

Міністерство освіти і науки України  
Національний технічний університет  
«Дніпровська політехніка»

Навчально-науковий  
інститут електроенергетики  
(навчально-науковий інститут)  
Факультет інформаційних технологій  
(факультет)  
Кафедра інформаційних технологій та комп'ютерної інженерії  
(повна назва)

## ПОЯСНЮВАЛЬНА ЗАПИСКА кваліфікаційної роботи ступеня магістра

Здобувача вищої освіти \_\_\_\_\_ Пасічного Сергія Ігоровича \_\_\_\_\_  
(ПІБ)  
академічної групи \_\_\_\_\_ 123М-23-1 \_\_\_\_\_  
(шифр)  
спеціальності \_\_\_\_\_ 123 Комп'ютерна інженерія \_\_\_\_\_  
(код і назва спеціальності)  
за освітньо-професійною програмою \_\_\_\_\_ «Комп'ютерна інженерія» \_\_\_\_\_  
(офіційна назва)

на тему «Обґрунтування структури та параметрів мережевих пристроїв комп'ютерної системи компанії інтернет-продажів з використанням додатку внутрішнього моніторингу та локального вікі-сайту»

(назва за наказом ректора)

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи	доц. Шедловський І. А.			
розділів:				
синтез системи	доц. Ткаченко С.М.			
розроблення програмного забезпечення	ас. Бешта Л.В.			
Рецензент				
Нормоконтролер	проф. Цвіркун Л.І.			

Дніпро  
2024

**ЗАТВЕРДЖЕНО:**

завідувач кафедри  
інформаційних технологій  
та комп'ютерної інженерії  
(повна назва)

\_\_\_\_\_ В.В. Гнатушенко  
(підпис) (ініціали, прізвище)

« \_\_\_\_\_ » \_\_\_\_\_ 2024 року

**ЗАВДАННЯ**  
**на кваліфікаційну роботу**  
**ступеня магістра**  
(бакалавра, магістра)

здобувача вищої освіти \_\_\_\_\_ Пасічного С.І. академічної групи \_\_\_\_\_ 123М-23-1  
(прізвище та ініціали) (шифр)

спеціальності \_\_\_\_\_ 123 Комп'ютерна інженерія

за освітньою-професійною програмою \_\_\_\_\_ «Комп'ютерна інженерія»  
(офіційна назва)

на тему «Обґрунтування структури та параметрів мережевих пристроїв комп'ютерної системи компанії інтернет-продажів з використанням додатку внутрішнього моніторингу та локального вікі-сайту».

затверджену наказом ректора НТУ «Дніпровська політехніка» від 17 жовтня 2024 р. №1388-с

Розділ	Зміст	Термін виконання
Стан питання та постановка завдання	На основі матеріалів практик, інших науково-технічних джерел обґрунтувати необхідність досліджень комп'ютерної системи компанії інтернет-продажів	11.10.2024
Теоретичний	На основі науково-технічних джерел проаналізувати теоретичні методи дослідження комп'ютерних мереж	25.10.2024
Синтез системи	Розробка комп'ютерної системи	15.11.2024
Розроблення програмного забезпечення	Розробка програмного забезпечення додатку внутрішнього моніторингу та локального вікі-сайту	29.11.2024
Експериментальний розділ	Проведення і обробка результатів експериментів	06.12.2024

Завдання видано \_\_\_\_\_  
(підпис керівника)

Дата видачі 06 вересня 2024 р.

Дата подання до екзаменаційної комісії

Прийнято до виконання \_\_\_\_\_  
(підпис здобувача вищої освіти)

доц. І. А. Шедловський  
(ініціали, прізвище)

10.12.2024 р.

Пасічний С.І.  
(ініціали, прізвище)

## РЕФЕРАТ

Пояснювальна записка: 118 с., 34 рис., 5 табл., 2 дод., 7 джерел.

Об'єктом дослідження є комп'ютерна система компанії з інтернет-продажів, яка включає мережеві пристрої, систему моніторингу та локальний вікі-сайт для управління внутрішньою документацією та обробки інформації.

Метою роботи є розробка ефективної структури та визначення оптимальних параметрів мережевих пристроїв для підтримки комп'ютерної системи компанії з інтернет-продажів.

Методи дослідження включають метод Бузена для візуалізації даних та планування, метод Гауса для розрахунків розподілу ресурсів, та метод Монте-Карло для моделювання ризиків та циклів роботи системи.

Результати та їх новизна. Визначено оптимальні параметри мережевих пристроїв та нові підходи до моніторингу. Новизна результатів полягає у пристосуванні теоретичних методів до практичних потреб інтернет-продаж.

Взаємозв'язок з іншими роботами. Дана робота є продовженням бакалаврської роботи, у якій було досліджено основні принципи створення мережевих архітектур та оптимізації їх процесів.

Галузь застосування результатів роботи охоплює компанії, що займаються інтернет-продажами, а також інші організації, які потребують оптимізації роботи внутрішніх комп'ютерних систем і збереження даних в умовах високої конкуренції та технічних вимог.

Висновки з роботи підтверджують важливість правильно спроектованої інфраструктури для забезпечення надійності та безпеки роботи компанії. Використання оптимальних параметрів мережевих пристроїв, ефективне впровадження систем моніторингу і підтримка внутрішнього вікі-сайту дозволяють покращити ефективність обробки даних та взаємодії з клієнтами, що в кінцевому результаті забезпечує стабільну і успішну роботу компанії.

Прогнозні припущення. У майбутньому поліпшення мережевих технологій дозволить розширити функціонал систем моніторингу та впровадити нові сервіси для автоматизації внутрішніх процесів.

## ЗМІСТ

Перелік скорочень, умовних познач, одиниць і термінів.....	6
Вступ.....	7
1 Стан питання та постановка завдання.....	8
1.1 Стислі відомості про об'єкт впровадження.....	8
1.2 Стан питання.....	9
1.3 Аналіз сучасного стану мережевих інфраструктур компаній з інтернет-продажів .....	10
1.4 Проблеми та виклики у створенні мережевої інфраструктури компаній з інтернет-продажів .....	11
1.5 Вимоги до мережевої інфраструктури та системи моніторингу.....	13
1.6 Локальний вікі-сайт як інструмент управління знаннями.....	14
1.7 Постановка завдання.....	14
2 Теоретичний розділ .....	16
2.1 Розрахунок основних характеристик трафіку комп'ютерної мережі .....	16
2.1.1 Попередня оцінка пропускнуої здатності сегмента мережі .....	16
2.1.2 Імітаційна модель мережі в пакеті CiscoPacketTracer.....	18
2.2 Імітаційне моделювання комп'ютерних та статистичних систем.....	20
2.3 Підходи до моделювання комп'ютерних мереж .....	24
2.3.1 Математичне моделювання мережі як системи обслуговування великої кількості користувачів.....	25
3 Синтез комп'ютерної системи підприємства з інтернет продажів .....	41
3.1 Призначення, функції комп'ютерної системи .....	41
3.2 Структура комплексу технічних засобів комп'ютерної системи підприємства.....	44
3.3 Схема функціональної структури комп'ютерної системи.....	46
3.5 Особливості структурної схеми комп'ютерної системи.....	49
3.6 Характеристики використаних апаратних засобів .....	52
4 Розроблення програмного забезпечення.....	57
4.1 Призначення й сфера застосування програми .....	57
4.2 Обґрунтування технічних характеристик програми .....	58
4.3 Опис розробленої програми.....	59
4.3.1 Опис логічної структури програми.....	60

4.5 Виклик і завантаження .....	64
5 Експериментальний розділ .....	66
5.1 Розробка математичної моделі мережі як замкнутої системи масового обслуговування.....	66
5.2 Розрахунок параметрів мережі по її моделі .....	68
5.3 Корекція характеристик вузлів .....	75
Висновки .....	79
Перелік посилань.....	80
Додаток А.....	81
Додаток Б.....	98

## ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАК, ОДИНИЦЬ І ТЕРМІНІВ

MeMO – мережа масового обслуговування;

CDN – мережі доставки контенту;

VPN – віртуальна захищена мережа;

DDoS – напад на комп'ютерну систему з наміром зробити комп'ютерні ресурси недоступними користувачам;

IDS/IPS – система виявлення вторгнень;

PoE – технологія передачі електроенергії та даних за допомогою «звитої пари»;

CPU – центральний процесор;

HTML – мова розмітки гіпертексту;

CSS – спеціально мова стилю сторінок;

## ВСТУП

Сучасні компанії, що займаються інтернет-продажами, стикаються з постійною необхідністю вдосконалення своєї інфраструктури для підтримки високого рівня доступності та надійності. В умовах жорсткої конкуренції будь-які збої в роботі комп'ютерної системи можуть призвести до фінансових втрат і зниження рівня довіри з боку клієнтів. Стабільне функціонування мережі, надійний захист даних, а також ефективне управління внутрішніми інформаційними потоками є ключовими факторами для забезпечення успішної роботи компанії. З огляду на це, обґрунтування структури та параметрів мережевих пристроїв є важливим для оптимізації процесів, забезпечення безпеки та швидкості роботи внутрішніх сервісів.

Мета дослідження – розробити ефективну структуру та визначити оптимальні параметри мережевих пристроїв для підтримки комп'ютерної системи компанії з інтернет-продажів, що використовує систему внутрішнього моніторингу та локальний вікі-сайт для управління внутрішньою документацією та обробки інформації.

Об'єктом дослідження є комп'ютерна система компанії з інтернет-продажів, що включає мережеві пристрої, систему моніторингу та локальний вікі-сайт.

Предмет дослідження – структура та параметри мережевих пристроїв, методи моніторингу мережі та використання локального вікі-сайту для підтримки внутрішньої роботи компанії.

# 1 СТАН ПИТАННЯ ТА ПОСТАНОВКА ЗАВДАННЯ

## 1.1 Стислі відомості про об'єкт впровадження

Об'єкт впровадження для якого було впроваджено систему внутрішнього моніторингу та локальний вікі-сайт, є провідним постачальником товарів для сну. Вона спеціалізується на продажу широкого асортименту матраців, подушок, постільної білизни, аксесуарів для сну та інших товарів, що сприяють здоровому і комфортному сну.

Компанія вже має відомий бренд і велику базу задоволених клієнтів. Її цінності полягають у якості товарів, персоналізованому обслуговуванні та швидкій доставці. З метою розширення свого бізнесу і присутності в онлайн-сегменті, компанія вирішила розробити власний Інтернет-магазин, що дозволить їй залучати більше клієнтів та забезпечувати зручність покупки товарів для широкого кола споживачів.

Розроблена система внутрішнього моніторингу та локальний вікі-сайт включають в себе інтуїтивний інтерфейс. Компанія має можливість ефективно просувати свої товари, аналізувати покупців та впроваджувати персоналізовані маркетингові стратегії для поліпшення конверсії і збільшення продажів.

На рисунку 1.1 наведена структурна схема об'єкту впровадження

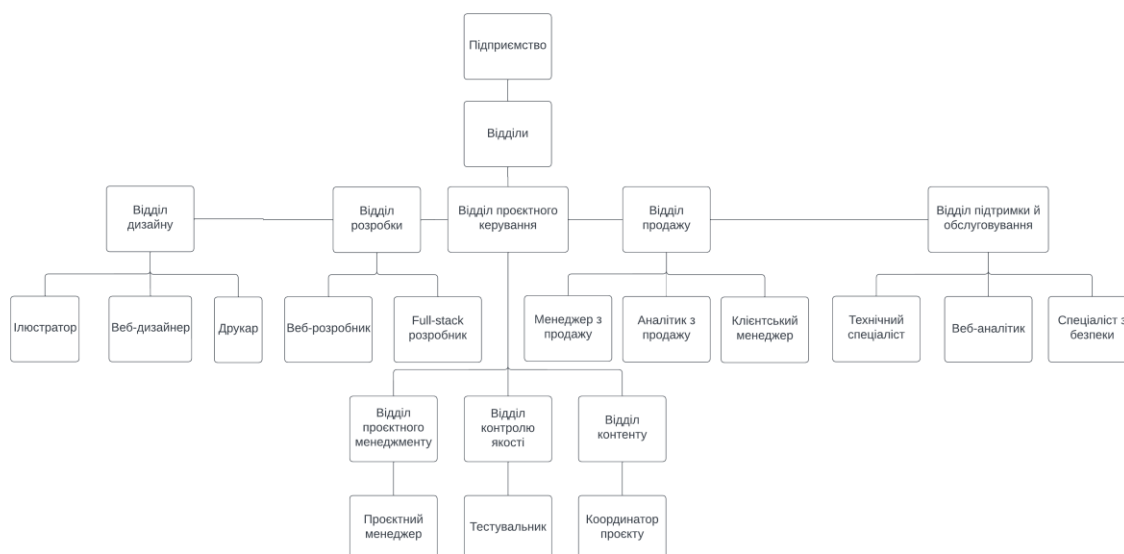


Рисунок 1.1 – Структурна схема об'єкту впровадження



## 1.2 Стан питання

В умовах стрімкого розвитку електронної комерції компанії, що працюють у сфері інтернет-продажів, стикаються з жорсткою конкуренцією та високими очікуваннями з боку споживачів. Для забезпечення ефективної роботи таких компаній необхідно мати надійну мережеву інфраструктуру, здатну забезпечувати високий рівень продуктивності, безпеки та доступності. Будь-які збої в роботі мережі можуть спричинити фінансові збитки, втрату довіри клієнтів та зниження конкурентоспроможності компанії.

Важливість дослідження структури та параметрів мережевих пристроїв полягає в тому, що оптимально налаштована мережа дозволяє ефективно обробляти та зберігати дані, забезпечувати швидке та безпечне передавання інформації між різними відділами компанії, а також своєчасно реагувати на будь-які збої чи загрози. Інтегровані рішення для моніторингу мережі дозволяють швидко виявити потенційні загрози та захистити важливі дані компанії.

Локальний вікі-сайт, який служить інструментом управління знаннями, також має велике значення для внутрішньої роботи компанії. Він дозволяє централізувати доступ до технічної документації, інструкцій з налаштування пристроїв, робочих протоколів та інших важливих матеріалів, знижуючи ймовірність помилок і підвищуючи загальну ефективність роботи співробітників.

Таким чином, дослідження, спрямоване на обґрунтування структури та параметрів мережевих пристроїв з використанням системи моніторингу та локального вікі-сайту, є важливим для розвитку інтернет-компаній, оскільки воно забезпечує основу для стабільного зростання бізнесу, підвищення якості обслуговування клієнтів та підвищення безпеки мережі.

