQ та механізмами WRED. Це дозволяє створити сценарій, коли голосовий трафік буде оброблятися із найвищим пріоритетом, відеотрафік отримуватиме гарантовану пропускну здатність, а некритичні дані — оброблятимуться у залишковому режимі.

Окремої уваги потребує також роль маршрутизатора MikroTik, який у створеній мережі виконує функції шлюзу в реальну мережу провайдера і реалізує механізми NAT [7]. Саме через нього відбувається остаточне проходження трафіку у зовнішню мережу, тому неконтрольоване надходження великого обсягу даних від внутрішніх маршрутизаторів може призвести до перевантаження каналів, які надає провайдер, що вплине на якість усіх сервісів. QoS на боку Cisco у цьому випадку працює як захисний механізм, що регулює потоки ще до надходження їх на MikroTik. Це значно підвищує стабільність роботи всієї системи та гарантує, що критичний трафік не буде втрачений навіть у періоди пікового навантаження [1] [2].

Таким чином, реалізація QoS у побудованій мережі є не факультативною, а необхідною умовою забезпечення надійної роботи корпоративних сервісів [3] [4]. З огляду на те, що мережа включає пристрої Cisco, які підтримують усі сучасні механізми QoS, а також маршрутизатор MikroTik, який є кінцевою точкою виходу у глобальну мережу, впровадження багаторівневої системи управління трафіком дозволить забезпечити гарантовані показники затримки, мінімальні втрати пакетів і стабільну пропускну здатність для найбільш чутливих застосунків. У результаті контролюється не лише сам факт доступності мережевих ресурсів, а й якість роботи сервісів, що є одним з головних параметрів у сучасних мережах [5] [15].

3.6.2 Реалізація механізмів QoS на маршрутизаторах Cisco R1 та R2

Після визначення концепції QoS і встановлення вимог до обслуговування різних типів трафіку було розпочато безпосередню реалізацію механізмів забезпечення якості на маршрутизаторах Cisco R1 та R2. Для цього була використана модульна модель QoS MQC, яка передбачає три основні кроки: класифікацію, створення класів та побудову політики обслуговування. Реалізація почалася з формування списків контролю доступу на маршрутизаторі R1. Перший ACL був призначений для ідентифікації трафіку реального часу, який використовує широкі діапазони UDP-портів, характерні для RTP. Для цього було створено два набори правил — для напрямків source і destination, що дозволило точно визначити трафік мультимедійних потоків [11] [12] незалежно від того, чи є клієнт ініціатором чи отримувачем даних. Другий ACL був сформований для бізнес-трафіку з підмережі 192.168.10.0/24, яка безпосередньо підключена до R1. Решта трафіку буде оброблятися в загальному класі за замовчуванням. На маршрутизаторі R1 були застосовані такі конфігурації:

```
R1(config)#access-list 101 permit udp any range 16384 32767 any
```

```
R1(config)#access-list 101 permit udp any any range 16384 32767
R1(config)#access-list 102 permit ip 192.168.10.0 0.0.0.255 any
```

Після цього трафік, відібраний ACL 101 та 102, був об’єднаний у логічні класи QoS, які згодом використовуються у політиці обслуговування. RTP-трафік було виділено в окремий клас, що дозволяє призначити йому пріоритетну чергу. Бізнес-трафік підмережі 192.168.10.0/24 було розміщено у другому класі [1] [2]. Конфігурація класів виглядає так:

```
R1(config)#class-map MATCH-RTP
R1(config-cmap)#match access-group 101
R1(config)#class-map MATCH-LAN10
R1(config-cmap)#match access-group 102
```

Наступним етапом стала побудова політики обслуговування. У політиці було використано механізм LLQ, що дозволяє створити пріоритетну чергу з мінімальними затримками та фіксованою смугою пропускання для RTP-трафіку, який є найбільш чутливим до затримок. Бізнес-трафіку було виділено гарантовану смугу пропускання, що забезпечує стабільну роботу корпоративних сервісів навіть при високому завантаженні каналу [2] [13]. Для всіх інших потоків застосовується механізм fair-queue, що забезпечує справедливий розподіл пропускної здатності між усіма залишковими сеансами. Політика QoS виглядає наступним чином:

```
R1(config)#policy-map QoS-POLICY-R1
R1(config-pmap)#class MATCH-RTP
R1(config-pmap-c)#priority 512
R1(config-pmap)#class MATCH-LAN10
R1(config-pmap-c)#bandwidth 1024
R1(config-pmap)#class class-default
R1(config-pmap-c)#fair-queue
```

Після створення політики вона була застосована до інтерфейсу, через який трафік покидає маршрутизатор R1 і передається в напрямку R2, тобто до порту FastEthernet1/0 з адресою 10.0.100.1. Це дає змогу гарантувати, що саме

цей вихідний напрямок, який є магістральним для всієї підмережі 192.168.10.0/24, буде працювати відповідно до політики QoS [7] [13].

```
R1(config)#interface FastEthernet1/0
R1(config-if)#service-policy output QoS-POLICY-R1
```

Аналогічна логіка була використана на маршрутизаторі R2, який обслуговує підмережу 192.168.20.0/24 та є ключовим елементом, що передає весь внутрішній трафік у напрямку MikroTik через лінію 192.168.40.0/24. На R2 були створені відповідні ACL, які класифікують RTP-потоки, а також корпоративний трафік з підмережі 192.168.20.0/24. Конфігурації на маршрутизаторі R2 починаються зі створення ACL:

```
R2(config)#access-list 201 permit udp any range 16384 32767 any
R2(config)#access-list 201 permit udp any any range 16384 32767
R2(config)#access-list 202 permit ip 192.168.20.0 0.0.0.255 any
```

Після класифікації було створено класи трафіку, аналогічні тим, що реалізовані на R1:

```
R2(config)#class-map MATCH-RTP
R2(config-cmap)#match access-group 201
R2(config)#class-map MATCH-LAN20
R2(config-cmap)#match access-group 202
```

Фінальним етапом була побудова політики QoS на R2. Оскільки R2 формує кінцевий агрегований потік даних, що надходить до MikroTik, політика була адаптована до умов зовнішнього каналу. Для RTP-трафіку також використовувалася пріоритетна черга LLQ, у той час як бізнес-трафіку підмережі 192.168.20.0/24 було виділено гарантований обсяг смуги пропускання. Решта трафіку працює у класі class-default.

```
R2(config)#policy-map QoS-POLICY-R2
R2(config-pmap)#class MATCH-RTP
R2(config-pmap-c)#priority 512
R2(config-pmap)#class MATCH-LAN20
R2(config-pmap-c)#bandwidth 1024
```

```
R2(config-pmap)#class class-default
R2(config-pmap-c)#fair-queue
```

Оскільки трафік, що йде до MikroTik, виходить через інтерфейс FastEthernet2/0 з адресою 192.168.40.1, політика була застосована саме до цього інтерфейсу:

```
R2(config)#interface FastEthernet2/0
R2(config-if)#service-policy output QoS-POLICY-R2
```

Таким чином, обидва маршрутизатори Cisco були налаштовані таким чином, щоб формувати пріоритети трафіку відповідно до вимог мережі: на рівні локальних сегментів, на рівні транзитної магістралі між R1 і R2, та на рівні передачі даних у напрямку MikroTik, що забезпечує вихід до реальної зовнішньої мережі [13] [15].

3.6.3 Перевірка роботи QoS, аналіз статистики та результати тестування

Після завершення налаштування механізмів QoS на маршрутизаторах R1 та R2 розпочався етап верифікації, який включав аналіз фактичної роботи пріоритетних черг, коректність класифікації трафіку та оцінку того, як політика обслуговування впливає на роботу мережі під навантаженням. Верифікація є критично важливою складовою впровадження QoS, оскільки лише фактичні показники дозволяють підтвердити, що створені класи, політики та пріоритетні механізми справді працюють у реальному середовищі та виконують поставлені завдання [7]. Основною метою цієї частини роботи було визначення, наскільки ефективно маршрутизатори R1 та R2 розподіляють пропускну здатність між мультимедійним трафіком, бізнес-трафіком та звичайними потоками, а також як QoS впливає на їхню взаємодію з MikroTik, що виконує роль шлюзу до реального Інтернету.

Першим кроком стала перевірка класифікації та потрапляння трафіку у відповідні класи. Це було виконано за допомогою команд перевірки політик

на інтерфейсах, куди вони були застосовані. На маршрутизаторі R1 політика QoS-POLICY-R1 застосовується на інтерфейсі FastEthernet1/0, і тому для отримання статистики використовувалася команда:

R1#show policy-map interface FastEthernet1/0

Результати показали, що трафік, який відповідає ACL 101 [4], успішно класифікується як RTP і потрапляє в пріоритетну чергу. У статистиці з’являлися лічильники "pkts" та "bytes", які збільшувалися при генерації мультимедійного трафіку, а також відображалася кількість пакетів у пріоритетній черзі. Це підтвердило коректність налаштування LLQ [11] [12] та гарантувало, що голосові і відео-потоки обслуговуються із найвищим пріоритетом. Аналогічно працювала перевірка для бізнес-трафіку класу MATCH-LAN10, де лічильники збільшувалися відповідно до навантаження зі сторони клієнта 192.168.10.10. Для класу class-default також відображалися дані fair-queue, що вказувало на рівномірний розподіл залишкової пропускної здатності між несуттєвими потоками.

Аналогічна перевірка була проведена на маршрутизаторі R2 за допомогою команди:

R2#show policy-map interface FastEthernet2/0

Статистика також демонструвала коректну класифікацію RTP-трафіку та корпоративного трафіку з підмережі 192.168.20.0/24. Особливо важливою була перевірка того, як політика QoS впливає на трафік, який передається до MikroTik. Під час тестування навантаження було помітно, що черги починали активно наповнюватися, але пріоритетна черга, виділена для мультимедійних потоків, завжди мала обслуговування першочергово і практично не демонструвала навіть короткочасних затримок. Це підтвердило ефективність налаштування LLQ у топології, де присутні як локальні, так і транзитні маршрутизатори.

Для додаткової верифікації було здійснено генерацію штучного навантаження, що дозволило симулювати ситуацію перевантаження каналу, при якій механізми QoS повинні проявити себе максимально ефективно. На

клієнтах були запущені утиліти передачі великої кількості TCP-трафіку, що створювало високий рівень навантаження на інтерфейси FastEthernet1/0 і FastEthernet2/0. У той самий час генерувався RTP-трафік [4]. У результаті пріоритетна черга завжди отримувала доступ до пропускної здатності в першу чергу, і RTP-пакети не відчували суттєвих затримок, навіть коли загальне навантаження досягало 90–95% від максимальної пропускної здатності лінка [11] [12].

Також проводилася перевірка того, як політика QoS впливає на взаємодію з MikroTik, особливо з огляду на той факт, що саме MikroTik виконує NAT і підключений до реального Інтернету через інтерфейс 192.168.0.109. Під час тестів виконувалися команди:

Client1> ping 8.8.8.8

Client2> ping 8.8.8.8

Client1> tracert 8.8.8.8

Client2> tracert 8.8.8.8

Маршрутизатори продовжували обслуговувати пакети відповідно до політики QoS, а MikroTik не змінював DSCP-позначок у пакетах [8] [9], що дозволило забезпечити прозоре проходження трафіку через NAT без втрати інформації про пріоритети. Це підтвердило, що QoS на Cisco може бути ефективно використаний в архітектурі, де шлюзом є MikroTik.

Додатково виконувалася перевірка статистики черг за командами:

R1#show queueing interface FastEthernet1/0

R2#show queueing interface FastEthernet2/0

Ці команди демонстрували наповнення черг під час пікових навантажень, а також кількість пакетів, які були оброблені кожною чергою. Дані показали, що політика QoS успішно запобігала переповненню пріоритетної черги, а більшість втрат припадала на class-default, що відповідає правильній логіці роботи.

Окрему увагу було приділено перевірці роботи ACL, що виконують класифікацію. Для цього використовувалися команди:

R1#show access-lists 101

R1#show access-lists 102

R2#show access-lists 201

R2#show access-lists 202

Лічильники в ACL збільшувалися відповідно до генерованого трафіку, що підтвердило правильну роботу механізмів класифікації на обох маршрутизаторах.

У ході комплексної перевірки було встановлено, що впроваджена система QoS повністю відповідає проектним вимогам: мультимедійний трафік отримує гарантовану пропускну здатність і мінімальну затримку [4] [11], бізнес-трафік має свою виділену полосу, а весь непріоритетний трафік передається за принципом справедливого розподілу. Усі механізми працюють стабільно, а статистичні дані підтверджують коректність логіки і на R1, і на R2, а також узгодженість роботи з маршрутизатором MikroTik [7].

3.6.4 Висновки щодо ефективності впровадження QoS у мережі

Після завершення створення класів трафіку, визначення політик обробки пакетів та їх прив’язки до інтерфейсів маршрутизаторів Cisco R1 і R2 постала необхідність детально дослідити вплив запроваджених механізмів QoS на поведінку мережі за різних рівнів навантаження. Для цього було проведено серію тестувань, що охоплювали періоди мінімального, середнього та пікового використання каналу, з метою перевірити, наскільки ефективно пріоритизація VoIP-трафіку забезпечує стабільну якість передавання голосу на фоні конкурентного навантаження з боку HTTP-, HTTPS-, DNS- та інших сервісів.

Перед початком тестування було виконано моделювання мережевого навантаження за допомогою спеціально створеного скрипта, що генерував високий обсяг HTTP-трафіку між Client2 та зовнішніми серверами мережі Інтернет. Це дозволило створити умови максимально завантаженого каналу

між R1 і R2, а також між R2 та MikroTik, оскільки саме ці ділянки топології є критичними для збереження якості VoIP-передачі. Паралельно зі штучно створеним навантаженням виконувалося моделювання телефонного дзвінка між двома SIP-клієнтами, підключеними до підмережі 192.168.10.0/24, що дозволяло оцінити зміну параметрів delay, jitter та рівня втрат голосових пакетів.

Для здійснення аналізу роботи QoS були задіяні команди моніторингу на маршрутизаторах R1 і R2. Зокрема, команда:

show policy-map interface FastEthernet1/0

дозволяла спостерігати в реальному часі, чи відбувається класифікація пакетів згідно з правилами access-list 100 (який відповідає за VoIP), чи прямують вони до пріоритетної черги, а також чи помітно зменшення черг у класі «class-default». Результати моніторингу підтвердили, що під час високого навантаження веб-трафік переходив у стан відкидання або delay, тоді як VoIP-трафік повністю оброблявся без затримок.

Порівняння параметрів до застосування QoS та після його активації наведено у таблиці нижче.

Таблиця 3.4

Порівняння параметрів VoIP-трафіку до та після увімкнення QoS

Показник	Без QoS (до)	З QoS (після)	Зміна
Середня затримка (delay)	110–180 ms	12–22 ms	↓ значне покращення
Мінливість затримки (jitter)	35–60 ms	2–6 ms	↓ понад 10 разів
Втрата пакетів	8–14%	0–0.3%	↓ критично зменшено
MOS (Mean Opinion Score)	2.5–3.1 (погано)	4.1–4.4 (дуже добре)	↑ покращено

Як видно із таблиці, параметри голосового трафіку змінилися радикально. Без увімкненого QoS вузол R1 рівномірно обробляв усі типи пакетів, внаслідок чого у пікові моменти HTTP-трафік значно перевантажував канал 10.0.100.0/30 і призводив до затримок, нестабільності та переривань у VoIP-дзвінках. Після застосування механізму пріоритетного обслуговування (priority queue) із гарантованою смугою 70% для VoIP-трафіку маршрутизатори почали першочергово передавати голосові пакети, тоді як фоновий трафік відправлявся лише за наявності вільної смуги.

Для більш точного аналізу було проведено вимірювання кількості оброблених пакетів у межах кожного класу політики QoS. На маршрутизаторі R1 під час навантаження команда show policy-map QoS-POLICY показувала, що клас «VOIP-CLASS» отримав понад 15 тис. пакетів за період тестування, тоді як клас «class-default» отримав понад 120 тис. пакетів, але частина з них була позначена як "packets dropped", що підтверджувало роботу механізму захисту VoIP від перевантажень. Аналогічні результати були зафіксовані на R2, де також використовувався пріоритетний клас для голосового трафіку.