### **1.3 Аналіз сучасного стану мережевих інфраструктур компаній з інтернет-продажів**

Сучасні інтернет-магазини функціонують у динамічному середовищі, де технології постійно розвиваються, а вимоги клієнтів стають вибагливішими. Саме тому мережева інфраструктура таких компаній є критично важливим компонентом, що визначає їхню ефективність та конкурентоспроможність.

Мережеві інфраструктури інтернет-компаній складаються з комплексу апаратного та програмного забезпечення, що дозволяє підтримувати стабільний зв'язок між користувачами, серверами та базами даних.

Сучасні тенденції в цій галузі свідчать про перехід від традиційних локальних серверів до хмарних рішень. Це дозволяє компаніям легко масштабувати свої ресурси залежно від навантаження, знижувати витрати на обслуговування та підвищувати гнучкість. Мікросервісна архітектура, яка розбиває великі системи на дрібніші, більш керовані модулі, також набирає популярності. Це спрощує розробку, тестування та масштабування додатків.

Особливий акцент робиться на архітектурних підходах до побудови мережі, таких як зіркоподібна, ієрархічна та комбінована структури. Окремо розглядається роль хмарних технологій, VPN та балансувальних пристроїв у забезпеченні продуктивності та надійності мережі.

Контейнеризація, за допомогою таких технологій як Docker, стала стандартом де-факто для пакетування та розгортання додатків. Вона забезпечує портативність та ізоляцію, що є особливо важливим у складних багатосервісних середовищах. Безсерверні обчислення дають змогу розробникам зосередитися на написанні коду, звільняючи їх від виконання рутинних завдань з управління інфраструктурою[1].

Мережі доставки контенту (CDN) є важливим елементом для забезпечення швидкого доступу до статичних ресурсів, таких як зображення та відео. Технологія Інтернету речей відкриває нові можливості для персоналізації та автоматизації в електронній комерції. До того ж, розвиток мереж 5G і

оптоволоконних технологій гарантує високу пропускну здатність і мінімальну затримку, що є критично важливим для стабільної роботи онлайн-магазинів.

Однак, разом з новими можливостями, з'являються і нові виклики. Кібербезпека стала одним з найактуальніших питань, оскільки хакерські атаки становлять серйозну загрозу для будь-якого бізнесу в Інтернеті. Захист даних клієнтів, їхньої особистої інформації та фінансових даних є пріоритетом для будь-якої компанії, що займається електронною комерцією.

Персоналізація також є важливим аспектом сучасного e-commerce. Компанії прагнуть надавати індивідуальні пропозиції кожному клієнту, що вимагає ефективної обробки великих обсягів даних. Штучний інтелект допомагає в цьому, аналізуючи поведінку користувачів та прогнозуючи їхні потреби.

Підсумовуючи, сучасні мережеві інфраструктури в e-commerce є динамічними системами, які постійно розвиваються. Щоб залишатися конкурентоспроможними, компанії повинні інвестувати в нові технології, підтримувати високий рівень безпеки та швидко адаптуватися до мінливих потреб клієнтів.

#### **1.4 Проблеми та виклики у створенні мережевої інфраструктури компаній з інтернет-продажів**

Створення надійної та ефективної мережевої інфраструктури для компанії, що займається інтернет-продажами, пов'язане з низкою серйозних викликів. Ці виклики зумовлені як швидким розвитком технологій, так і високими вимогами сучасних споживачів до швидкості, доступності та безпеки онлайн-сервісів.

Одним з головних викликів є забезпечення високої доступності сервісів. Оскільки будь-який збій в роботі сайту чи інтернет-магазину може призвести до втрати клієнтів та фінансових втрат. Для вирішення цієї проблеми необхідно продумувати резервування обладнання, використовувати хмарні технології та розробляти системи автоматичного відновлення після збоїв.

Високі вимоги до продуктивності є ще одним викликом. Сучасні користувачі очікують миттєвої реакції веб-сайтів і додатків. Для забезпечення високої швидкості роботи необхідно оптимізувати всі компоненти системи, від серверів до мережевого обладнання.

Інтернет-компанії стикаються з численними викликами у створенні та підтримці мережевої інфраструктури, оскільки обробка великих обсягів даних потребує високої пропускну здатності і надійного захисту. Розглянемо основні труднощі:

Високе навантаження на мережу в пікові періоди. Під час пікових періодів (наприклад, розпродажі, святкові знижки) обсяг трафіку різко зростає, що може призвести до перевантаження системи. У таких випадках важливо забезпечити безперебійну роботу, для чого необхідно застосовувати методи балансування навантаження, а також створювати резервні з'єднання для підтримки стабільності.

Захист від кіберзагроз і захист персональних даних. Мережеві інфраструктури інтернет-компаній є вразливими до кібератак, таких як DDoS-атаки, фішинг, крадіжка даних та інші види загроз. Високий рівень кіберзахисту включає наявність сучасних міжмережових екранів, засобів шифрування, систем виявлення вторгнень (IDS/IPS), що мінімізує ризик компрометації особистих даних клієнтів. Зростання кількості кібератак вимагає від компаній постійного вдосконалення заходів безпеки. Необхідно захищати як внутрішню мережу компанії, так і дані клієнтів, включаючи особисту інформацію та дані про платежі.

Необхідність масштабування мережі. У разі зростання бізнесу інфраструктура повинна легко адаптуватися до збільшення трафіку та кількості користувачів. Зростання обсягу продажів та кількості клієнтів вимагає відповідного масштабування інфраструктури. Це потребує попередньо спланованого розширення, зокрема, можливості нарощування потужностей серверів та модернізації мережових пристроїв.

Проблема збоїв та потреба в моніторингу мережі. Оскільки більшість операцій інтернет-компаній пов'язані з онлайн-замовленнями та обробкою

даних, будь-які збої можуть призвести до значних фінансових втрат. Тому важливу роль відіграє система моніторингу, яка дозволяє постійно відстежувати стан мережі, виявляти аномалії та реагувати на проблеми[2].

Складність управління великими мережевими інфраструктурами також є проблемою. Кількість пристроїв, програмного забезпечення та налаштувань може бути дуже великою, що ускладнює їх адміністрування та підтримку.

## **1.5 Вимоги до мережевої інфраструктури та системи моніторингу**

Мережа для інтернет-компаній повинна відповідати певним функціональним та технічним вимогам, що забезпечують високу якість обслуговування, захист та швидку роботу з великим обсягом даних.

Пропускна здатність. Мережа має забезпечувати високу швидкість обміну даними між серверами, внутрішніми компонентами системи та користувачами. Пропускна здатність є критичним параметром для уникнення затримок у доступі до сайту, особливо під час пікових періодів.

Надійність і безперервність роботи. Мережеве обладнання повинне підтримувати постійну доступність, забезпечуючи стабільну роботу онлайн-платформи цілодобово. Система повинна включати механізми відновлення після збоїв, зокрема резервування мережевих підключень, для запобігання тривалим перебоєм.

Безпека. Одним із головних аспектів є захист персональних даних клієнтів. Для цього необхідні потужні засоби кіберзахисту, зокрема захищені протоколи передачі даних, міжмережеві екрани, VPN, системи шифрування та багаторівневе аутентифікація.

Моніторинг. Система моніторингу повинна забезпечувати можливість контролю всіх критичних параметрів мережі в реальному часі, що дозволяє оперативно реагувати на будь-які збої або аномальні зміни трафіку. Моніторинг має включати інструменти для аналізу трафіку, виявлення загроз, визначення продуктивності пристроїв, моніторинг основних параметрів безпеки.

## **1.6 Локальний вікі-сайт як інструмент управління знаннями**

Локальний вікі-сайт є важливим елементом внутрішньої інфраструктури компанії, оскільки дозволяє зберігати та структурувати технічну документацію, інструкції з налаштування пристроїв та іншу важливу інформацію. У підрозділі розглядаються функції локального вікі-сайту, що підвищують ефективність роботи співробітників:

Централізований доступ до інформації. Всі співробітники компанії мають можливість доступу до технічної документації через єдину платформу. Це дозволяє зберігати документи, пов'язані з налаштуванням мережевих пристроїв, протоколами роботи, діагностикою та іншою інформацією, що сприяє стандартизації робочих процесів і знижує ризик виникнення помилок.

Зберігання та оновлення інструкцій. Локальний вікі-сайт надає можливість швидкого оновлення інструкцій та іншої інформації. Оскільки інфраструктура може змінюватися з часом, важливо, щоб всі зміни в документації були оперативним внесени до системи, що дозволяє уникнути застарілої інформації.

Доступ до процедур безпеки та реагування на збої. У вікі-сайті зберігаються процедури роботи з мережевими пристроями, протоколи безпеки, алгоритми реагування на збої та інші інструкції, які можуть допомогти співробітникам швидко відреагувати у разі виникнення критичних ситуацій.

## **1.7 Постановка завдання**

На основі проведеного аналізу сформульовано завдання цього дослідження, яке включає обґрунтування структури мережевих пристроїв і параметрів мережевої інфраструктури для інтернет-компанії з інтеграцією системи моніторингу та локального вікі-сайту. Поставлене завдання включає такі підзадачі:

Розробка та обґрунтування структури мережевих пристроїв. Визначення оптимальної структури мережевих пристроїв, що забезпечує високу

продуктивність, надійність та безперервність роботи для підтримки інтернет-продажів.

Параметризація мережевої інфраструктури. Вибір параметрів мережевих пристроїв, включаючи пропускну здатність, резервування, способи захисту та конфігурацію з'єднань, щоб забезпечити стабільну роботу в умовах високих навантажень.

Впровадження системи моніторингу. Інтеграція системи моніторингу, що дозволить у реальному часі спостерігати за станом мережевих пристроїв, попереджати про збої та швидко реагувати на інциденти.

Створення локального вікі-сайту. Розробка та налаштування локального вікі-сайту для управління внутрішніми знаннями, включаючи технічну документацію, протоколи безпеки, інструкції з обслуговування та діагностики мережевих пристроїв.

## 2 ТЕОРЕТИЧНИЙ РОЗДІЛ

### 2.1 Розрахунок основних характеристик трафіку комп'ютерної мережі

#### 2.1.1 Попередня оцінка пропускної здатності сегмента мережі

Для розрахунку інтенсивності трафіку підмережі відділу підтримки та обслуговування вона повинна бути навантажена на сто відсотків.

Для запобігання перевантаженню комутатора, швидкість надходження пакетів повинна бути меншою за швидкість їх обробки. Розрахунок передбачає, що всі користувачі користуються послугою одночасно. Середня щільність трафіку становить 550 кадрів в секунду, а середній розмір повідомлення дорівнює 1480 байт. Загальна кількість користувачів – 96.

Для розрахунку пропускної здатності мережі наступна формула застосовується на рівні доступу.

$$P_{p.p} = \mu * I * N * 8 = 550 * 1480 * 96 * 8 = 625 \text{ Мбіт/с} \quad (2.1)$$

Результати розрахунків показують, що вони відповідають параметрам мережі, таким чином, зазначене обладнання не буде перевантажено. Комутатори розподільчого рівня спрямовують трафік до маршрутизаторів через вихідні лінії на 1000 Мбіт/с.

$$\mu_{\text{вих}} = 1\,000\,000\,000 / (1480 \cdot 8) = 84\,460 \text{ пакетів/с} \quad (2.2)$$

Кожне джерело в середньому генерує 550 пакетів на секунду, що обмежує його можливість підключення до максимального рівня розподілу при комутації.

$$N = 84\,460 / 550 = 153,6 \text{ джерел.} \quad (2.3)$$

Він заповнює мережу з 96 ПК.



Інтенсивність потоку заявок, що відправляється кожним з 96 комп'ютерів, становить 550 кадрів на секунду.

Щільність вихідного трафіку від усіх користувачів:

$$\lambda = N \cdot \mu = 96 \cdot 550 = 52\,800 \text{ (пакетів/с)} \quad (2.4)$$

Параметри затримки на рівні розподілу впливають на час затримки в черзі і визначають рівень завантаження каналу зв'язку передачі.

$$\rho = \lambda / \mu_{\text{вих}} = 52\,800 / 84\,460 = 0,63 \quad (2.5)$$

Коефіцієнт зайнятості комутатора рівня розподілу:

$$r = \rho / (1 - \rho) = 0,63 / (1 - 0,63) = 1,7 \quad (2.6)$$

Середній час затримки кадру, обумовлений чергою M/M/1, становить:

$$T = 1 / ((\mu - \lambda)) = 1 / (84\,460 - 52\,800) = 31,6 \text{ мкс} \quad (2.7)$$

Середній розмір черги:

$$L_{\text{чер}} = \rho^2 / (1 - \rho) = (0,63)^2 / (1 - 0,63) = 1,07 \quad (2.8)$$

Ця цифра є важливою при управлінні чергою пристрою. У апаратному забезпеченні можна встановити максимальну кількість пакетів у черзі.

Середній час пакетів у черзі:

$$T_{\text{оч}} = L_{\text{чер}} / \lambda = 1,07 / 52\,800 = 20,3 \text{ мкс} \quad (2.9)$$

Це значення є нижчим за потрібне  $\leq 5$  мс, що відповідає вимогам.

Пропускна здатність каналу:

$$\lambda = (\text{пропускна здатність})/(\text{довжина кадру}) = b / l \quad (2.10)$$

$$b = \lambda * l = 52\,800 * 1480 * 8 = 625\,152\,000 \text{ біт/с} = 625,2 \text{ Мбіт/с} \quad (2.11)$$

Він відповідає пропускній здатності каналу на виході – 1000 Мбіт / с.

### 2.1.2 Імітаційна модель мережі в пакеті CiscoPacketTracer

На рисунку 2.1 зображена топологічна схема корпоративної мережі.

Для організації кінцевих мережних пристроїв використано п'ять підмереж, які розділені в залежності від функціональності та напрямку підрозділів організації. Підмережа №1, позначена як "Відділ дизайну", призначена для підключення 77 абонентів. Підмережа №2, позначена як "Відділ розробки", призначена для підключення 81 абонента. Підмережа №3, позначена як "Відділ продажу", розрахована на підключення 57 абонентів. Підмережа №4, позначена як "Відділ проєктного керування", розрахована на підключення 71 абонента. Підмережа №5, позначена як "Відділ підтримки й обслуговування", розрахована на підключення 96 абонентів.