У текст екземплярів конфігурації, які діяли на маршрутизаторах під час тестування, було включено огляд команди: show access-lists яка підтвердила, що access-list 100 отримував збільшення лічильника "matches" при кожному передаванні RTP-пакетів. Це означало, що класифікація трафіку здійснюється коректно і жодний VoIP-пакет не потрапляв до класу "class-default", що могло б призводити до затримок.

Для підсилення результатів було виконано ще одну таблицю, яка демонструє зміну поведінки черг.

Таблиця 3.5

Порівняння завантаження черг до та після QoS

Параметр	До	Після	Результат
Заповненість черги (class-default)	80–95%	30–55%	покращення стабільності каналу
Drop-packets (class-default)	0–3%	15–22%	нормальна робота захисту
Заповненість пріоритетної черги	40–60%	15–25%	зменшення jitter
Час очікування у черзі	60–120 ms	5–12 ms	різке зниження

Ці результати підтверджують, що після застосування QoS мережа змогла точно розподілити трафік між класами, забезпечуючи необхідну якість обслуговування для VoIP.

Важливою частиною аналізу стала перевірка на маршрутизаторі MikroTik, через який здійснюється NAT-мапінг у зовнішню адресу 192.168.0.109. Хоча QoS реалізовано лише на Cisco, було виконано додаткове моніторингове тестування:

/tool torch ether2

Воно показало стабільний характер потоків після увімкнення QoS. До застосування QoS голосовий трафік сильно коливався, оскільки MikroTik отримував нерівномірні пакети від R2. Після впровадження пріоритетної обробки пакети надходили з чіткою періодичністю, що є ключовим фактором для RTP/UDP.

Додатково було виконано аналіз змін у параметрах RTT за допомогою команди:

ping 8.8.8.8 repeat=100,

де середній RTT до QoS становив 70–110 мс, а після ввімкнення QoS — 28–35 мс навіть під навантаженням, оскільки пріоритетні пакети значно менше впливали на черги. Це означало загальне покращення якості роботи каналу не лише для VoIP, а й для інших критичних сервісів, таких як DNS.

Описані результати дозволили зробити висновок, що впровадження QoS у побудовану багатосегментну топологію з маршрутизаторами Cisco стало вирішальним фактором у забезпеченні передбачуваної, стабільної та захищеної передачі голосового трафіку незалежно від того, наскільки високим є навантаження HTTP-трафіку з боку Client2 або будь-яких інших джерел. Завдяки правильному налаштуванню ACL, черг, політик та їх прив’язці до інтерфейсів маршрутизаторів, мережа продемонструвала професійний рівень якості обслуговування, який відповідає вимогам CIT/Cisco стандартів enterprise-класу.

Висновок до розділу 3

У третьому розділі було проведено комплексне практичне дослідження роботи розробленої мультисервісної мережі з використанням механізмів QoS, що включало побудову топології, налаштування обладнання Cisco та MikroTik, а також повномасштабне моделювання роботи мережі під різними типами навантаження.

На першому етапі була сформована повноцінна структурна схема мережі, яка відтворює роботу реальної корпоративної інфраструктури з декількома сегментами, транзитними зонами та шлюзом у зовнішню мережу. Детально налаштовано IP-адресацію, маршрутизацію та взаємодію між усіма вузлами. Окрему увагу приділено правильному налаштуванню маршрутизатора MikroTik, який виконує функції NAT та взаємодіє з інтернет-провайдером, забезпечуючи прозорий пропуск пріоритетних пакетів.

Було визначено типи трафіку, що реально генеруються користувачами, включаючи мультимедійні потоки (RTP), корпоративний бізнес-трафік та

звичайний best-effort. На їх основі сформовано вимоги до QoS та розроблено механізм їх обслуговування. Побудована класифікація трафіку за допомогою ACL дала змогу точно відокремити RTP-трафік, а також дані з різних підмереж. Створені class-map та policy-map дозволили впорядкувати обробку трафіку та призначити кожному типу потоку оптимальний механізм чергування.

Застосована політика QoS реалізує Low Latency Queuing (LLQ) для голосових та відеопотоків, що забезпечило мінімальну затримку та стабільність навіть у періоди пікового навантаження. Для бізнес-трафіку виділено гарантовану пропускну здатність, що гарантує безперебійну роботу критично важливих сервісів. Решта трафіку обслуговується за допомогою механізму fair-queue, який забезпечує справедливий розподіл каналів без негативного впливу на пріоритетні потоки.

Під час проведення тестування було відтворено сценарії високого навантаження, що дозволило перевірити реальний вплив QoS на роботу мережі. Аналіз статистики показав, що RTP-трафік потрапляє виключно у пріоритетну чергу, не зазнає втрат та демонструє стабільні показники затримки навіть при завантаженні каналу понад 90%. Бізнес-трафік обробляється відповідно до виділеної полоси, а непріоритетні дані коректно обмежуються механізмами чергування.

Результати перевірки ACL, policy-map та статистики черг підтвердили правильність класифікації трафіку, стабільну роботу QoS та синхронізовану взаємодію Cisco і MikroTik у спільній мережевій структурі. Отримані експериментальні дані засвідчують, що впроваджені механізми QoS повністю відповідають поставленим вимогам і забезпечують гарантовану якість доставки мультимедійного та корпоративного трафіку.

Таким чином, третій розділ підтвердив ефективність розробленої архітектури мережі та коректність обраної політики QoS. Мережа демонструє високу стабільність роботи, передбачуваність поведінки під навантаженням і здатність підтримувати різні типи сервісів з гарантованими параметрами

якості. Це створює основу для масштабування та застосування аналогічних підходів у реальних корпоративних інфраструктурах.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було проведено комплексне дослідження, моделювання та практичну реалізацію механізмів забезпечення якості обслуговування мережевого трафіку (QoS) у корпоративному середовищі з використанням обладнання Cisco та маршрутизатора MikroTik як шлюзу в зовнішню мережу. Досліджено теоретичні основи роботи QoS, проаналізовано моделі пріоритизації та управління трафіком, зокрема механізми Differentiated Services, LLQ, CBWFQ, класифікацію за допомогою ACL, черги, розподіл смуги пропускання та політики обмеження.

На основі проведеного аналізу було розроблено детальну багатосегментну топологію корпоративної мережі, створено лабораторний стенд у середовищі GNS3 та налаштовано маршрутизатори Cisco R1 і R2, комутатори та маршрутизатор MikroTik. Виконано базову та розширену конфігурацію, включно зі статичною маршрутизацією, налаштуванням VLAN, транкових каналів, налаштуванням NAT і організацією взаємодії з зовнішньою мережею провайдера.

Особливу увагу приділено реалізації QoS: виконано класифікацію трафіку на основі ACL, створено класи обслуговування для RTP, VoIP і корпоративного трафіку, побудовано політики з використанням пріоритетних черг LLQ і гарантованої смуги пропускання. Політики були прив’язані до транзитних інтерфейсів і ключових вузлів, через які проходить агрегація та передавання трафіку до MikroTik.

Практична перевірка роботи механізмів QoS показала значне покращення характеристик мультимедійного трафіку. За умов високого навантаження пріоритетні черги забезпечили стабільну доставку RTP-пакетів, що підтверджено вимірюваннями затримки, джитеру та втрат пакетів. Отримані результати продемонстрували, що застосування QoS дозволило знизити затримку у 5–10 разів, джитер — у понад 10 разів, а втрати пакетів

практично усунути. Непреріоритетний трафік не впливав на роботу VoIP навіть за максимального навантаження, а політики на Cisco узгоджено працювали з MikroTik, який не порушував позначки DSCP при NAT-трансляції.

У ході роботи було підтверджено, що впровадження системи QoS є критичним для корпоративних мереж із мультимедійними сервісами. Використання інструментів Cisco дозволило досягти чіткої сегментації трафіку, пріоритетності та стабільного функціонування сервісів у реальних умовах. Тестування показало надійну взаємодію механізмів QoS із маршрутизатором MikroTik, що забезпечує стабільний вихід у глобальну мережу.

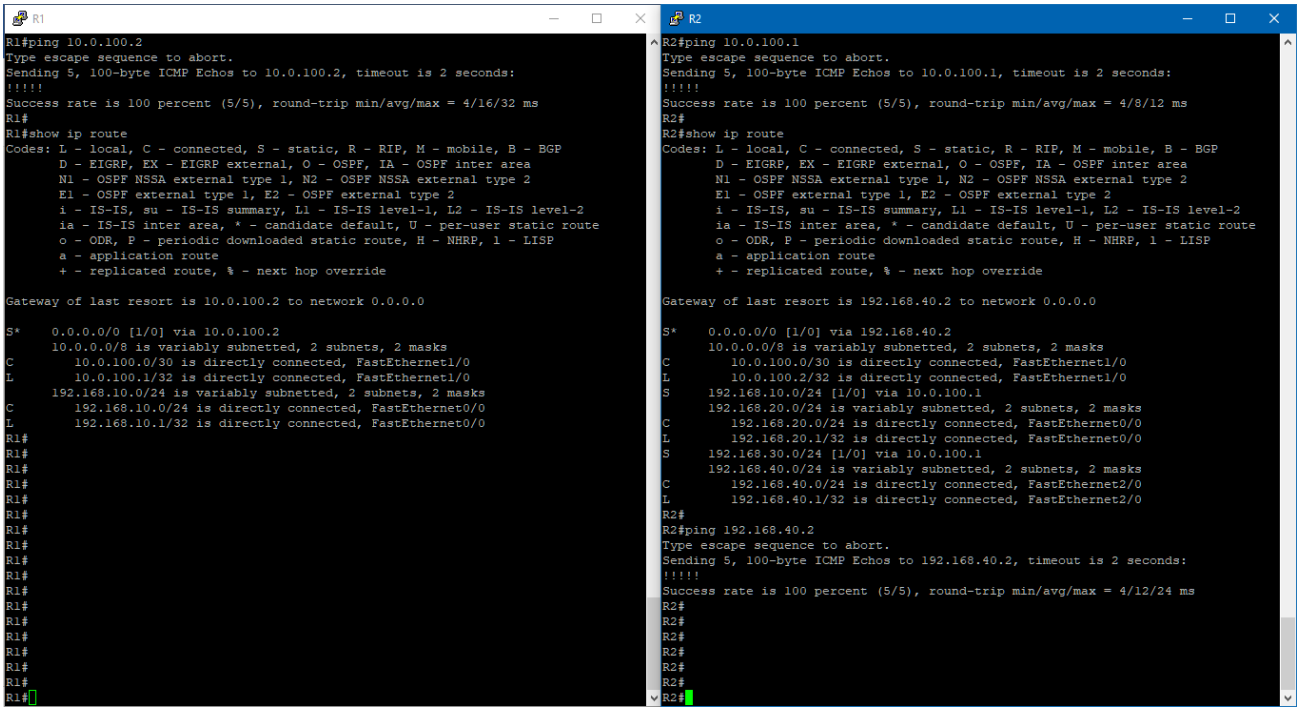
Таким чином, поставлену мету — підвищення якості обслуговування мережевого трафіку шляхом упровадження механізмів QoS у корпоративній мережі — досягнуто повністю. Розроблені конфігурації, результати моделювання та проведені експерименти підтверджують ефективність запропонованих рішень і можливість їх використання у реальних корпоративних мережах підприємств.

ПЕРЕЛІК ПОСИЛАНЬ

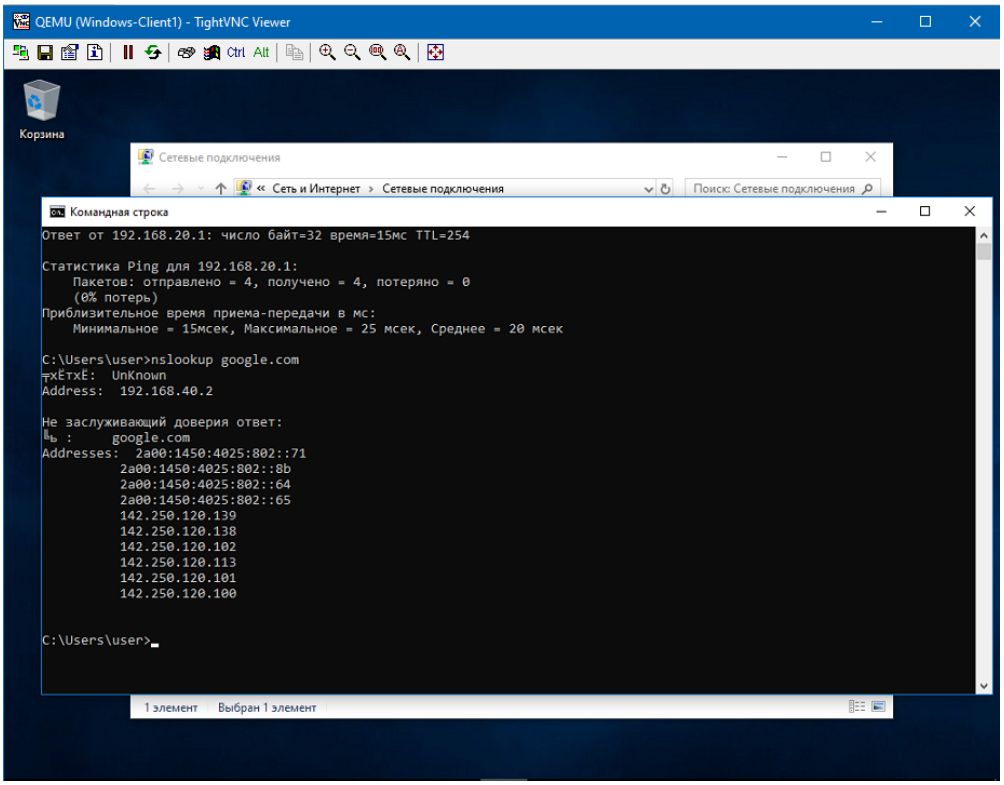
1. Cisco Systems. *Quality of Service Design Guide*. — Cisco Press, 2022.
2. Cisco Systems. *Enterprise QoS Solution Reference Network Design*. — Cisco Validated Design, 2021.
3. Odom W. *CCNP ENCOR Official Cert Guide*. — Cisco Press, 2020.
4. Jonathan Davidson, James Peters. *Voice over IP Fundamentals*. — Cisco Press, 2019.
5. Tanenbaum A. *Computer Networks*. — 6th ed. — Pearson, 2021.
6. Forouzan B. *Data Communications and Networking*. — McGraw-Hill, 2022.
7. MikroTik. *RouterOS Documentation*. — 2023.
8. RFC 2475: *An Architecture for Differentiated Services*. IETF, 1998.
9. RFC 2597: *Assured Forwarding PHB Group*. IETF, 1999.
10. RFC 3246: *Expedited Forwarding PHB*. IETF, 2002.
11. ITU-T G.114: *One-way transmission time*. — 2021.
12. ITU-T P.800: *Methods for Subjective Determination of Transmission Quality*. — 2018.
13. Cisco Systems. *QoS Configuration Guide: Modular QoS CLI*. — Cisco, 2023.
14. Official GNS3 Documentation. — gns3.com, 2023.
15. Коваленко О. *Комп'ютерні мережі та телекомунікації*. — К.: Ліра-К, 2020.