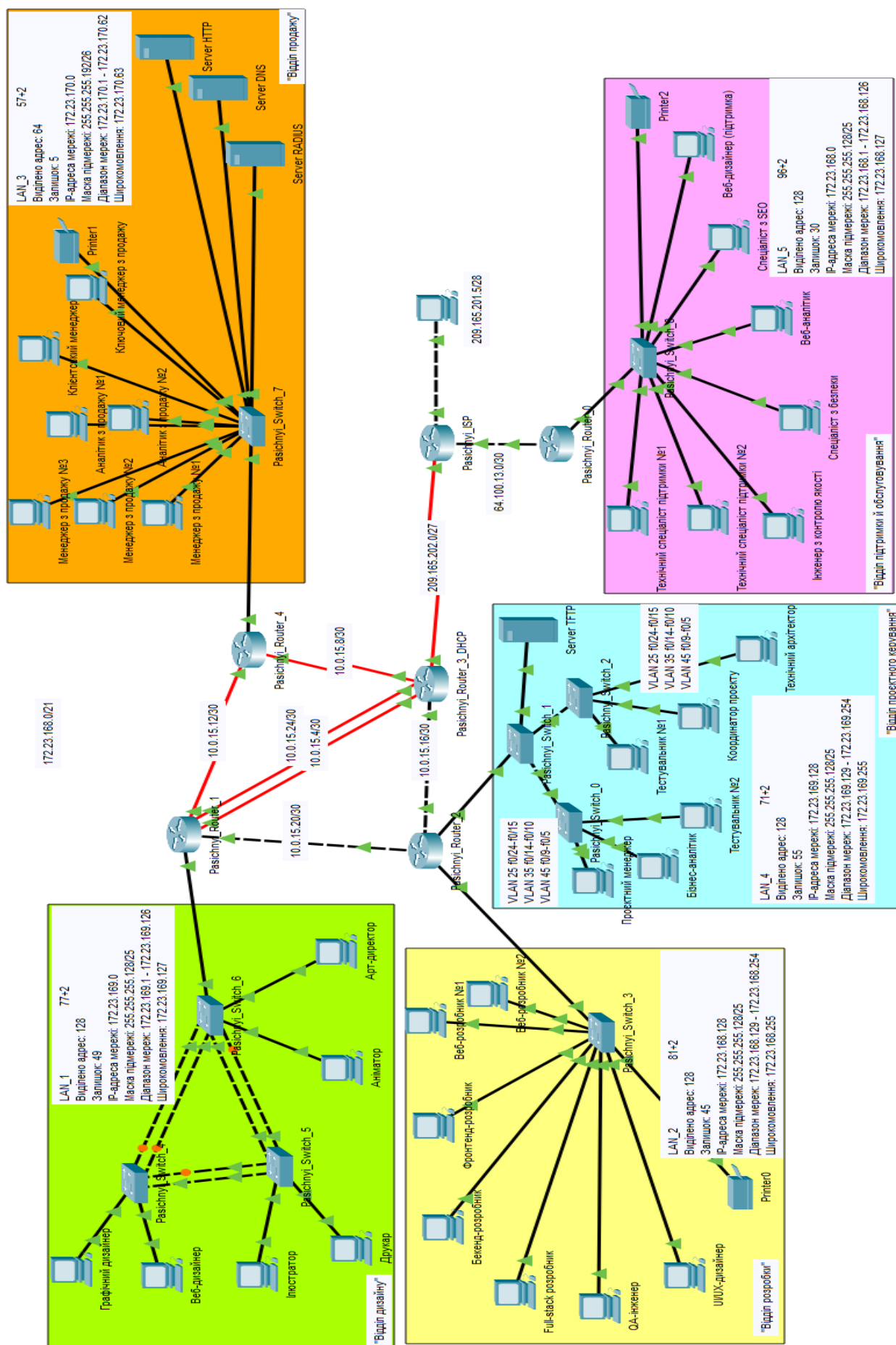


Рисунок 2.1 – Архітектура комп'ютерної системи

## 2.2 Імітаційне моделювання комп'ютерних та статистичних систем.

Основні принципи комп'ютерного моделювання полягають у такому: в основі математичної моделі та за допомоги комп'ютера проводяться обчислювальні експерименти, що дозволяє вивчити характеристики об'єкта або процесу. Наприклад, маючи рівняння, що описує процес, можна аналізувати, як змінюється поведінка об'єкта при варіюванні його коефіцієнтів, початкових умов і граничних умов.

Комп'ютерні експерименти на основі математичних моделей, які відтворюють поведінку реальних об'єктів, процесів або систем – це імітаційні моделі. Задля розгляду реальних систем та процесів використовують два основних типи моделей: аналітичні та імітаційні.

Поведінка реального процесу або системи в аналітичних моделях описується чітко визначеними функціональними залежностями, такими як лінійні або нелінійні рівняння, диференціальні або інтегральні рівняння, чи рівняння одночасності. Однак ці моделі підходять лише для відносно простих процесів і систем.

При вивченні складних і різноманітних явищ часто доводиться спрощувати реальну ситуацію, в результаті чого виникають наближені моделі, які не завжди точно відображають дійсність.

Імітаційне моделювання – це чисельний метод, який дозволяє проводити комп'ютерні експерименти на основі математичних моделей, що відтворюють поведінку реальних систем, об'єктів та процесів в часі. У процесі моделювання функціонування реального процесу або системи розбивається на окремі модулі, явища та підсистеми. Ці елементи можуть бути охарактеризовані алгоритмами, які відображають явища, що лежать в основі, зберігаючи їх логічний і часовий порядок.

Імітаційне моделювання передбачає розробку алгоритмів моделювання об'єктів, програмну реалізацію цих алгоритмів, а також організацію і планування обчислювальних експериментів на комп'ютерах з використанням МатМоделей

для відновлення функціонування реальних систем та процесів впродовж певного періоду часу. Воно охоплює методи, включаючи реалізацію.

Алгоритмізація ДПС передбачає поетапний опис функцій підсистем і окремих модулів з таким рівнем деталізації, який відповідає вимогам моделі.

Термін «імітаційне моделювання» (ІМ) є поєднанням двох понять – «імітація» і «моделювання», які є взаємозамінними. У різних галузях науки і техніки математичні моделі широко використовуються для вивчення реальних процесів.

Для того, щоб класифікувати різні типи моделей, дослідники ввели додаткову термінологію. Імітаційне моделювання – це розробка математичних моделей, які можуть передбачати та аналізувати поведінку системи. Це здійснюється шляхом проведення обчислювальних експериментів (симуляцій) з конкретними вхідними даними на основі математичної моделі.

Основні переваги імітаційного моделювання включають:

- Здатність детально описувати поведінку складових процесів і систем з високим рівнем точності.
- Відсутність обмежень на взаємозв'язки між параметрами моделі та умовами зовнішнього середовища РПС.
- Можливість аналізувати динаміку взаємодії елементів як у часовому, так і в просторовому вимірі.

Завдяки цим властивостям імітаційне моделювання набуло значного поширення.

Рекомендації для застосування імітаційного моделювання:

- Відсутність чіткої постановки завдання: Застосовується для вивчення та аналізу об'єкта моделювання, виконуючи роль інструменту для дослідження явищ.
- Складність аналітичних методів: Коли математичні обчислення є складними та часозатратними, імітаційне моделювання дозволяє спростити розв'язання задачі.

– Потреба аналізу поведінки системи: Дозволяє не лише оцінити вплив параметрів, але й спостерігати динаміку функціонування елементів системи протягом заданого часу.

– Єдиний метод дослідження: Застосовується для вивчення складних систем, які неможливо вивчати в реальних умовах, таких як термоядерні реакції або дослідження космосу.

– Контроль процесів: Застосовується для управління перебігом подій, зокрема для їх прискорення чи уповільнення під час моделювання.

– Навчання персоналу: Використовується для підготовки фахівців, які здобувають практичні навички роботи з новою технікою на основі моделей.

– Аналіз нових ситуацій у РПС: дає змогу оцінити ефективність нових стратегій і правил перед проведенням реальних експериментів.

– Визначення вузьких місць: Дає можливість передбачити проблемні ділянки у функціонуванні проєктованих систем шляхом аналізу послідовності подій.

Імітаційне моделювання, незважаючи на свої переваги, має також певні недоліки:

– Створення якісної моделі часто вимагає більших фінансових та часових витрат порівняно з аналітичним моделюванням.

– Модель може бути неточною, і в багатьох випадках неможливо визначити рівень цієї неточності.

– Науковці іноді звертаються до імітаційного моделювання, не усвідомлюючи складнощів, з якими можуть зіткнутися, що призводить до методологічних помилок.

Попри це, імітаційне моделювання залишається одним із найбільш популярних методів для аналізу та розробки складних систем і процесів.

Один із видів цього методу – статистичне імітаційне моделювання, яке дає змогу моделювати роботу складних випадкових процесів за допомогою комп'ютерних систем.

Для дослідження систем, що піддаються випадковим впливам, використовуються ймовірнісні аналітичні та імітаційні моделі. У ймовірнісних

аналітичних моделях випадкові фактори описуються через ймовірнісні характеристики, такі як закони розподілу, спектральна щільність або кореляційні функції. Однак розробка таких моделей є обчислювально складною задачею, тому вони застосовуються здебільшого для аналізу відносно простих систем.

З'ясовано, що введення випадкових збурень в імітаційні моделі зазвичай не спричиняє серйозних труднощів, тому дослідження складних випадкових процесів зазвичай проводиться саме за допомогою таких моделей.

У процесі імовірнісного імітаційного моделювання працюють не з загальними характеристиками випадкових процесів, а з конкретними значеннями випадкових параметрів ПС. Результати, отримані під час моделювання, є випадковими реалізаціями. Для отримання об'єктивних та стабільних характеристик процесу необхідно проводити моделювання багато разів, а потім здійснювати статистичний аналіз отриманих даних. Тому дослідження систем і процесів, що піддаються випадковим впливам, через імітаційне моделювання називають статистичним моделюванням.

Статистичне моделювання випадкових процесів – це алгоритм, який дозволяє відтворювати функціонування складних систем, що піддаються випадковим збуренням, і моделювати стохастичну взаємодію їх елементів. При статистичному імітаційному моделюванні на комп'ютері необхідно згенерувати послідовність чисел з випадковими значеннями, що відповідають заданим стохастичним характеристикам.

Корисно розрізняти ці поняття, оскільки методи Монте-Карло використовуються не тільки в статистичному моделюванні, але і в чисельних методах, таких як обчислення інтегралів і розв'язування рівнянь.

Статистичне моделювання – це метод дослідження складних процесів або систем, що піддаються випадковим збуренням, за допомогою імітаційних моделей. Методи Монте-Карло – це чисельні підходи, які дозволяють створювати на комп'ютері псевдовипадкові числові послідовності із заданими ймовірнісними характеристиками.

Процес статистичного моделювання складається з таких основних етапів:

- Генерація псевдовипадкових числових послідовностей з заданими характеристиками, такими як кореляція та закон розподілу ймовірностей, за допомогою методу Монте-Карло. Ці послідовності використовуються для моделювання випадкових значень параметрів у кожному випробуванні.
- Перетворення згенерованих числових послідовностей через застосування імітаційних математичних моделей.
- Статистичний аналіз отриманих результатів, що дозволяє зробити висновки про досліджувані процеси чи системи.

### 2.3 Підходи до моделювання комп'ютерних мереж

Основними підходами до моделювання комп'ютерних мереж є аналітичне та імітаційне моделювання.

Аналітичні моделі базуються на математичних методах, таких як теорія масового обслуговування, ймовірнісні теорії, марковські процеси та методи дифузійної апроксимації. Для опису динаміки мережі також можуть застосовуватись диференціальні та алгебраїчні рівняння.

Застосування аналітичного моделювання дозволяє створювати моделі, які допомагають вирішувати різноманітні завдання, пов'язані з вивченням комп'ютерних мереж. Проте цей підхід має низку суттєвих недоліків, зокрема:

- значне спрощення, яке властиве більшості аналітичних моделей, що інколи ставить під сумнів точність отриманих результатів;
- складність і висока трудомісткість обчислень для більш комплексних моделей.

Отже, незважаючи на значний прогрес в аналітичному моделюванні, багато реальних ситуацій не можна повноцінно описати за допомогою таких математичних підходів. У деяких випадках цьому заважає "жорсткість" математичних методів, які обмежують здатність точно відображати події та явища. В інших випадках, навіть якщо складну ситуацію вдається формалізувати



через математичну модель, задача оптимізації, що виникає, може бути занадто складною для сучасних алгоритмів.

Імітаційне моделювання, як другий основний метод моделювання комп'ютерних мереж, часто є найбільш ефективним і в деяких випадках навіть єдиним способом аналізу реальних систем, зокрема мереж. Термін "імітаційне моделювання" вказує на те, що досліджувані процеси є складними для точного прогнозування, і для передбачення їх поведінки необхідно проводити обчислювальні експерименти (імітації) з використанням заданих вхідних даних.

Відмінність між аналітичними та імітаційними моделями полягає в підході до опису системи. В імітаційних моделях замість явного математичного опису залежностей між вхідними та вихідними параметрами система поділяється на невеликі функціональні елементи або модулі. Поведінка системи моделюється як сукупна робота цих елементів, між якими встановлюються певні взаємозв'язки, що формують єдину модель.

Обчислювальне виконання імітаційної моделі починається з опрацювання вхідного елемента, далі поступово проходить через усі елементи моделі, доки не досягне вихідного елемента.

### **2.3.1 Математичне моделювання мережі як системи обслуговування великої кількості користувачів**

Механізм обслуговування визначається обсягом роботи, необхідного для обробки повідомлення, а одиниця виміру навантаження залежить від типу сервісного центру. Швидкість роботи сервісного обладнання в центрі визначається обсягом роботи, виконуваним за одиницю часу. Тривалість обслуговування одного повідомлення розраховується як співвідношення між робочим навантаженням і швидкістю обслуговування цього повідомлення.

$$\mu_i(n_i) = \begin{cases} n_i \mu_i, & 0 \leq n_i \leq A_i, \\ A_i \mu_i, & n_i > A_i, \end{cases} \quad i = \overline{1, M}, \quad (2.1)$$

де  $A_i$  – це кількість ідентичних обслуговуючих пристроїв в  $i$ -му центрі.

$$\tilde{F}(s) = \int_0^{\infty} e^{-st} dF(t) \quad (2.2)$$

є доцільною функцією такого виду:

$$\tilde{F}(s) = \tilde{E}(s)/\tilde{B}(s), \quad (2.3)$$

де  $\tilde{E}(s)$  і  $\tilde{B}(s)$  – відповідно поліномами ступенів  $d_1$  та  $d_2$

Клас функцій з раціональними перетвореннями Лапласа є дуже широким і включає експоненціальний, ерлангівський та суперекспоненціальний розподіли. Характеристики цих розподілів наведено в таблиці 2.1.