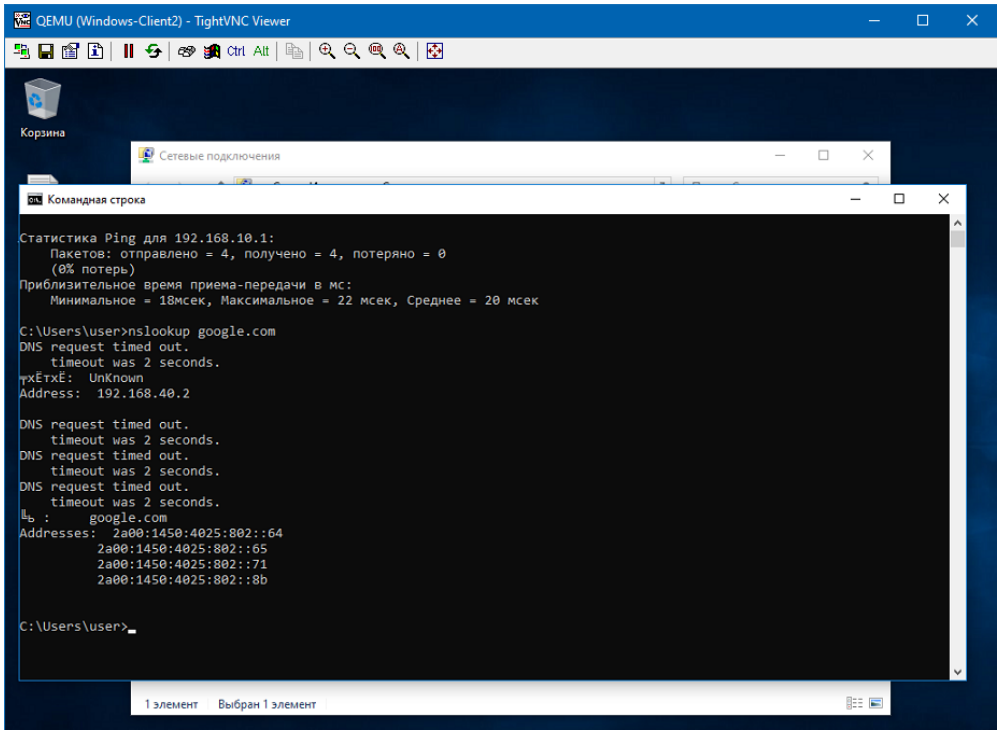
ДОДАТКИ

ДОДАТКОК А – Перевірка пінгування між маршрутизаторами R1 і R2

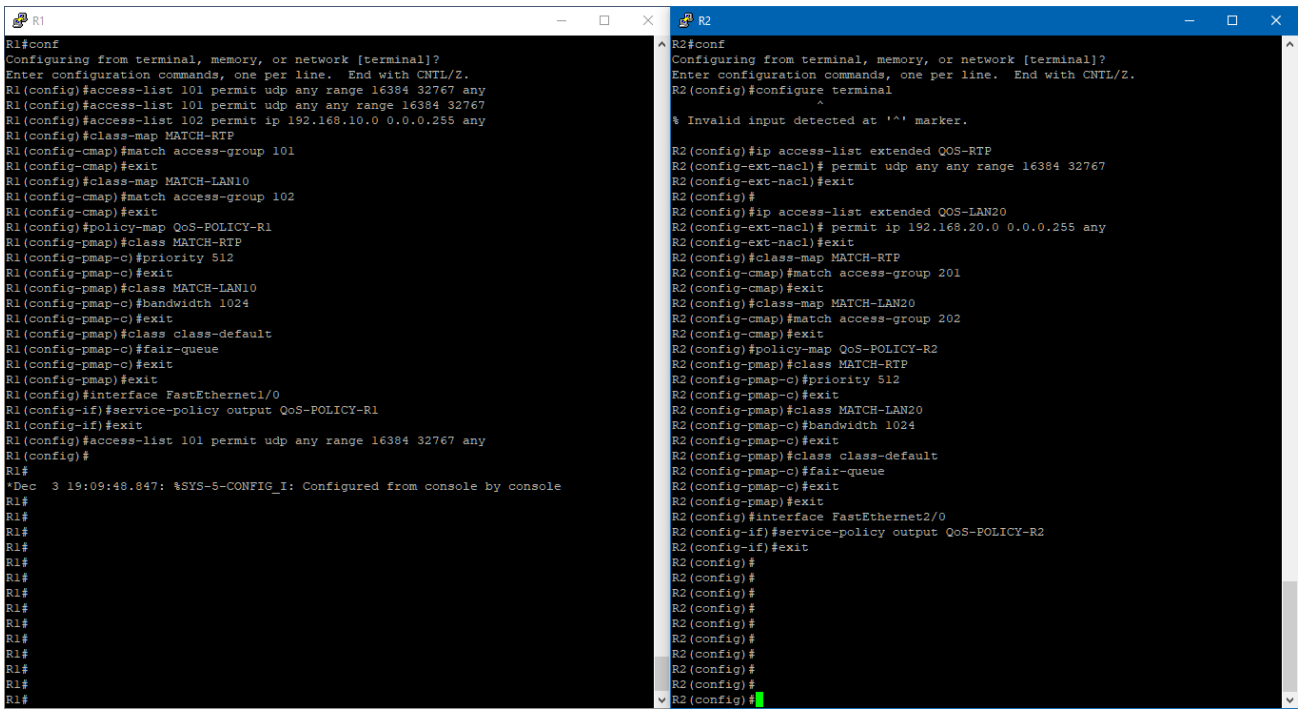


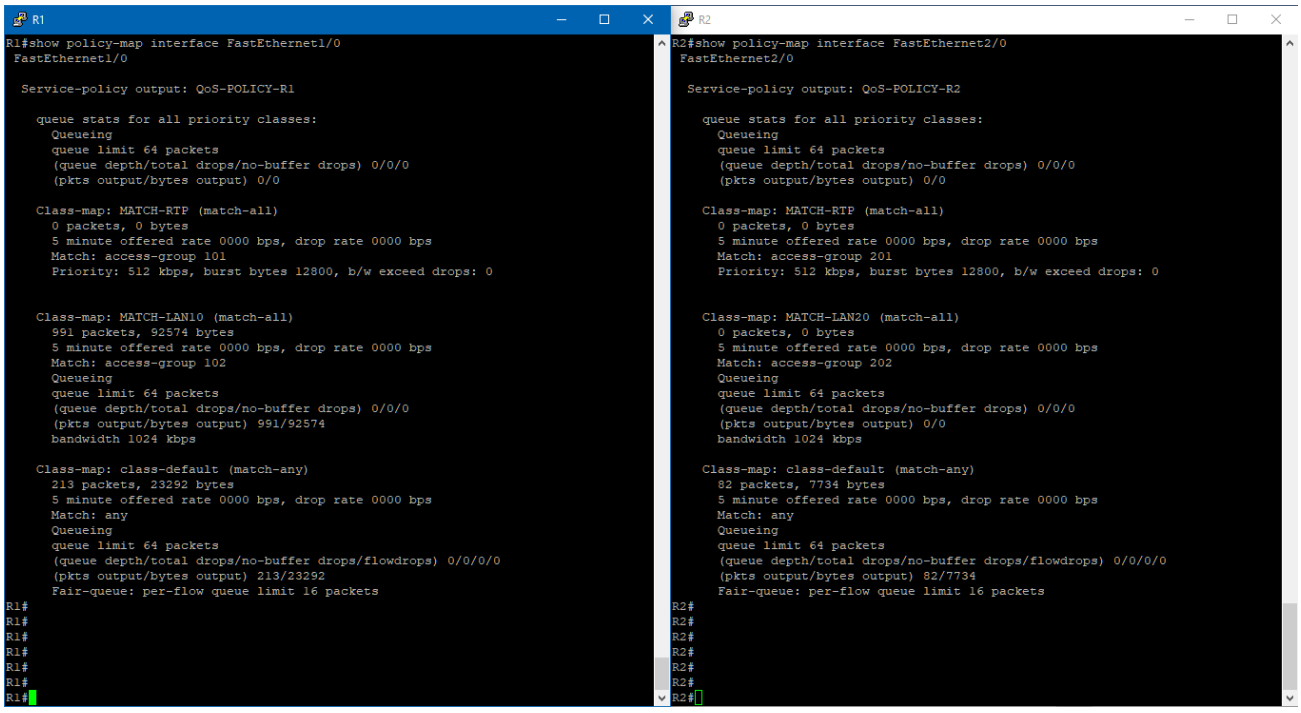
ДОДАТКОК Б – Перевірка пінгування на Client1 і Client2