Таблиця 2.1 Характеристики деяких розподілів

Тип розподілу	Функція розподілу	Математичне сподівання	Дисперсія	Перетворення Лапласа
Експоненціальний	$F(t) = 1 - e^{-\mu t}, t \geq 0$	$\tau = \frac{1}{\mu}$	$D = \frac{1}{\mu^2}$	$\tilde{F}(s) = \frac{\mu}{\mu + s}$
Ерланга k-го порядку	$F(t) = 1 - e^{-\mu t} \times \sum_{j=0}^{k-1} \frac{(k\mu t)^j}{j!}, t \geq 0$	$\tau = \frac{1}{\mu}$	$D = \frac{1}{k\mu^2}$	$\tilde{F}(s) = \left(\frac{k\mu}{k\mu + s}\right)^k$
Гіперекспоненціальний k-го порядку	$F(t) = 1 - \sum_{i=1}^k b_i \mu_i e^{-\mu_i t}, \sum_{i=1}^k b_i = 1, t \geq 0$	$\tau = \begin{cases} \frac{1}{\mu}, \mu_i = \mu \\ \sum_{i=1}^k \frac{b_i}{\mu_i}, \text{ в ін. сум.} \end{cases}$	$D = \frac{1}{\mu^2} + \sum_{i=1}^k \sum_{j=1}^k b_i b_j \times \left(\frac{1}{\mu_i} - \frac{1}{\mu_j}\right)^2$	$F(s) = \sum_{i=1}^k \frac{b_i \mu_i}{\mu_i + s}$

### 2.3.1.1 Потоки в мережах

Після завершення обслуговування повідомлення переходять з одного центру в інший відповідно до визначеного маршруту. Маршрут повідомлення в мережі з чергою задається маршрутною матрицею  $P$ . У відкритій мережі повідомлення надходять із зовнішнього джерела і можуть покинути мережу після завершення обслуговування. Якщо зовнішнє джерело вважається новим центром мережі, а індекс дорівнює 0, то шлях відкритої мережі визначається

матрицею ймовірностей  $P$ , де  $p_{0j}$  – ймовірність надходження повідомлення до  $j$ -го центру з джерела,  $p_{ij}$  – ймовірність переходу повідомлення з  $i$ -го центру до  $j$ -го, а  $p_{i0}$  – ймовірність того, що повідомлення залишить мережу після обслуговування в  $i$ -му центрі.

Очевидно, що справедлива наступна рівність:

$$\sum_{j=0}^M P_{ij} = 1 \quad (i = 0, 1, \dots, M) \quad (2.4)$$

Випадком рекурентного потоку є пуассонівський потік, де сила потоку  $\lambda$  залежить від загальної кількості повідомлень в мережі ( $N$ ).

Задля визначення потоків, які регулярно циркулюють у відкритій мережі МеМО, вводиться коефіцієнт передачі  $e_i$  так що  $e_i \lambda(N)$  – це загальна інтенсивність потоків повідомлень до  $i$ -го центру в мережі. Інтенсивність  $e_i \lambda(N)$  складається з інтенсивності надходження повідомлень в  $i$ -й центр з джерела  $P_{0i} \lambda(N)$  і інтенсивності надходження від інших центрів  $e_j P_{ji} \lambda(N)$  ( $j = \overline{1, M}$ ). Таким чином, значення  $e_i$  задовольняє наступним одночасним лінійним рівнянням:

$$e_i = P_{0i} + \sum_{j=1}^M e_j P_{ji}, \quad i = 1, 2, \dots, M, \quad (2.5)$$

У мережі із закритою чергою жодне повідомлення не входить і не виходить з мережі, а циркулюючою кількістю повідомлень у мережі, є постійною і дорівнює  $N$ . Матриця  $P$ , визначає імовірнісний шлях руху повідомлень, як і у випадку відкритої мережі, яка є стохастичною і нестационарною, але в даному випадку не містить нульових стовпців і нульових рядків (оскільки відсутнє джерело повідомлень). Одночасне лінійне рівняння (2.5) набуває вигляду

$$e_i = \sum_{j=1}^M e_j P_{ji}. \quad (2.6)$$

Оскільки кількість незалежних рівнянь у системі рівнянь (2.6) на одиницю менша за кількість змінних, то її розв'язок є єдиним у межах константи множника.

### 2.3.1.2 Однорідні експоненціальні мережі

Описування математичного аналізу МеМО потрібно починати з завчення простих однорідних експоненціальних мереж або мереж, в яких функція розподілу часу обслуговування в центрі має експоненціальний розподіл, а вхідні потоки повідомлень мають пуассонівський розподіл (у випадку відкритих мереж) Це випадок відкритих мереж.

Розглянемо багатовимірний випадковий процес, що описує цю мережу.

$$N(t) = \{n_1(t), n_2(t), \dots, n_M(t)\}, \quad (2.7)$$

де  $n_k(t)$  – повідомлень у в  $k$ -му центрі (у черзі та на обробці) у момент часу  $t$  ( $k = \overline{1, M}$ ),  $i$  позначається

$$P(n, t) = P\{n_1(t) = n_1, n_2(t) = n_2, \dots, n_M(t) = n_M, t\} \quad (2.8)$$

ймовірність того, що в момент часу  $t$  мережа знаходиться у стані  $n = \{n_1, n_2, \dots, n_M\}$ .

Нехай  $S(N, M)$  – множина  $M$ -вимірних векторів з невід'ємними цілочисельними координатами.

$$S(N, M) = \left\{ n; n_i \geq 0, \sum_{i=1}^M n_i = N \right\}, \quad (2.9)$$

потужність якої (кількість станів) дорівнює  $C_{N+M-1}^{M-1}$ .

Випадковий процес  $N(t)$ , який визначений на просторі станів  $S(N, M)$ , є марковським. Аналізуючи можливі переходи цього процесу за певний проміжок

часу  $dt$  і рухаючись до межі при  $dt \rightarrow 0$ , отримаємо наступну систему прямих диференціальних рівнянь Колмогорова:

$$\begin{aligned} \frac{dP(n, t)}{dt} = & - \left( \sum_{k=1}^M \gamma(n_k) \alpha(n_k) \mu_k \right) P(n, t) + \\ & + \sum_{i=1}^M \sum_{k=1}^M \gamma(n_k) \alpha_i(n_i + 1) \mu_i P_{ik} P(n - 1_k + 1_i), \end{aligned} \quad (2.10)$$

де  $1_i$  – вектор з  $i$ -а координатою, що дорівнює 1, а решта – нулю. Ця функція  $\alpha_k(n_k) = \min\{n_k, A_k\}$  використовується для визначення загальної кількості повідомлень в  $k$ -му центрі ( $k = \overline{1, M}$ ), де  $n_k; \gamma(n_i) = 0$  при  $n_i = 0$  і  $\gamma(n_i) = 1$  при  $n_i \neq 0$ .

Розглянемо розв'язок системи (2.10) в стаціонарній області, яка завжди присутня в замкненій мережі МО. Для ймовірності стаціонарного розподілу марковського процесу  $N(t)$ , прирівнявши похідні лівої частини системи рівнянь (2.10) до нуля, отримаємо.

$$P(n) = \lim_{t \rightarrow \infty} P(n, t)$$

з урахуванням тотожності  $\gamma(n_k) \alpha_k(n_k) = \alpha_k(n_k)$  одержуємо наступну систему лінійних різницевих рівнянь:

$$\sum_{k=1}^M \alpha_k(n_k) \mu_k P(n) = \sum_{i=1}^M \sum_{k=1}^M \gamma(n_k) \alpha_i(n_i + 1) \mu_i P_{ik} P(n - 1_k + 1_i), n \in S(N, M) \quad (2.11)$$

Ліва частина є швидкістю переходів із стану  $n$ , а права – швидкість переходів в цей стан. Рівняння (2.11) називають рівнянням глобального балансу.

### 2.3.1.3 Розв'язок моделі в мультиплікативній формі

Переходячи до розв'язання системи рівнянь (2.11), введемо функцію

$$\beta_k(n_k) = \begin{cases} n_k! \mu_k, n_k \leq A_k, \\ A_k! A_k^{n_k - A_k}, n_k > A_k, \end{cases} \quad (2.12)$$

яка може бути представлена також рекурсивно:

$$\begin{aligned} \beta_k(0) &= 1, \\ \beta_k(n_k) &= \alpha_k(n_k) \beta_k(n_k - 1), n_k = 1, 2, \dots, N, \end{aligned} \quad (2.13)$$

введемо заміну змінних в (2.11):

$$P(n) = \prod_{i=1}^M \beta_i^{-1}(n_i) Q(n) \quad (2.14)$$

Після підстановки (2.14) в (2.11) одержимо нову систему рівнянь відносно  $Q(n)$ :

$$\sum_{k=1}^M \alpha_k(n_k) \mu_k Q(n) = \sum_{i=1}^M \sum_{k=1}^M \alpha_k(n_k) \mu_i P_{ik} Q(n - 1_k + 1_i), \quad (2.15)$$

Якщо представити  $Q(n)$  у вигляді функції  $M$  невідомих параметрів  $x_i$ :

$$Q(n) = \prod_{i=1}^M x_i \text{const} \quad (2.16)$$

і підставити у вираз (2.15), то (2.15) набуває наступного простого вигляду:

$$\sum_{k=1}^M \alpha_k(n_k) \left\{ \mu_k - \sum_{i=1}^M \mu_i P_{ik} \left( \frac{x_i}{x_k} \right) \right\} = 0 \quad (2.17)$$

Для подальшого спрощення останньої системи рівнянь припустимо, що всі  $N$  повідомлень знаходяться в  $j$ -му центрі  $j = \overline{1, M}$ , а отже, в інших центрах

повідомлень немає. Враховуючи, що функція  $\alpha_k(n_k)$  в цьому випадку за визначенням дорівнює нулю для всіх  $k \neq j$ , одержуємо

$$\mu_k x_k = \sum_{i=1}^M (\mu_i x_i) P_{ik} \quad (2.18)$$

або, ввівши позначення  $e_i = \mu_i x_i$ ,

$$e_k = \sum_{i=1}^M e_i P_{ik}, k = \overline{1, M}. \quad (2.19)$$

Таким чином, невідомий параметр  $x_k$  визначається з системи лінійних рівнянь (2.19), і її розв'язок є єдиним з точністю до константи множення  $\varepsilon$ , що обумовлено припущенням про вид матриці шляхів  $P$ .

Отже, стаціонарний розподіл ймовірності станів замкненої однорідної експоненціальної мережі МО має наступний вигляд:

$$P(n) = \prod_{i=1}^M [x_i^{n_i} / \beta_i(n_i)] G_M^{-1}(N). \quad (2.20)$$

Нормалізуюча константа  $G_M(N)$  визначається з умови нормування

$$1 = \sum_{n \in S(N, M)} P(n) = G_M^{-1} \sum_{n \in S(N, M)} \prod_{i=1}^M [x_i^{n_i} / \beta_i(n_i)] \quad (2.21)$$

Тут підсумовування ведеться по всіх  $C_{N-1}^{M+N-1}$  можливих станах вектора  $n \in S(N, M)$ .

З (2.21) витікає, що

$$G_M(N) = \sum_{n \in S(N, M)} \prod_{i=1}^M [x_i^{n_i} / \beta_i(n_i)] \quad (2.22)$$

Підставляючи (2.22) в (2.20) і враховуючи, що  $x_i = e_i / \mu_i$ , одержуємо

остаточно

$$P(n) = \frac{\prod_{i=1}^M \frac{e_i^{n_i}}{[\mu_i^{n_i} \beta_i(n_i)]}}{\sum_{n \in S(N,M)} \prod_{i=1}^M \frac{e_i^{n_i}}{[\mu_i^{n_i} \beta_i(n_i)]}} \quad (2.23)$$

Останнє рівняння показує, що стаціонарна ймовірність  $P(n)$  має вигляд добутку, який враховує стаціонарну ймовірність стану  $i$ -го центру  $Z_i(n_i) = e_i^{n_i} / [\mu_i^{n_i} \beta_i(n_i)] (i = \overline{1, M})$ , що розглядається окремо від мережі, незалежно від константи  $\epsilon$  точності, з якою визначається значення вектора  $\epsilon$ .

Рівняння (2.22) і (2.23) дозволяють визначити різні ймовірнісні властивості мережі МО. Наприклад, ймовірність того, що кількість повідомлень для  $i$ -го центру з одностороннім рухом ( $i = \overline{1, M}$ ) буде більшою або дорівнюватиме  $n$ , є такою.

$$P(n_i \geq n, N) = \frac{x_i^n}{G_M(N)} \sum_{n \in S(N-n, M)} \prod_{j=1}^M x_i^{n_j} = \frac{x_i^n G_M(N-n)}{G_M(N)} \quad (2.24)$$

З останнього виразу, з урахуванням того, що

$P_i(n, N) = P(n_i \geq n, N) - P(n_i \geq n+1, N)$ , визначається граничний розподіл числа повідомлень, що знаходяться в  $i$ -му вузлі ( $i = \overline{1, M}$ ):

$$P_i(n, N) = x_i^n [G_M(N-n) - x_i G_M(N-n-1)] / G_M(N) \quad (2.25)$$

Ці рівняння справедливі як для незалежних від навантаження мереж, так і для мереж, в яких інтенсивність обслуговування центру залежить від кількості повідомлень.

За аналогією з рівнянням (2.25), граничний розподіл кількості повідомлень в  $M$ -му (граничному) центрі можна записати наступним чином:

$$\begin{aligned} P_M(n, N) &= \sum_{n \in S(N, M), n_M = n} \prod_{i=1}^M Z_i(n_i) / G_M(N) = \\ &= Z_M(n) \sum_{n \in S(N-n, M-1)} \prod_{i=1}^M Z_i(n_i) / G_M(N), \quad n = \overline{1, N}. \end{aligned} \quad (2.26)$$



З урахуванням виразу (2.26) для нормалізуючої константи маємо

$$P_M(n, N) = Z_M(n)G_{M-1}(N - n)/G_M(N) \quad (2.27)$$

Граничний розподіл в будь-якому центрі  $i \neq M$  отримується шляхом перенумерації центрів таким чином, щоб цей центр став граничним центром. Рівняння (2.27) застосовується до отриманої таким чином мережі. Для незалежних центрів  $M$ , який не залежить від навантаження ( $\mu_M(n) = \mu_M$ ), легко можна побачити, що рівняння (2.23) може бути перетворено до вигляду рівняння (2.11). У цьому випадку  $i=1,2,\dots,M-1$  при визначенні властивостей пронумерованих вузлів не потрібно додаткової перенумерації.

За визначенням, інтенсивність  $\lambda_i(N)$  вихідного потоку повідомлень з  $i$ -го центру дорівнює середній кількості запитів, оброблених цим центром за одиницю часу. Отже:

$$\lambda_i(N) = \sum_{n=1}^N P_i(n, N)\mu_i(n) \quad (2.28)$$

З визначення  $Z_i(N)$  слідує, що

$$Z_i(n)\mu_i(n) = Z_i(n - 1)e_i \quad (2.29)$$

Підставляючи (2.26) і (2.27) в (2.28), одержуємо для  $i = \overline{1, M}$

$$\lambda_i(N) = e_i \sum_{n=1}^N \frac{Z_i(n - 1)G_{i-1}(N - n)}{G_i(N)} = e_i G_i(N - 1)/G_i(N) \quad (2.30)$$

Якщо інтенсивність обслуговування в  $i$ -му центрі висока, то продуктивність  $\lambda_i(N)$  можна визначити за середньою кількістю зайнятих пристроїв  $u_i(n)$  в  $i$ -му центрі:

$$\lambda_i(N) = \sum_{n=1}^N \mu_i(n)P_i(n, N) = \mu_i u_i(N), i = \overline{1, M} \quad (2.31)$$

де  $u_i(n)$  задовольняє наступному співвідношенню:

$$u_i(N)/u_j(N) = e_i \mu_i^{-1} / e_j \mu_j^{-1}. \quad (2.32)$$

З співвідношення (2.32), після того, як характеристики одного вузла були розраховані або виміряні, можна визначити зайнятість і пропускну здатність всіх центрів.

З (2.31) і (2.32) випливає рівність нормованої пропускну здатності кожного центру.

$$\lambda_i(N)/e_i = \lambda_j(N)/e_j \quad (2.33)$$

Із співвідношення  $u_i(N) \leq u_i(N+1)$  ( $N = 0, 1, \dots$ ) слідує, що при  $N \rightarrow \infty$  один з центрів, наприклад  $j$ -й, виявиться насиченим, так що  $u_j(\infty) = A_j$ . Очевидно, що пропускну спроможність цього центру при  $N \rightarrow \infty$  буде  $\lambda_j(\infty) = A_j \mu_j$  і значення  $u_j(\infty), \lambda_j(\infty)$  для решти центрів мережі можуть бути визначені з (2.29). Цей метод часто використовується для дослідження асимптотичних властивостей мережі з закритими чергами.

Враховуючи рівняння (2.27), математичне сподівання кількості повідомлень у  $m$ -му центрі має вигляд

$$L_M(N) = \sum_{n=1}^N n P_M(n, N) = \frac{1}{G_M(N)} \sum_{n=1}^N n Z_M(n) G_{M-1}(N-n) \quad (2.34)$$

Для вузла, що не залежить від навантаження, вираз, що описує середню довжину черги в центрі, спрощується:

$$L_i(N) = \sum_{n=1}^N x_i^n G_M(N-n) / G_M(N), \quad i = \overline{1, M} \quad (2.35)$$

Згідно з формулою Літгла, середній час очікування повідомлення  $T_i(N)$  в  $i$ -му центрі обслуговування дорівнює відношенню середньої довжини черги до

середньої інтенсивності вхідного потоку. У стаціонарному стані інтенсивність вихідних потоків дорівнює інтенсивності вхідних потоків.

$$T_i(N) = L_i(N)/\lambda_i(N), \quad i = \overline{1, M}. \quad (2.36)$$

Особливий інтерес представляє час циклу  $V_i(N)$  – середнє значення інтервалу часу між моментом виходу з  $i$ -го центру і першим приходом повідомлення, призначеного для цього центру.  $e_i=1$ , то значення  $e_j$  можна інтерпретувати як середню кількість разів, коли повідомлення відвідує  $j$ -й центр протягом двох послідовних візитів до  $i$ -го центру. Його можна інтерпретувати як середню кількість візитів. Таким чином, загальний середній час перебування повідомлення в  $j$ -м у центрі протягом одного циклу становить. Таким чином.

$$V_i(N) = \sum_{j=1}^M e_j T_j(N) \quad (2.37)$$

Підставляючи в (2.37) вираз (2.36) і враховуючи співвідношення (2.33), яке в даному випадку має вигляд  $\lambda_i(N) = \lambda_j(N)e_j$ , маємо:

$$V_i(N) = [N - L_i(N)]/\lambda_i(N).$$

Таким чином, час циклу  $V_i(N)$  однозначно визначається середньою довжиною черги та продуктивністю  $i$ -го центру.

Метод дослідження локально збалансованих мереж МО, описаний у попередніх параграфах, дозволяє отримати розв'язок у зручній мультиплікативній формі. Однак цей розв'язок залежить від константи нормалізації, яка має відносно простий вигляд у відкритих мережах МО, але є сумою добутків у закритих мережах. Кількість доданків у цій сумі відповідає степені простору станів мережі, навіть в однорідних закритих мережах. В результаті комбінаторного зростання простору станів мережі МО зі збільшенням кількості центрів, класів і повідомлень, потужність простору станів мережі швидко зростає, що робить пряме обчислення константи нормалізації дуже складним для мереж з великою розмірністю.

Тому були розроблені спеціальні методи для обчислення стаціонарних ймовірностей та інших властивостей замкнутих мереж МО, які складають окремий розділ теорії мереж МО.

### 2.3.1.4 Метод Бузена

Відповідно до цього методу алгоритм розрахунку нормалізуючої константи

$$G_M(N) = \sum_{n \in (N, M)} \prod_{i=1}^M Z_i(n_i), \quad (2.38)$$

де множник  $Z_i(n_i)$  має вигляд

$$Z_i(n_i) = \begin{cases} e_i^{n_i} / \prod_{j=1}^{n_i} \mu_i(j), & \text{якщо центр } i \text{ залежить від навантаження,} \\ (e_i / \mu_i)^{n_i}, & \text{якщо центр } i \text{ не залежить від навантаження,} \end{cases}$$

а простір станів мережі

$$S(n, M) = \left\{ \mathbf{n} = (n_1, n_2, \dots, n_M) \mid \sum_{i=1}^M n_i = N, n_i \geq 0, i = \overline{1, M} \right\} \quad (2.39)$$

зводиться до простої ітеративної процедури, суть якої полягає в наступному.

Розглянемо функцію 
$$g(n, m) = \sum_{n \in S(n, m)} \prod_{i=1}^m Z_i(n_i). \quad (2.40)$$

Очевидно, що при  $g(N, M) = G_M(N)$  та  $g(n, M) = G_M(n)$  для  $n = \overline{0, N}$  при  $m > 1$  маємо

$$\begin{aligned} g(n, m) &= \sum_{k=1}^n \left[ \sum_{n \in S(n, m), n_m = k} \prod_{i=1}^M Z_i(n_i) \right] = \sum_{k=0}^n Z_m(k) \sum_{n \in S(n-k, m-1)} \prod_{i=1}^{m-1} Z_i(n_i) = \\ &= \sum_{k=0}^n Z_m(k) g(n-k, m-1). \end{aligned}$$

Таким чином, для мережі, що складається з центрів, інтенсивність обслуговування яких залежить від навантаження

$$G_M(N) = \sum_{k=1}^N Z_M(k) G_{M-1}(N-k).$$

Якщо центр  $t$  не залежить від навантаження, той вираз (2.39) можна спростити. В цьому випадку

$$Z_m(k) = (e_m/\mu_m)^k = x_m^k = x_m Z_m(k-1). \quad (2.41)$$

Підставивши (2.41) у вираз (2.39), після спрощень одержуємо

$$g(n,m) = g(n,m-1) + x_m g(n-1,m). \quad (2.42)$$

Формули (2.42) і (2.39) дозволяють здійснювати рекурентне обчислення  $g(n,m)$

У табл. 2.2 і 2.3 показаний схематичний опис алгоритму Бузена. Стівці таблиці заповнюються послідовно зверху вниз. Якщо центр не залежить від навантаження, то елементи стівця  $t$  обчислюються по формулі (2.42), інакше – використовується вираз (2.40) і для відшукування  $Z_j(k)$  застосовується рекурентний вираз

$$Z_i(k) = \begin{cases} \frac{e_i}{\mu_i(k)} Z_i(k-1), & \text{якщо вузол } i \text{ залежить від навантаження,} \\ x_i Z_i(k-1), & \text{якщо вузол } i \text{ не залежить від навантаження.} \end{cases} \quad (2.43)$$

Шукане значення нормалізуючої константи  $G_M(N)$  знаходиться у нижньому правому куті таблиці.

Таблиця 2.2 Метод Бузена. Центри не залежать від навантаження.

$n \backslash m$	1	2	...	$m-1$	$m$	...	$M$
0	1	1	...	1	1	...	1
1	$x_1$						$g(1,M)$ $G_M(1)$
$\vdots$	...				$g(n-1,m) \times x_m$ $\downarrow$		...
$n-1$	$x_1^{n-1}$			$g(n,m-1) \rightarrow$	$g(n,m)$		$g(n-1,M)$ $G_M(n-1)$
$n$	$x_1^n$						$g(n,M)$ $G_M(n)$
...	...						...
$N$	$x_1^N$						$g(N,M)$ $G_M(N)$

Таблиця 2.3 Метод Бузена. Центри залежать від навантаження.

$\begin{matrix} m \\ n \end{matrix}$	1	...	m-1	m	...	M
0	1	...	$g(0,m-1)Z_m(n) \rightarrow$	$\begin{matrix} \rightarrow + \\ \downarrow \end{matrix}$	...	1
1	$Z_1(1)$		$g(1,m-1)Z_m(n-1) \rightarrow$	$\begin{matrix} \rightarrow + \\ \downarrow \end{matrix}$		$\begin{matrix} g(1,M) \\ G_M(1) \end{matrix}$
...	...		...	$\begin{matrix} \downarrow \\ \downarrow \end{matrix}$		...
n-1	$Z_1(n-1)$		$g(n-1,m-1)Z_m(1) \rightarrow$	$\begin{matrix} \rightarrow + \\ \downarrow \end{matrix}$		$\begin{matrix} g(n-1,M) \\ G_M(n-1) \end{matrix}$
n	$Z_1(n)$		$g(n,m-1)Z_m(0) \rightarrow$	$\rightarrow + \rightarrow g(n,r)$		$\begin{matrix} g(n,M) \\ G_M(n) \end{matrix}$
...	...					...
N	$Z_1(N)$					$\begin{matrix} g(N,M) \\ G_M(N) \end{matrix}$

Подальше спрощення обчислювальної процедури можливе в окремому випадку мереж, що залежать від навантаження, коли  $m$ -й центр мережі містить  $A_m$  однакових обслуговуючих приладів. При цьому інтенсивність обслуговування  $m$ -го центру:

$$\mu_m(k) = \begin{cases} k\mu_m, & \text{якщо } 0 \leq k \leq A_m, \\ A_m\mu_m, & \text{якщо } k > A_m; \end{cases} \quad (2.44)$$

$$Z_m(k) = \begin{cases} \frac{x_m^k}{k!}, & \text{якщо } 0 \leq k \leq A_m, \\ \frac{x_m^k}{A_m! A_m^{k-A_m}}, & \text{якщо } k > A_m \end{cases} \quad (2.45)$$

Останній вираз можна записати у рекурентному вигляді

$$Z_m(k) = x_m d_m(k) Z'_m(k-1), k \geq 1, \quad (2.46)$$

де  $Z'_m(k) = x_m^k$ ;

$$d_m(k) = \begin{cases} [k!]^{-1}, & \text{якщо } 0 \leq k \leq A_m \\ [A_m! A_m^{k-A_m}]^{-1}, & \text{якщо } k > A_m + 1. \end{cases} \quad (2.47)$$

Підставляючи (2.46) в (2.39), одержуємо

$$g(n, m) = g(n, m - 1) + x_m \left[ \sum_{k=0}^{A_m-1} \frac{1}{k+1} Z_m(k) g(n-k-1, m-1) + \sum_{k=A_m}^{n-1} \frac{1}{A_m} Z_m(k) g(n-k-1, m-1) \right] \quad (2.48)$$

Після перетворень маємо остаточно

$$g(n, m) = g(n, m - 1) + \frac{x_m}{A_m} \left[ g(n-1, m) + \sum_{k=0}^{A_m-2} \left( \frac{A_m-k-1}{k+1} \right) Z_m(k) g(n-k-1, m-1) \right] \quad (2.49)$$

Для мережі, що не залежить від навантаження ( $A_m = 1, m = \overline{1, M}$ ), вираз (2.49) співпадає з (2.42). Якщо  $A_m = 2$ , то

$$g(n, m) = g(n, m - 1) + x_m [g(n-1, m) + g(n-1, m-1)]/2.$$

вираз (2.49) дозволяє значно скоротити обчислення по порівнянню з прямим використанням формули (2.39) для випадку, коли центр  $\tau$  складається з кількох однакових обслуговуючих центрів.

Відмітимо, що нормалізуючі константи, які обчислені в результаті роботи описаного вище алгоритму не є однозначними, оскільки залежать від конкретного вибору величин  $e_j$ , які визначаються з системи рівнянь

$$e_j = \sum_{i=1}^M e_j P_{ij}, j = \overline{1, M}.$$

Розв'язок цієї системи рівнянь, єдиний з точністю до мультиплікативної константи, тому конкретне значення нормалізуючи констант  $G_M(n)$  можна одержати шляхом довільного завдання  $e_{i^*}$ , наприклад  $e_{i^*} = 1$ [3].

### 2.3.1.5 Оцінка характеристик моделі як замкнутої мережі СМО

Ми проаналізували алгоритм розрахунку ймовірностей станів мережі СМО. Знаючи ці ймовірності, можна розрахувати всі необхідні характеристики мережі, зокрема середній час відгуку мережі на запити користувачів.