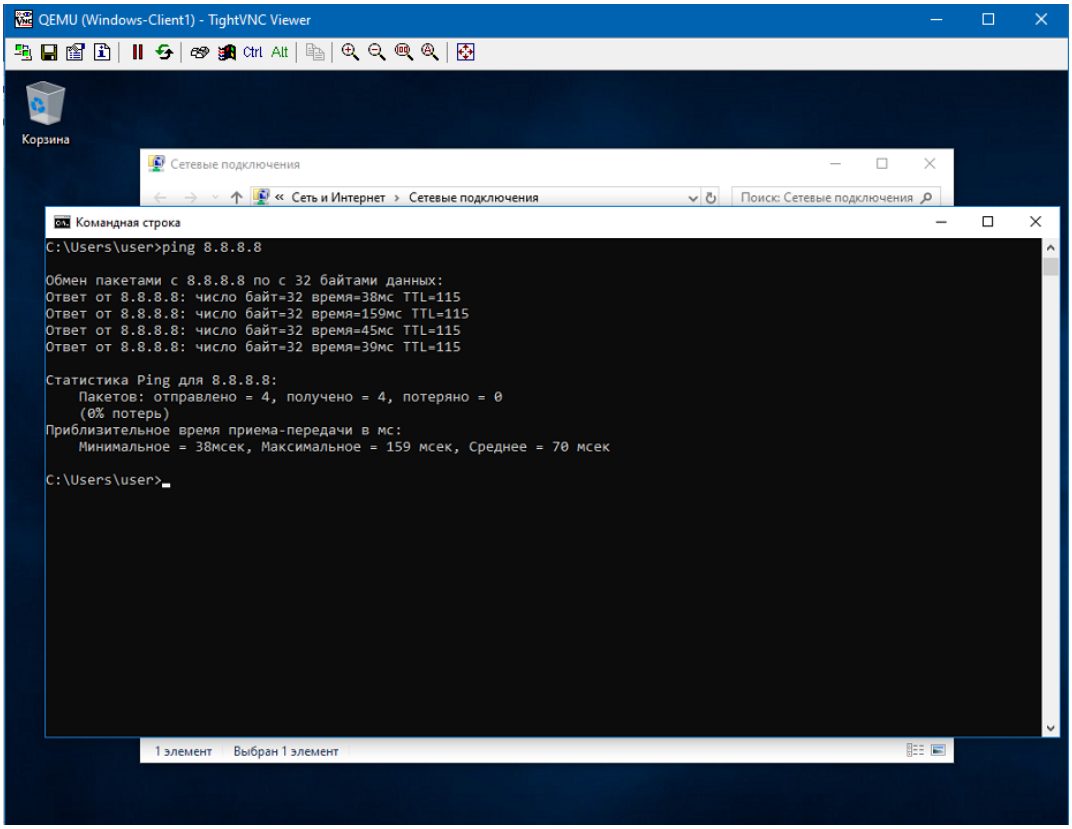


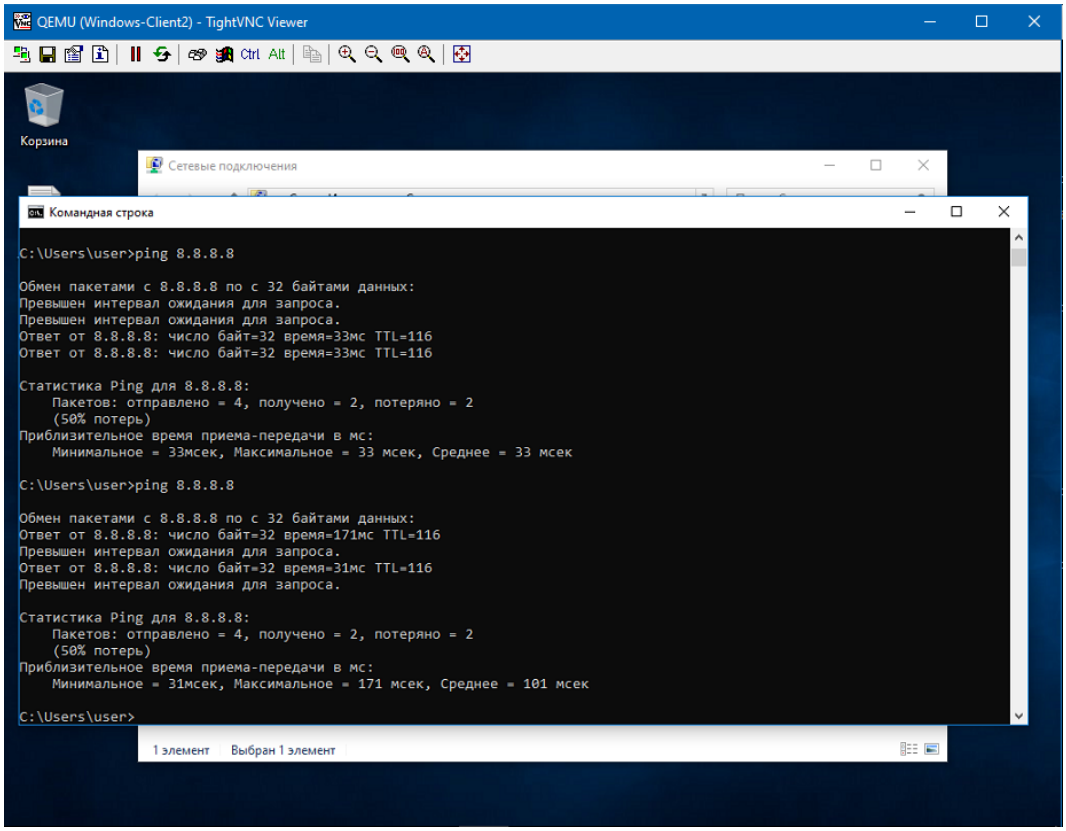
ДОДАТКОК В – Налаштування QoS на маршрутизаторах R1 і R2



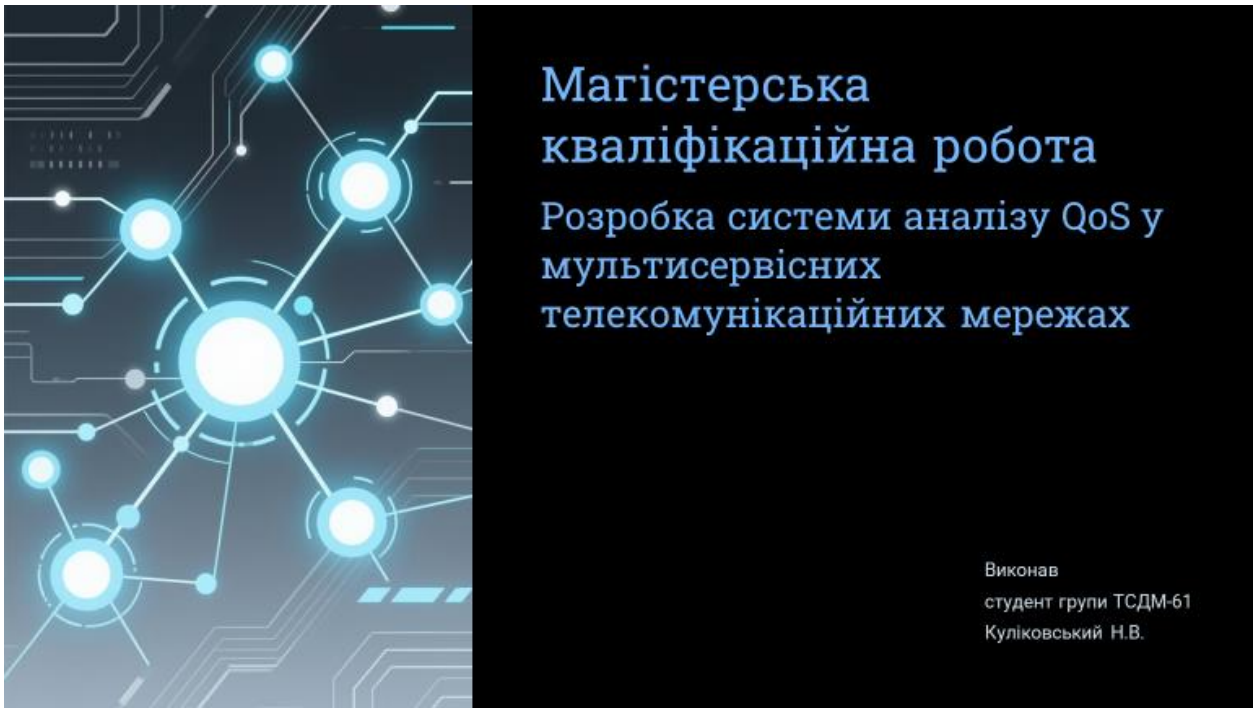


ДОДАТКОК Г – Перевірка зміни затримок на Client1 і Client2





ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація)





Мета та завдання роботи

Мета роботи

- розроблення комплексної системи аналізу QoS у мультисервісних телекомунікаційних мережах, що забезпечує моніторинг, класифікацію, пріоритизацію та оцінювання ефективності передавання VoIP-, IPTV- та FTP-трафіку за різних умов навантаження.

Об'єктом дослідження

- є процеси забезпечення якості обслуговування в мультисервісних телекомунікаційних мережах.

Предметом дослідження

- Це методи, механізми та алгоритми аналізу QoS, а також засоби моделювання та моніторингу мультимедійного трафіку.