Послідовність розрахунків може бути такою:

- 1) Розрахунок коефіцієнтів передачі  $e_i$  СМО, що входять вмережу СМО:  $ae_i = \lambda_j/\lambda_i$ , де  $\lambda_j$ , - інтенсивність потоку на вході (виході).
- 2) Обчислення ймовірностей  $P_N = \text{Вер}(N_1, \dots, N_N)$  станів мереж і СМО й ймовірностей  $P$  знаходження  $k$  запитів в кожній  $j$ -й СМО мережі.
- 3) Обчислення середніх чисел запитів  $n_i$ , у кожній  $i$ -ой СМО мережі по формулі.
- 4) Обчислення середніх чисел запитів  $n_{wi}$  в черзі кожної СМО:

$$\bar{n}_{wi} = \sum_{k=m_i+1}^N (k - m_i) \cdot P_k^{(i)}, i = 1, 2, \dots, n. \quad (2.50)$$

- 5) Обчислення середніх термінів знаходження  $m_{Ti}$ , запиту в кожній  $i$ -й СМО

$$m_{Ti} = \frac{\bar{n}_i}{(\bar{n}_i - \bar{n}_{wi})\mu_i}, i = 1, 2, \dots, n. \quad (2.51)$$

- 6) Обчислення середніх термінів реакції мережі СМО  $m_T$  на запит користувача:

$$m_T = \sum_{j=2}^n \alpha_j m_{Tj}. \quad (2.52)$$



## **3 СИНТЕЗ КОМП'ЮТЕРНОЇ СИСТЕМИ ПІДПРИЄМСТВА З ІНТЕРНЕТ ПРОДАЖІВ**

### **3.1 Призначення, функції комп'ютерної системи**

Призначення комп'ютерної системи для компанії з інтернет-продажів полягає у забезпеченні стабільної, швидкої та безпечної роботи всіх ключових бізнес-процесів, пов'язаних з електронною комерцією та обслуговуванням клієнтів. Основними завданнями системи є:

**а) Підтримка онлайн-продажів**

1) Забезпечення безперебійної роботи веб-сайтів та онлайн-магазинів для прийому та обробки замовлень від клієнтів.

2) Оптимізація роботи веб-додатків та інтегрованих платіжних систем для швидкого оформлення покупок.

**б) Управління клієнтською інформацією та замовленнями**

1) Збір, обробка та збереження інформації про клієнтів, їхні замовлення та історію покупок у централізованій базі даних.

2) Синхронізація даних між усіма відділами компанії для ефективного управління замовленнями, доставкою та логістикою.

**в) Підтримка серверної інфраструктури**

1) Централізоване зберігання даних на файлових серверах для обробки замовлень, роботи з каталогами товарів та звітністю.

2) Резервне копіювання та захист даних від втрати чи несанкціонованого доступу.

**г) Оптимізація внутрішньої комунікації**

1) Забезпечення швидкої та безпечної передачі даних між відділами компанії, філіалами та головним офісом.

2) Робота поштових серверів для обміну інформацією з клієнтами, партнерами та між співробітниками компанії.

**г) Моніторинг та аналіз діяльності**

1) Збір аналітичних даних про відвідуваність веб-ресурсів, ефективність рекламних кампаній та обсяги продажів.

2) Надання звітності для прийняття управлінських рішень та оптимізації бізнес-процесів.

д) Підтримка логістики та обслуговування клієнтів

1) Управління процесами доставки товарів, включаючи інтеграцію з логістичними службами.

2) Забезпечення швидкого обслуговування клієнтів через кол-центри, онлайн-чати та служби підтримки.

Підрозділи системи:

– Дизайну: відповідає за розробку зовнішнього вигляду продукту або веб-ресурсу.

– Розробки: займається технічною реалізацією веб-сайтів, платформ та інтеграцій.

– Продажу: працює з клієнтами, проводить комунікації, укладає угоди.

– Проєктного керування: координує роботу всіх підрозділів, слідкує за виконанням строків.

– Підтримки й обслуговування: забезпечує післяпродажну підтримку клієнтів і технічне обслуговування.

Відділ дизайну:

– Задача: створення дизайну веб-сайту або продукту.

– Виконавець: графічний дизайнер, ілюстратор, аніматор, веб-дизайнер.

– Процес: дизайнери отримують завдання від проєктного менеджера, розробляють макети і передають їх на узгодження з клієнтом.

– Взаємозв'язок: дизайнери тісно співпрацюють з продажами та розробкою для узгодження технічних і візуальних аспектів.

Відділ розробки:

– Задача: технічна реалізація веб-сайту або продукту.

– Виконавець: веб-розробники, бекенд-розробник, фронтенд-розробник.

- Процес: розробники отримують затверджений дизайн та технічне завдання, розробляють код і проводять тестування.

- Взаємозв'язок: тісна взаємодія з відділом дизайну і проєктного керування для дотримання вимог та узгодження технічних нюансів.

Відділ продажу:

- Задача: робота з клієнтом, фіналізація продажу.

- Виконавець: менеджер з продажів, аналітик з продажів.

- Процес: після завершення розробки відділ продажів взаємодіє з клієнтом, демонструє кінцевий продукт, веде переговори та підписує контракти.

- Взаємозв'язок: співпраця з підтримкою для передачі продукту на обслуговування.

Відділ проєктного керування:

- Задача: отримання замовлення, постановка задачі.

- Виконавець: проєктний менеджер, тестувальник, координатор проєкту, технічний архітектор.

- Процес: замовлення надходить до проєктного керування, яке формулює технічне завдання та створює дорожню карту.

Відділ підтримки й обслуговування:

- Задача: технічна підтримка та обслуговування клієнтів після продажу.

- Виконавець: технічний спеціаліст підтримки, спеціаліст з безпеки.

- Процес: надання технічної допомоги, вирішення проблем клієнтів, оновлення програмного забезпечення.

- Взаємозв'язок: тісна співпраця з розробкою для вирішення технічних питань і з продажами для підтримки клієнтських запитів.

### **3.2 Структура комплексу технічних засобів комп'ютерної системи підприємства**

У сучасних умовах функціонування підприємств, орієнтованих на інтернет-продажі, ключову роль відіграє оптимальна структура комп'ютерної системи, що забезпечує ефективну взаємодію між усіма компонентами бізнес-процесів. Основна мета побудови такої системи – створення технічної інфраструктури, здатної підтримувати високі вимоги до обробки даних, масштабованості та безпеки.

Комп'ютерна система підприємства включає апаратне та програмне забезпечення, яке дозволяє реалізовувати основні функції електронної комерції, такі як обробка замовлень, управління товарними запасами, здійснення платежів і комунікація з клієнтами. Центральним елементом технічного комплексу є сервери, що забезпечують обробку великих обсягів інформації та підтримку веб-інтерфейсу інтернет-магазину. Важливим доповненням до серверної інфраструктури є мережеве обладнання, що забезпечує стабільність і швидкість передачі даних.

На рисунку 3.1 зображена структурна схема підключення пристроїв в мережі.

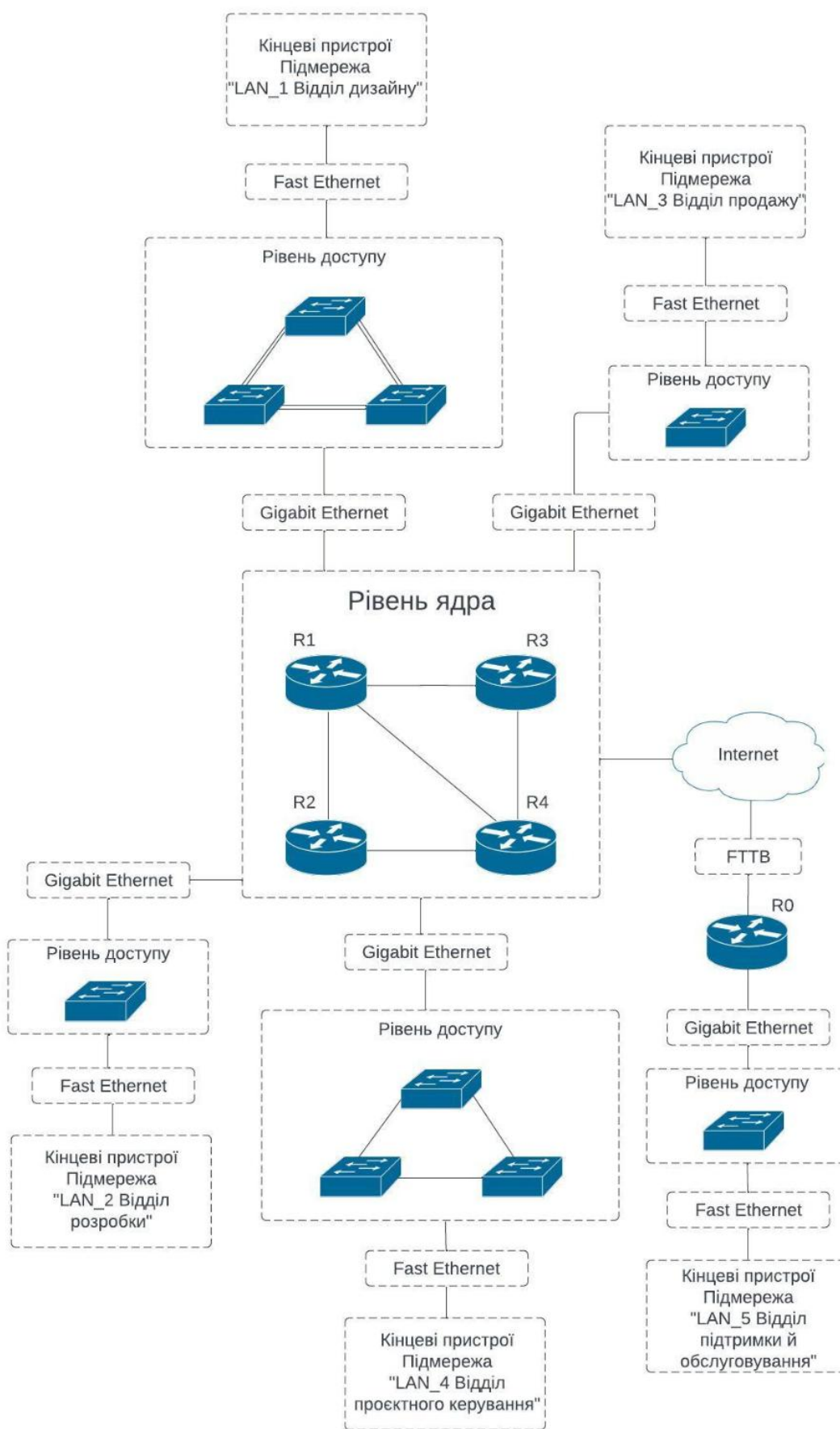


Рисунок 3.1 – Структурна схема комплексу технічних засобів комп'ютерної системи

### 3.3 Схема функціональної структури комп'ютерної системи

Проаналізувавши структурну схему комп'ютерної системи, розроблена функціональна схема, зображена на рисунку 3.2.

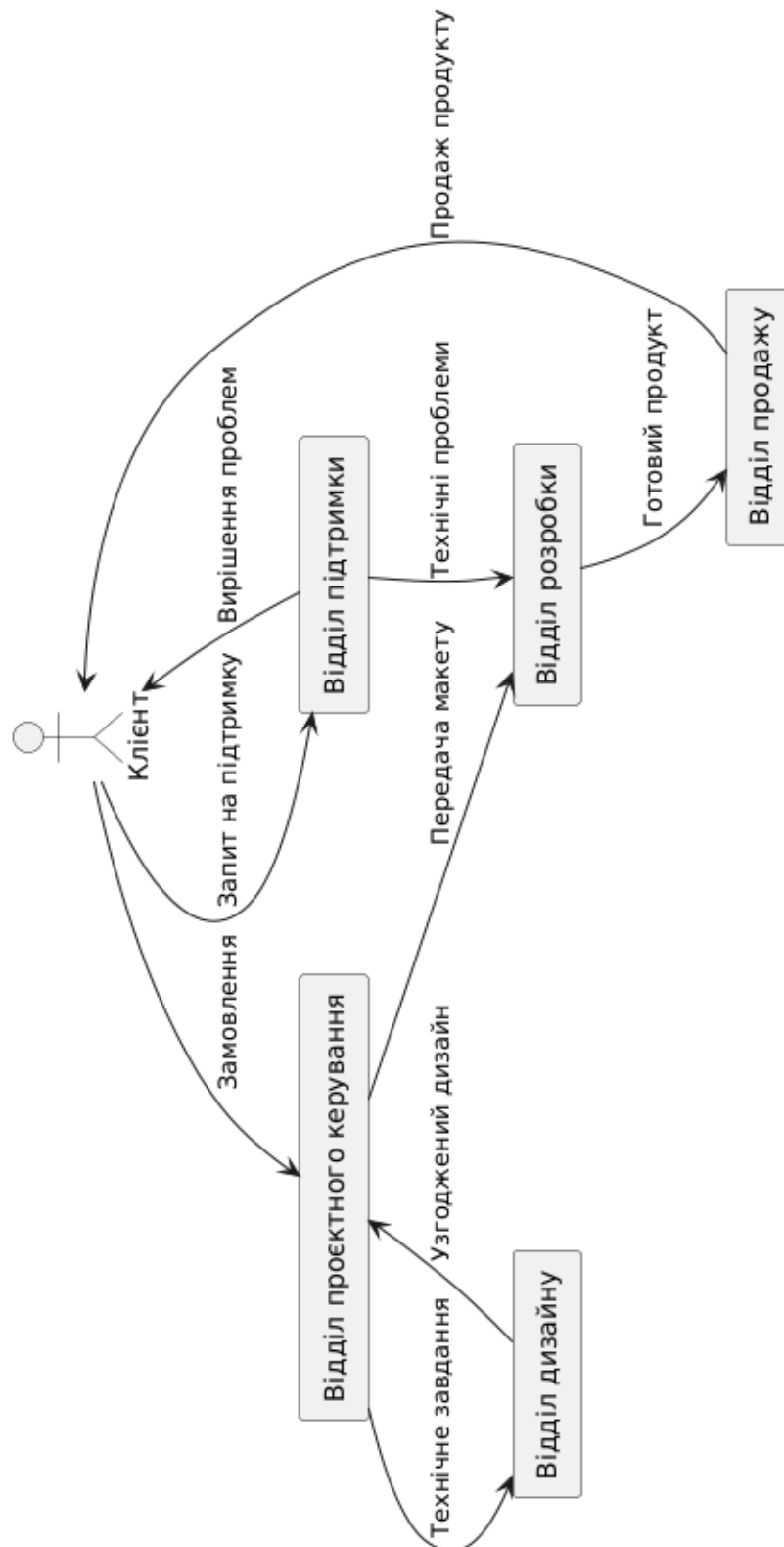


Рисунок 3.2 – Схема функціональної структури комп'ютерної системи

На цій схемі зображено функціональну архітектуру мережевої інфраструктури підприємства, яка складається з кількох взаємопов'язаних рівнів.