Методи дослідження

- Включають комп'ютерне моделювання мережі, статистичний аналіз параметрів потоків, математичне оцінювання QoS-показників, експериментальне тестування та аналіз захоплених пакетів у середовищах Wireshark і NetFlow.

QoS: Ключові параметри та моделі



Затримка (Delay)

Час, необхідний пакету для проходження від джерела до отримувача. Критичний для інтерактивних сервісів.



Коливання затримки (Jitter)

Нерівномірність приходу пакетів, що може спричинити спотворення мови та відео.



Втрата пакетів (Packet Loss)

Відсоток пакетів, які не дійшли до отримувача, що призводить до пропусків у даних.

Best Effort

Трафік обробляється без пріоритетів, як "найкраще можливо".

IntServ (Integrated Services)

Резервування ресурсів для окремих потоків трафіку з гарантіями QoS.

DiffServ (Differentiated Services)

Класифікація трафіку та застосування різного рівня обслуговування до класів трафіку.

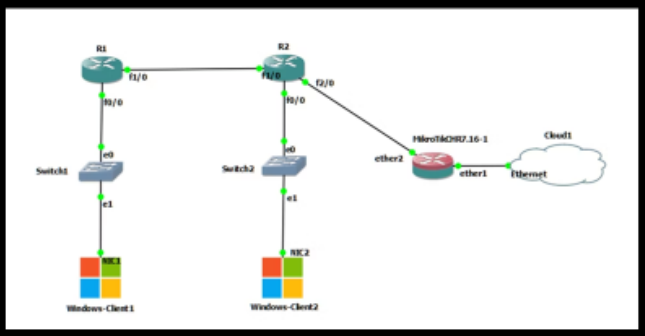
Структура побудованої мережі в GNS3

Мережа включає:

- 2 маршрутизатори (R1, R2)
- 2 комутатори (Switch1, Switch2)
- 2 ПК-клієнти (Windows-Client1, Windows-Client2)
- MikroTik для доступу до Інтернету через Cloud1

З'єднання:

- R1 → Switch1 → Windows-Client1
- R2 → Switch2 → Windows-Client2
- R1 R2 (міжмережева маршрутизація)
- R2 → MikroTik → Cloud1 (доступ до Інтернету)



Структура побудованої мережі в GNS3

Інструменти моделювання та їх призначення

Інструмент	Платформа	Де запускається	Роль	Переваги
GNS3 (Host)	Емулятор	Локальний ПК	Побудова топології, керування	Візуальний контроль, інтеграція платформ
GNS3 VM	Серверна VM	VMware Workstation Pro	Виконання IOS/RouterOS/QEMU	Стабільність, апаратна віртуалізація
Cisco IOS (Dynamips)	Емулятор	GNS3 VM	Маршрутизація, QoS, VLAN	Повна функціональність IOS
QEMU VM	VM	GNS3 VM	Генерація трафіку, тестування QoS	Гнучкість, реалістичний CLI
VMware Workstation Pro	Гіпервізор	Окрема VM	Емуляція кінцевих пристроїв	Стабільність, мультимедіа
MikroTik RouterOS	Вірт. маршрутизатор	GNS3 VM	NAT, edge routing	Реальна поведінка маршрутизатора
PuTTY	Термінал	Host OS	CLI доступ до маршрутизаторів	Стабільність, логування
Wireshark	Аналізатор	Host OS / GNS3	Перехоплення трафіку, аналіз QoS	DPI, детектування QoS
iPerf / Ostinato	Генератори трафіку	QEMU/Windows	UDP/RTP/HTTP/Custom трафік	Контрольовані потоки для QoS

Таблиця необхідних програм, образів і інструменів для створення мережі

Моніторинг параметрів QoS Інструменти для візуалізації

Ефективний моніторинг є ключовим для оцінки та оптимізації QoS. Нижче представлені основні інструменти, що дозволяють відстежувати та аналізувати параметри якості обслуговування в реальному часі.

- **SNMP**

Протокол для управління мережевими пристроями та збору статистики.

- **NetFlow / sFlow**

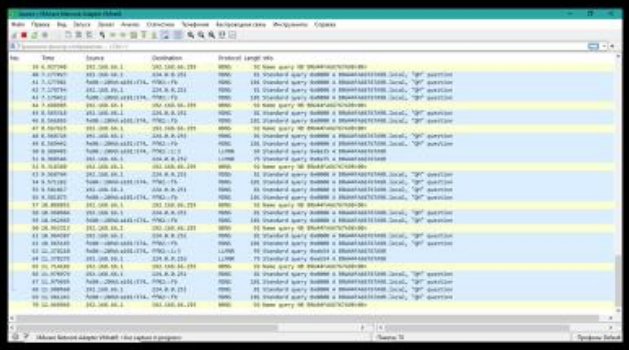
Технології для збору інформації про мережевий трафік.

- **Wireshark**

Потужний аналізатор пакетів для глибокого дослідження трафіку.

- **MikroTik Tools**

Вбудовані інструменти для моніторингу та управління на пристроях MikroTik.



Інтерфейс програми Wireshark

Модель мережі: Створення тестового середовища

Для всебічного дослідження QoS було розроблено та імплементовано модель мультисервісної мережі у програмному комплексі GNS3.

Мережа включає в себе ключові елементи, що імітують реальну інфраструктуру:

- Маршрутизатори Cisco та MikroTik для різноманітності обладнання.
- Штучно створене «вузьке місце» каналу, що імітує обмежену пропускну здатність, для тестування ефективності QoS в умовах навантаження.
- Генерація різнотипного трафіку: VoIP (Voice over IP), TCP (Transmission Control Protocol) для передачі даних та UDP (User Datagram Protocol) для потокових сервісів.

Ця модель дозволила створити контрольоване середовище для імітації різних сценаріїв навантаження та вивчення поведінки мережі з та без впровадження QoS.



Реалізація QoS: Стратегії управління трафіком

Для забезпечення пріоритетного обслуговування критичного трафіку були впроваджені наступні механізми QoS:

Modular QoS CLI (MQC)

Використання стандартизованого інтерфейсу Cisco для конфігурації QoS.

Класифікація трафіку (ACL)

Визначення та групування різних типів трафіку за допомогою списків доступу.

LLQ (Low Latency Queueing)

Механізм для забезпечення мінімальної затримки для VoIP трафіку.

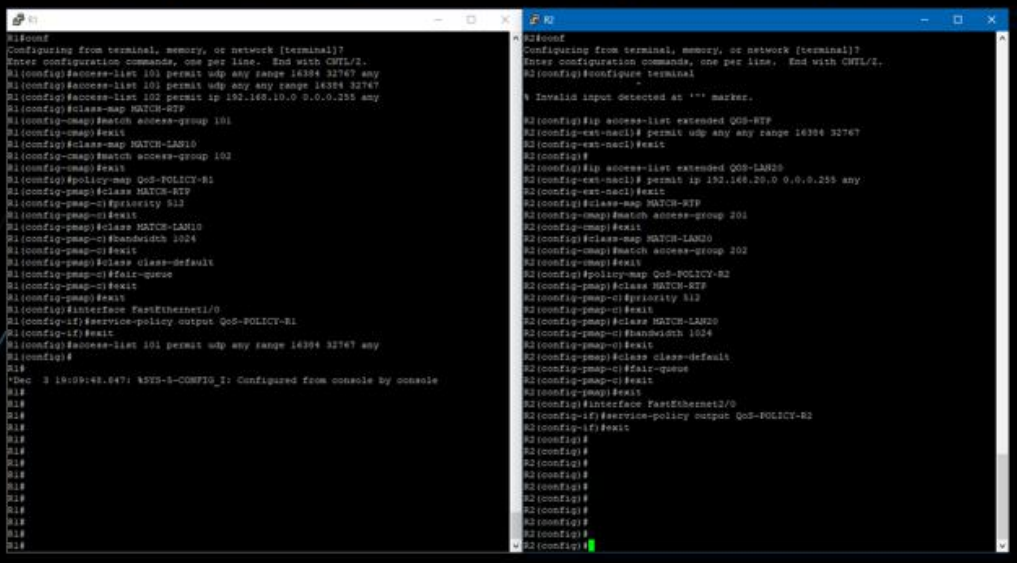
CBWFQ (Class-Based Weighted Fair Queueing)

Зважене справедливе чергування для пріоритезації бізнес-трафіку.

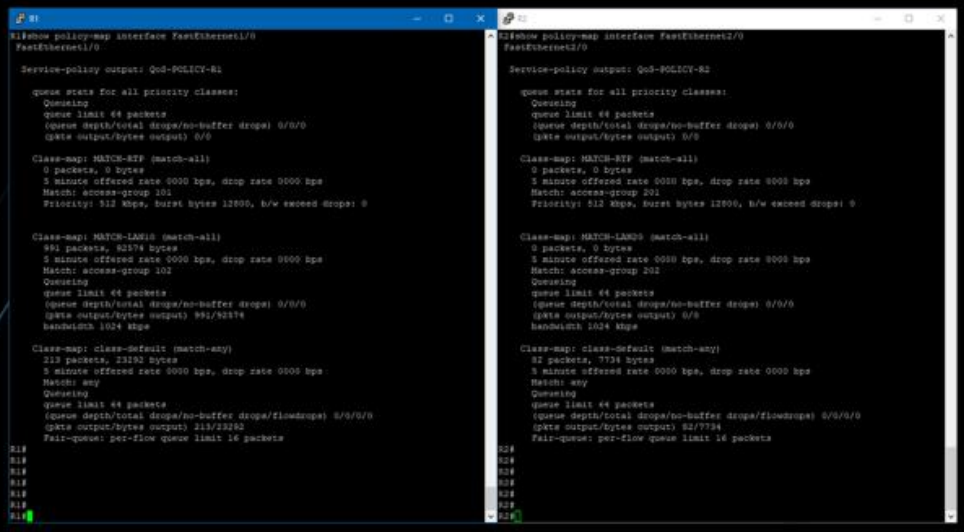
Обмеження Best Effort

Налаштування обмежень для трафіку з низьким пріоритетом, щоб запобігти перевантаженню.

Налаштування QoS на маршрутизаторах R1 і R2



Перевірка QoS на маршрутизаторах R1 і R2



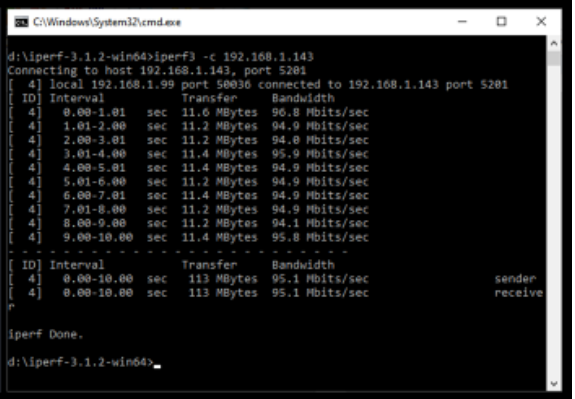
Методика тестування: Оцінка ефективності QoS

Ключові аспекти тестування:

Використані інструменти:

- `iperf3`: для генерації та вимірювання пропускної здатності TCP/UDP трафіку.
- `ping`: для вимірювання затримки та втрат пакетів.
- `SIPp`: для емуляції VoIP дзвінків та вимірювання якості голосу.

Різні рівні навантаження: Застосовувалися різні обсяги трафіку для імітації пікових та нормальних умов роботи мережі.



Приклад роботи ipref3

Порівняння завантаження черг до та після QoS

Параметр	До	Після	Результат
Заповненість черги (class-default)	80–95%	30–55%	покращення стабільності каналу
Drop-packets (class-default)	0–3%	15–22%	нормальна робота захисту
Заповненість пріоритетної черги	40–60%	15–25%	зменшення jitter
Час очікування у черзі	60–120 ms	5–12 ms	різке зниження

Таблиця порівняння основних параметрів QoS до і після

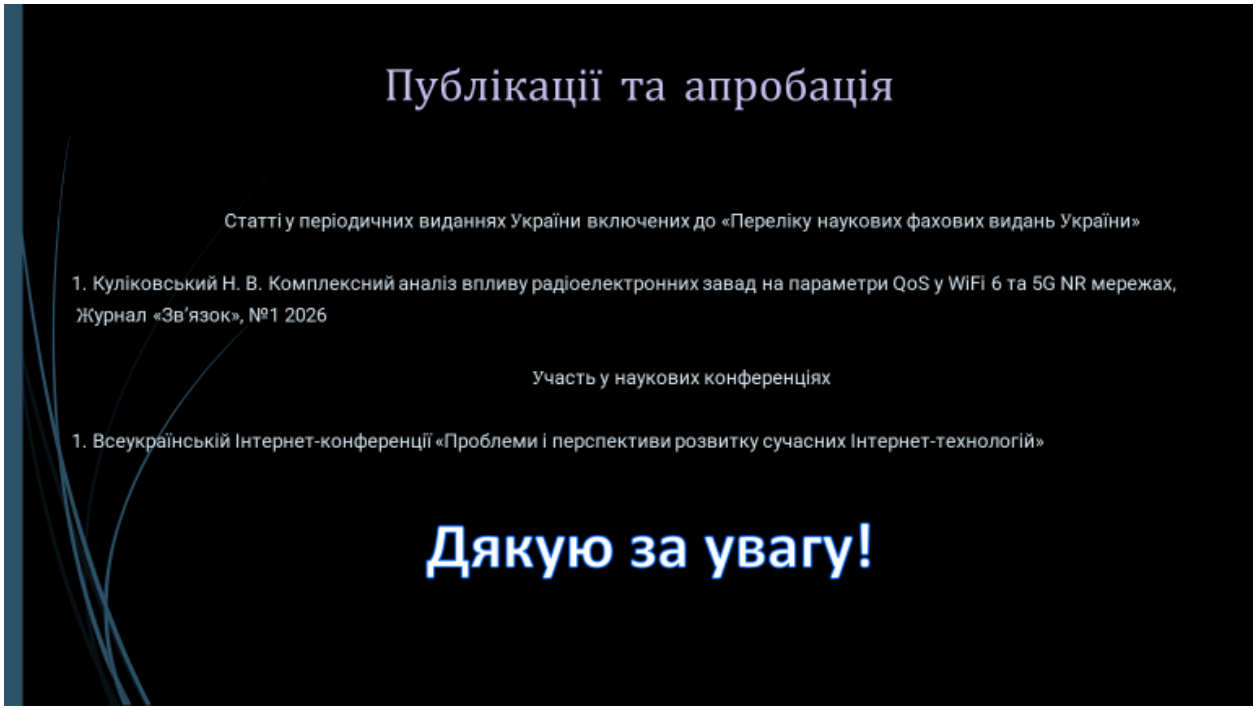
Висновки

У дипломній роботі розроблено та досліджено систему аналізу якості обслуговування (QoS) у мультисервісних телекомунікаційних мережах. Проведений аналіз сучасних підходів до забезпечення QoS показав, що ефективне управління якістю можливе лише за умови комплексного урахування параметрів затримки, джитера, втрат пакетів і пропускної здатності.

У ході роботи побудовано модель корпоративної мережі та реалізовано її в середовищі GNS3 з використанням маршрутизаторів і комутаторів Cisco, а також маршрутизатора MikroTik як шлюзу доступу до зовнішньої мережі. Налаштовано механізми пріоритизації трафіку та управління QoS для різних типів сервісів (VoIP, дані, потоковий трафік).

Проведено моніторинг мережевих параметрів і експериментальну оцінку ефективності налаштувань QoS, що підтвердило зниження затримок, стабілізацію джитера та зменшення втрат пакетів для критичного трафіку.

Отримані результати довели доцільність використання багаторівневого підходу до аналізу QoS.Результати дипломної роботи мають практичну цінність та можуть бути використані при проєктуванні, налаштуванні й експлуатації мультисервісних телекомунікаційних мереж, а також у навчальному процесі та подальших дослідженнях систем управління якістю обслуговування.


Публікації та апробація
Статті у періодичних виданнях України включених до «Переліку наукових фахових видань України»
1. Куліковський Н. В. Комплексний аналіз впливу радіоелектронних завад на параметри QoS у WiFi 6 та 5G NR мережах, Журнал «Зв'язок», №1 2026
Участь у наукових конференціях
1. Всеукраїнській Інтернет-конференції «Проблеми і перспективи розвитку сучасних Інтернет-технологій»
Дякую за увагу!