У верхній частині відображено ядро мережі, що складається з маршрутизаторів, які виконують функції маршрутизації, резервування, обробки даних та передачі трафіку між різними сегментами мережі. Маршрутизатори підключені до інтернет-шлюзу, що забезпечує доступ до глобальної мережі Інтернет.

Наступним рівнем є рівень доступу, де розміщені комутатори. Ці пристрої відповідають за з'єднання кінцевих пристроїв і забезпечують доступ відділам підприємства до мережевих ресурсів.

На нижньому рівні представлено кінцеві пристрої різних відділів підприємства. Кожен відділ виконує свою конкретну функцію, наприклад, дизайн, розробку, продаж, керування проектами або технічну підтримку.

Таким чином, схема демонструє, як дані передаються від кінцевих пристроїв через рівень доступу до ядра мережі та як забезпечується взаємодія між різними підрозділами та Інтернетом.

### **3.4 Схема структури інформаційних зав'язків комп'ютерної системи**

Проаналізувавши структурну схему комп'ютерної системи, розроблена схема структури інформаційних зав'язків, зображена на рисунку 3.3.

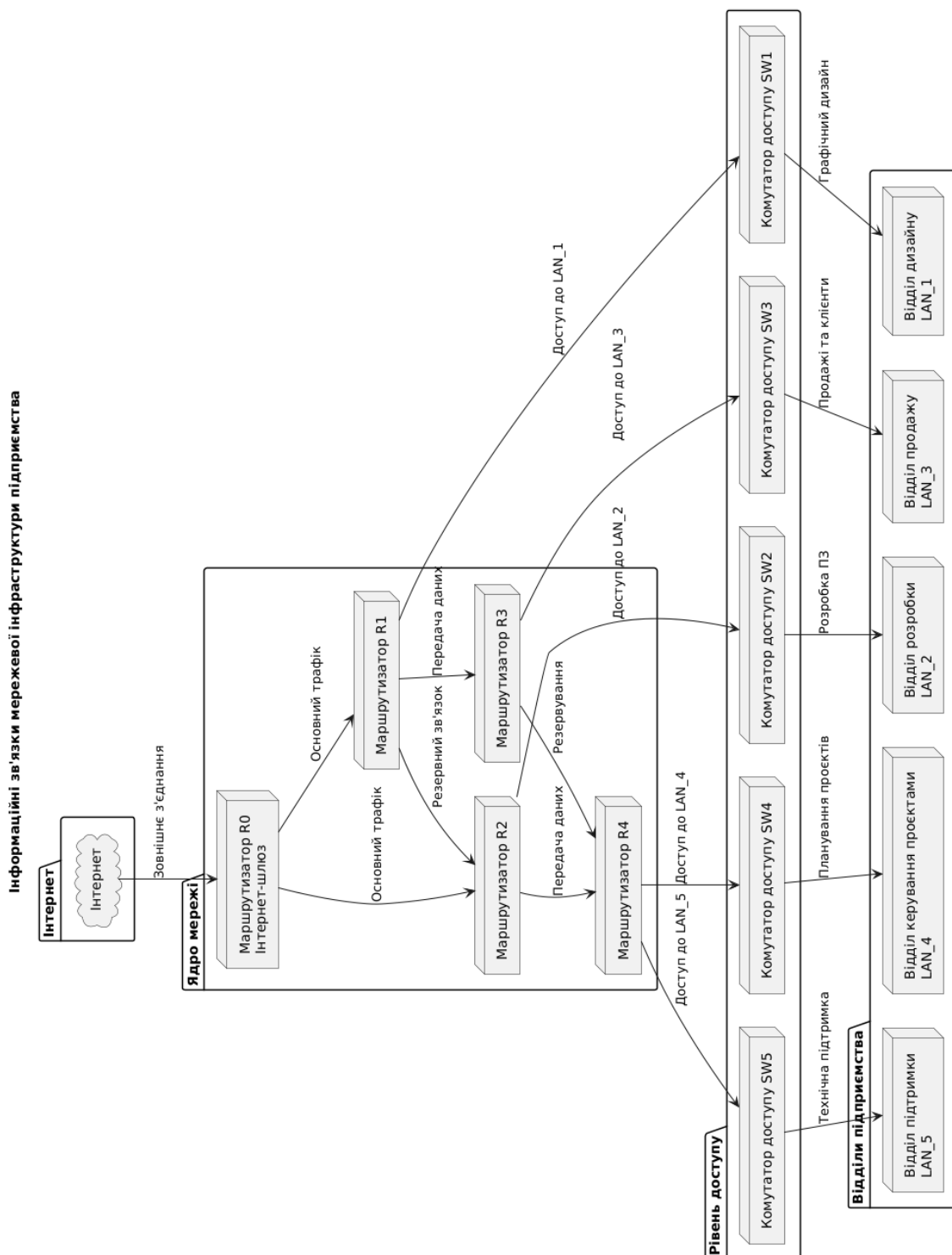


Рисунок 3.3 – Схема структури інформаційних зав'язків

На цій схемі зображена мережева інфраструктура підприємства з інформаційними зв'язками між її основними елементами. Центральним вузлом є Інтернет, який забезпечує зовнішній зв'язок і підключається до маршрутизатора-шлюзу. Далі трафік розподіляється на ядро мережі, що складається з маршрутизаторів, які керують потоками даних, забезпечують основні зв'язки та резервування.



Ядро мережі з'єднане з рівнем доступу, що представлений комутаторами. Ці комутатори виступають проміжними вузлами між ядром та кінцевими користувачами в різних відділах підприємства. Кожен відділ відповідає за виконання конкретних завдань: дизайн, розробку програмного забезпечення, продажі, управління проєктами та технічну підтримку.

Схема демонструє чітку ієрархію мережі: від глобального Інтернету до локальних підмереж різних відділів. Кожен рівень мережі має свою функцію: розподіл трафіку, обробка даних і надання доступу кінцевим користувачам. Завдяки резервуванню зав'язків забезпечується стійкість та надійність системи в разі збоїв.

### **3.5 Особливості структурної схеми комп'ютерної системи**

З огляду на структуру, корпоративна мережа є мережею змішаної топології, що складається з кількох локальних мереж. Ця мережа об'єднує структурні підрозділи, створюючи спільний інформаційний корпоративний простір. Таким чином, корпоративна мережа відображає структуру підприємства.

До складу технічних засобів комп'ютерної системи входять робочі станції (хости), сервери підприємства, маршрутизатори, комутатори і мережні комунікації, включаючи кабелі і бездротові адаптери.

Для перемикання трафіку використовується ядро, яке складається з чотирьох маршрутизаторів (R1 до R4), вони забезпечують з'єднання мереж. Для забезпечення доступу до ядра використовується технологія Gigabit Ethernet. Для підключення мережі до Інтернету використовується роутер R0. Підмережі, які керує підприємство, є віддаленими мережами, доступ до яких здійснюється через Інтернет.

У решті локальних мереж використовується по одному комутатору, який забезпечує підключення кінцевих мережевих пристроїв до локальної мережі.

Підмережа «Відділ проєктного керування» містить три комутатора. За допомогою VLAN, було здійснено підключення до мережі.

У таблиці 3.1 вказано кількість обладнання, яке надає користувачам або працівникам відділу, доступ до ресурсів та послуг. Ця таблиця була створена згідно зі схемою організаційної структури, топологічною схемою розташування структурних підрозділів та робочих місць.

Таблиця 3.1 – Кількість кінцевих пристроїв

Відділ	Ідентифікатор	Тип	Посада	Кількість
Дизайну	PC1	ПК	Графічний дизайнер	1
	PC2	ПК	Веб-дизайнер	1
	PC3	ПК	Ілюстратор	1
	PC4	ПК	Друкар	1
	PC5	ПК	Аніматор	1
	PC6	ПК	Арт-директор	1
Розробки	PC1-PC2	ПК	Веб-розробник	2
	PC3	ПК	Фронтенд-розробник	1
	PC4	ПК	Бекенд-розробник	1
	PC5	ПК	Full-stack розробник	1
	PC6	ПК	QA-інженер	1
	PC7	ПК	UI/UX-дизайнер	1
	Printer0	Принтер	-	1
Продажу	PC1-PC3	ПК	Менеджер з продажу	3
	PC4-PC5	ПК	Аналітик з продажу	2
	PC6	ПК	Клієнтський менеджер	1
	PC7	ПК	Ключовий менеджер з продажу	1
	Server HTTP	Server	-	1
	Server DNS	Server	-	1
	Server RADIUS	Server	-	1
	Printer1	Принтер	-	1

Продовження таблиці 3.1

Проектного керування	PC1 VLAN 25	ПК	Проектний менеджер	1
	PC2 VLAN 35	ПК	Бізнес-аналітик	1
	PC3-PC4 VLAN 45	ПК	Тестувальник	2
	PC5 VLAN 35	ПК	Координатор проєкту	1
	PC6 VLAN 45	ПК	Технічний архітектор	1
	Server TFTP	Сервер	-	1
Підтримки й обслуговування	PC1-PC2	ПК	Технічний спеціаліст підтримки	2
	PC3	ПК	Інженер з контролю якості	1
	PC4	ПК	Спеціаліст з безпеки	1
	PC5	ПК	Веб-аналітик	1
	PC6	ПК	Спеціаліст з SEO	1
	PC7	ПК	Веб-дизайнер(підтримка)	1
	Printer2	Принтер	-	1
В підсумку: 33 PC, 3 принтерів, 4 сервери   40 пристроїв				

Схема фізичної топології мережі зображена на рисунку 3.4

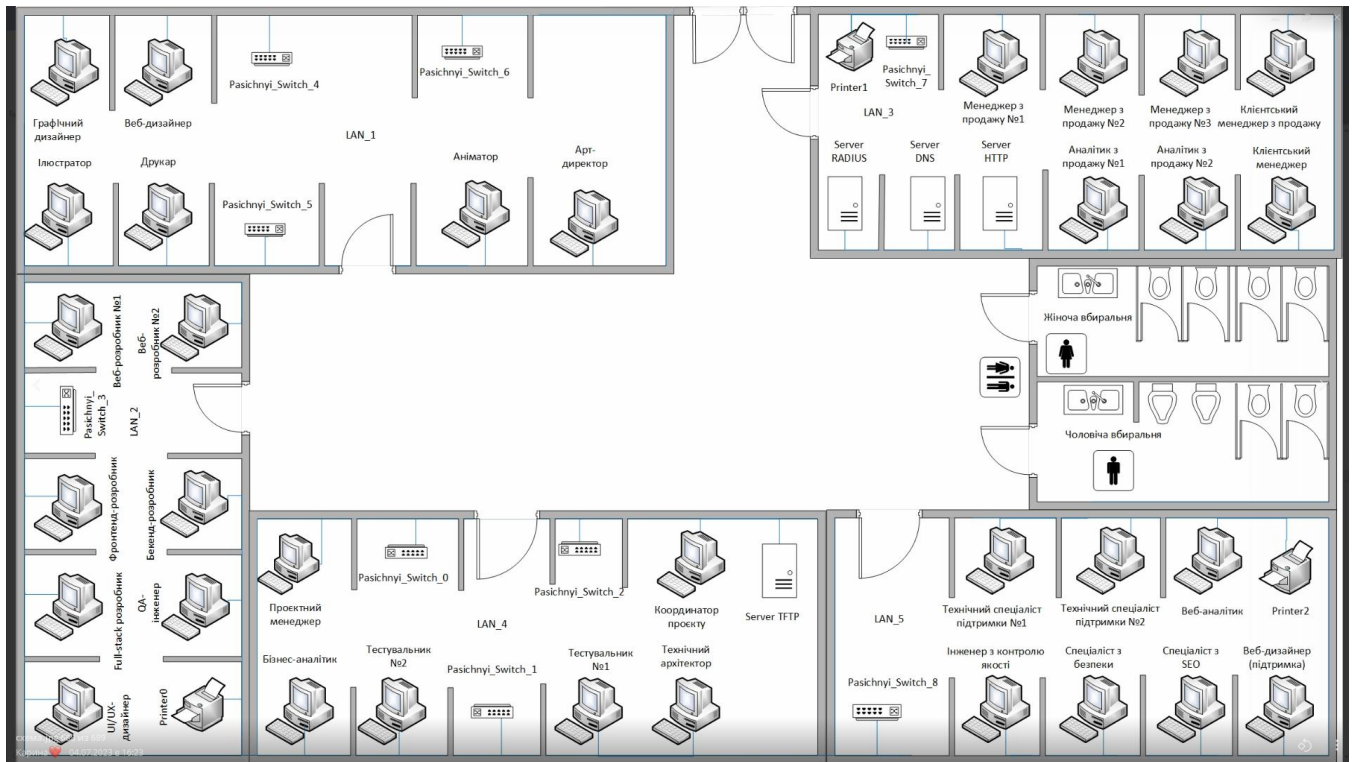


Рисунок 3.4 – Схема фізичної топології мережі

### 3.6 Характеристики використаних апаратних засобів

Ключовим етапом є вибір "інтелектуальних" пристроїв, таких як маршрутизатори, комутатори, мережеві адаптери і принт-сервери. Ці пристрої відносять до активної апаратури, оскільки вони не лише передають сигнал, але й здатні виконувати інші функції. З іншого боку, пасивна апаратура включає різні типи кабелів (оптоволоконні, мідні), розетки, конектори, сплітери і так далі. Ці пристрої є необхідними для створення єдиної мережі, яка забезпечує передачу сигналу.

Для функціональності мережі компанії з інтернет-продажів важливою є кожна складова, оскільки вона забезпечує безперебійну роботу, високу швидкість передачі даних, масштабованість та безпеку. У корпоративній мережі ключову роль відіграють комутатори та маршрутизатори, які відповідають за з'єднання всіх компонентів системи, керування трафіком та підтримку необхідної продуктивності.

Комутатор CBS250-24T-4G є основою для об'єднання всіх кінцевих пристроїв мережі у єдину інфраструктуру. Завдяки підтримці Gigabit Ethernet та

можливості роботи з 10 Gigabit Ethernet каналами, цей комутатор забезпечує високу швидкість передачі даних, що є критично важливим для компанії, яка працює з великими обсягами інформації в реальному часі. Надійність і гнучкість комутатора дозволяють підключати нові робочі станції або розширювати мережу без збоїв. Завдяки розширеним функціям, таким як якість обслуговування (QoS) та статична маршрутизація, система ефективно управляє пріоритетами трафіку, що забезпечує стабільну роботу критичних бізнес-додатків, наприклад, онлайн-магазину та баз даних клієнтів. Зручний веб-інтерфейс значно спрощує процес налаштування й підтримки, а можливість Power over Ethernet Plus (PoE+) дозволяє живити пристрої, зменшуючи потребу у додаткових джерелах живлення.

Маршрутизатор Cisco 2911 відіграє ключову роль у забезпеченні зв'язку як всередині компанії, так і з зовнішніми ресурсами. Ця модель надає надійну маршрутизацію трафіку між різними сегментами мережі, зокрема між головним офісом, філіями та Інтернетом. Завдяки підтримці VPN і тунелювання маршрутизатор створює захищені канали передачі даних для віддалених підключень, що є надзвичайно важливим для забезпечення конфіденційності та безпеки клієнтської інформації. Наявність PoE на портах Ethernet дозволяє живити мережеві пристрої, що додатково підвищує ефективність використання обладнання.

Обидва компоненти працюють у тісній взаємодії, створюючи основу для стабільної та безпечної мережі. Висока продуктивність, масштабованість та надійність комутатора і маршрутизатора забезпечують швидке функціонування бізнес-процесів компанії, таких як обробка замовлень, обслуговування клієнтів, робота онлайн-сервісів та внутрішніх систем управління даними.

Для забезпечення ефективної роботи відділів компанії з інтернет-продажів важливо використовувати комп'ютери, які відповідають специфічним завданням кожного підрозділу, а також забезпечують продуктивність, надійність та швидку обробку даних.

Відділ дизайну потребує високої продуктивності для роботи зі складними графічними додатками та програмами для обробки зображень і відео. Для цього

оптимальним рішенням є комп'ютер на базі Intel i5 9400F з GTX 1070 на 8 ГБ відеопам'яті, 16 ГБ оперативної пам'яті та комбінацією SSD на 250 ГБ і жорсткого диска на 1 ТБ. Ця конфігурація дозволяє працювати з великими файлами, здійснювати рендеринг графіки та обробляти відео без затримок. Швидкий SSD забезпечує оперативне завантаження програм і системи, а HDD на 1 ТБ дозволяє зберігати значний обсяг робочих матеріалів.

Відділу розробки, що займається програмуванням та тестуванням систем, також необхідний потужний ПК. Використання системи з Intel i5, великою кількістю оперативної пам'яті та графічною картою забезпечує можливість запуску віртуальних машин, тестових середовищ і розробницьких додатків одночасно, що значно підвищує продуктивність роботи команди.

Для відділів продажів та підтримки й обслуговування, де основна робота зосереджена на обробці документів, електронному листуванні та взаємодії з базами даних, достатньо використання Artline Business B27 на базі Intel Core i3-10100 з 8 ГБ оперативної пам'яті та SSD на 240 ГБ. Ця модель забезпечує стабільну та швидку роботу офісних програм і систем управління клієнтською інформацією. Завдяки вбудованій графіці Intel UHD Graphics 630, комп'ютер ефективно справляється із завданнями, що не вимагають додаткової обчислювальної потужності. Швидкий SSD-накопичувач гарантує оперативне завантаження програм і мінімізує час простою, що особливо важливо для відділу продажів, де швидкість обробки запитів впливає на результативність.

Таким чином, розподіл обчислювальних ресурсів відповідає конкретним потребам кожного відділу: продуктивніші ПК використовуються у відділах, де необхідна обробка складної графіки чи програмування, а більш доступні й економічні рішення ідеально підходять для офісних завдань у відділах продажів та підтримки клієнтів[4].

Таблиця 3.2 – Специфікація обладнання

Позиція	Найменування і технічна характеристика	Тип, марка, позначення документа, опитувального листа	Од. вим.	Кільк.
1	Cisco2911 WAASX/K9 3 x 10Base-T/100Base-TX/1000Base-T - RJ-45, 2 x EHWIC, 1 x SM	Pasichnyi_Router_0 Pasichnyi_Router_1 Pasichnyi_Router_2 Pasichnyi_Router_3_DHCP Pasichnyi_Router_4	Шт.	5
2	CISCO CBS250-24T-4G 4 x SFP, 24 x Gigabit Ethernet(10/100/1000)	Pasichnyi_Switch_0 Pasichnyi_Switch_1 Pasichnyi_Switch_2 Pasichnyi_Switch_3 Pasichnyi_Switch_4 Pasichnyi_Switch_5 Pasichnyi_Switch_6 Pasichnyi_Switch_7 Pasichnyi_Switch_8	Шт.	20
3	Busk Intel i5 9400F / GTX 1070 8GB / 16GB / SSD 250GB + HDD 1TB	PC	Шт.	240
	Artline Business B27 v34 Intel Core i3-10100 (3.6 – 4.3 ГГц) / RAM 8 ГБ / SSD 240 ГБ / Intel UHD Graphics 630			90

Кінець таблиці 3.2

4	ARTLINE Business T15 v15 Intel Core i3-10100/ 16 ГБ/ SSD: 250 ГБ/ HDD: 2 x 1 ТБ/	Server HTTP Server DNS Server TFTP Server RADIUS	Шт.	4
5	Canon Pixma TS3340 1200x4800 dpi/ PG-445/ CL-446/ 10 x 15 см без полів: прибл. 65 сек	Printer_0 Printer_1 Printer_2	Шт.	10



## 4 РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 4.1 Призначення й сфера застосування програми

Система моніторингу та локальний вікі-сайт розроблені для підвищення ефективності управління інфраструктурою інтернет-магазину. Застосування цих рішень спрямоване на спрощення операційних процесів, підвищення швидкості вирішення проблем та зниження ризиків, пов'язаних з перебоями у роботі мережі та сервісів.

Програми моніторингу призначені для виявлення аномалій та проблем з продуктивністю, а також для забезпечення відповідності вимогам під час їх роботи. Вони аналізують логи та інші дані, що генеруються під час роботи системи, що дозволяє швидко реагувати на помилки та складнощі. Крім того, вони дозволяють перевіряти, чи відповідає програмна система вимогам, що є важливим для складних та гетерогенних систем.

Локальний вікі-сайт у компанії слугує інструментом для обміну знаннями та підтримки неформального навчання серед працівників. Він дозволяє співробітникам створювати та редагувати сторінки, що сприяє колективному підходу до управління знаннями.

Сфера застосування системи моніторингу. Моніторинг є особливо важливим для критичних систем, де надійність є ключовим фактором. Використовуються різні техніки, такі як логування подій та перевірка тверджень, для забезпечення надійності.

Моніторинг може бути використаний для відстеження процесу розробки програмного забезпечення, що дозволяє порівнювати різні проекти та виявляти загальні закономірності в їх розвитку.

Сфера застосування локального вікі-сайту. Вікі-сайти використовуються для управління знаннями. Вони допомагають у створенні та поширенні знань, а також у вирішенні складних проблем у віртуальних командах.

## 4.2 Обґрунтування технічних характеристик програми

Основною метою розробки програми є створення системи моніторингу інфраструктури та локального вікі-сайту для ефективного обміну знаннями серед працівників. Завдання програми моніторингу полягає у своєчасному виявленні аномалій у роботі серверів та застосунків, автоматичному сповіщенні про неполадки та забезпеченні безперебійної роботи критичних систем. Локальний вікі-сайт має функціонувати як інтерактивна платформа для створення, зберігання та редагування знань, що дозволяє підвищити ефективність роботи співробітників компанії.

У системі моніторингу для інфраструктури використовуються кілька математичних методів для оцінки стану мережевих пристроїв, виявлення аномалій та прогнозування потенційних проблем на основі статистичних даних.

Моніторинг і виявлення аномалій. Для моніторингу стану пристроїв на мережі, програма визначає статус кожного пристрою (онлайн чи офлайн). З кожним циклом моніторингу (кожні 5 секунд), випадковим чином генерується подія, де ймовірність того, що пристрій вийде з ладу, становить 10%. Це здійснюється за допомогою порівняння випадкового числа з пороговим значенням (0.1). Якщо випадкове число менше за 0.1, статус пристрою змінюється на "Offline", що вказує на можливу проблему в мережі.

Візуалізація та аналіз даних. Для графічного відображення даних використовується бібліотека `matplotlib`, що дозволяє будувати графіки для аналізу різних параметрів, таких як використання процесора. Дані про використання процесора та інші параметри змінюються випадковим чином, а потім ці зміни візуалізуються у вигляді лінійного графіку, що дозволяє спостерігати за змінами в часі. Алгоритм генерує випадкові значення для CPU-індексу і побудовує графік змін в часі.

Алгоритм оновлення даних. Параметри, що відображають стан пристроїв, оновлюються кожні кілька секунд у циклі, забезпечуючи постійне оновлення даних у реальному часі. Це забезпечує безперервне спостереження за станом

мережі, що є важливим для виявлення аномалій і оперативного реагування на зміни.

### 4.3 Опис розробленої програми

Програма складається з двох основних компонентів: системи моніторингу мережі та локального вікі-сайту. Система моніторингу забезпечує реальний моніторинг стану мережевих пристроїв в режимі реального часу, відображаючи статус пристроїв, їх трафік та повідомляючи про виникнення проблем або аномалій. Локальний вікі-сайт служить для внутрішнього обміну знаннями серед співробітників компанії, надаючи доступ до документації, новин, задач і навчальних матеріалів.

Опис логічної структури:

а) Модуль моніторингу мережі:

- 1) Має головне вікно з таблицею пристроїв, що оновлюється кожні 5 секунд.
- 2) Кожен пристрій має свої атрибути, такі як ім'я, IP-адреса, статус і трафік.
- 3) Має можливість перевірки стану кожного пристрою та оновлення їх статусу (онлайн/офлайн) у реальному часі.
- 4) Програма генерує випадкові дані для тестування та моніторингу, що дозволяє симулювати різні сценарії.

б) Модуль локального вікі-сайту:

- 1) Містить просту HTML-структуру з кількома основними сторінками (Головна, Новини, Документація, Задачі, Навчання).
- 2) Забезпечує доступ до різних ресурсів і документів компанії, які можуть бути переглянуті та змінені співробітниками.

Використовувані технічні засоби:

- Мова програмування: Python для системи моніторингу (з бібліотеками tkinter, customtkinter, matplotlib для графіки та взаємодії з користувачем) і HTML/CSS для вікі-сайту[5].

– Бібліотеки: `tkinter` та `customtkinter` для створення графічного інтерфейсу користувача. `matplotlib` для візуалізації статистики. `random` для генерації випадкових даних. `threading` для запуску багатозадачності у системі моніторингу.

– Система управління контентом: стандартний набір HTML, CSS для створення інтерфейсу локального вікі-сайту, що дозволяє зберігати та відображати документи[6].

### 4.3.1 Опис логічної структури програми

Логічна структура програми моніторингу мережі описує основні компоненти програми, їх взаємодію та функціональну організацію. Програма має модульну структуру, яка включає декілька основних частин, що виконують конкретні завдання.

Графічний інтерфейс користувача побудований за допомогою модуля `Tkinter`, який дозволяє користувачу взаємодіяти з програмою. Всі елементи інтерфейсу (кнопки, таблиці, вікна) побудовані за допомогою бібліотеки `Tkinter`, з використанням додаткових можливостей `CustomTkinter` для покращеного вигляду.

Користувач може переглядати статуси пристроїв, їх трафік, а також взаємодіяти з системою через кнопки оновлення даних і тестування сповіщень.

Графічний інтерфейс зображено на рисунках 4.1 – 4.2.

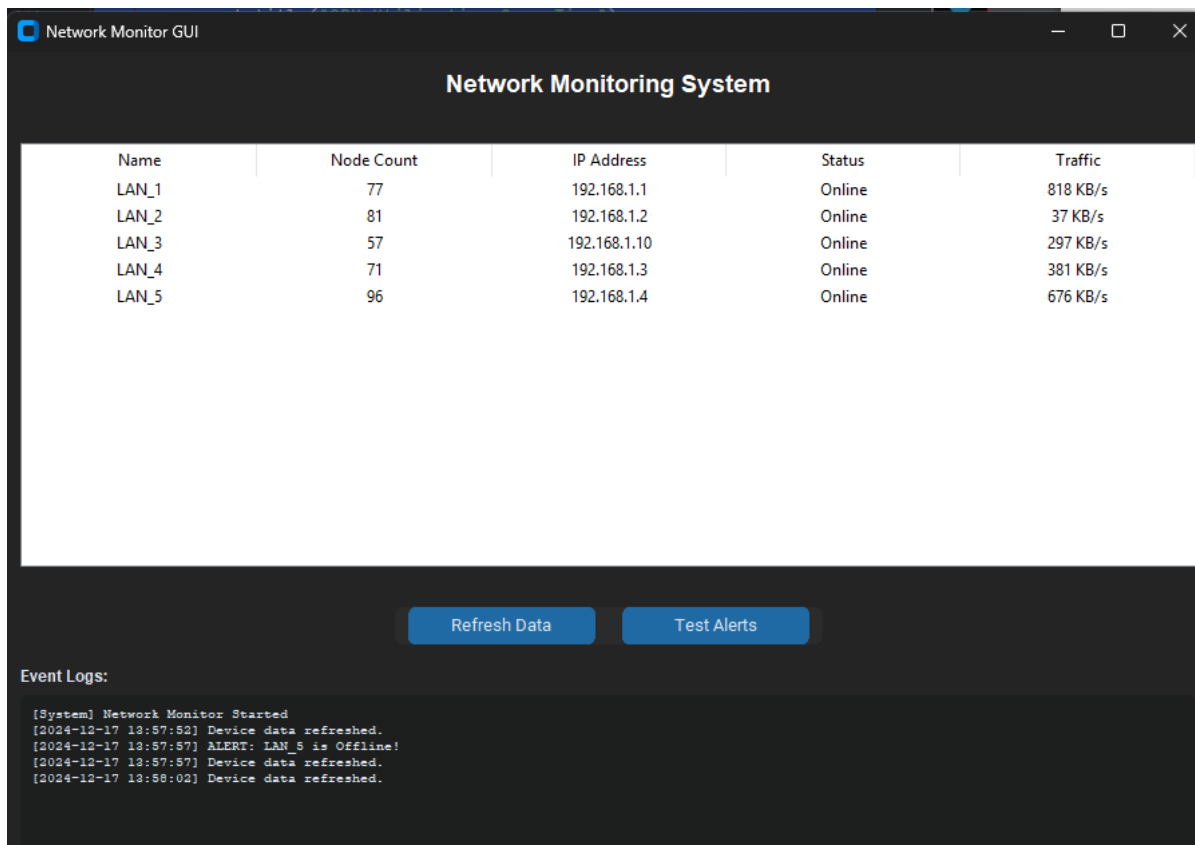


Рисунок 4.1 – Головне вікно системи моніторингу

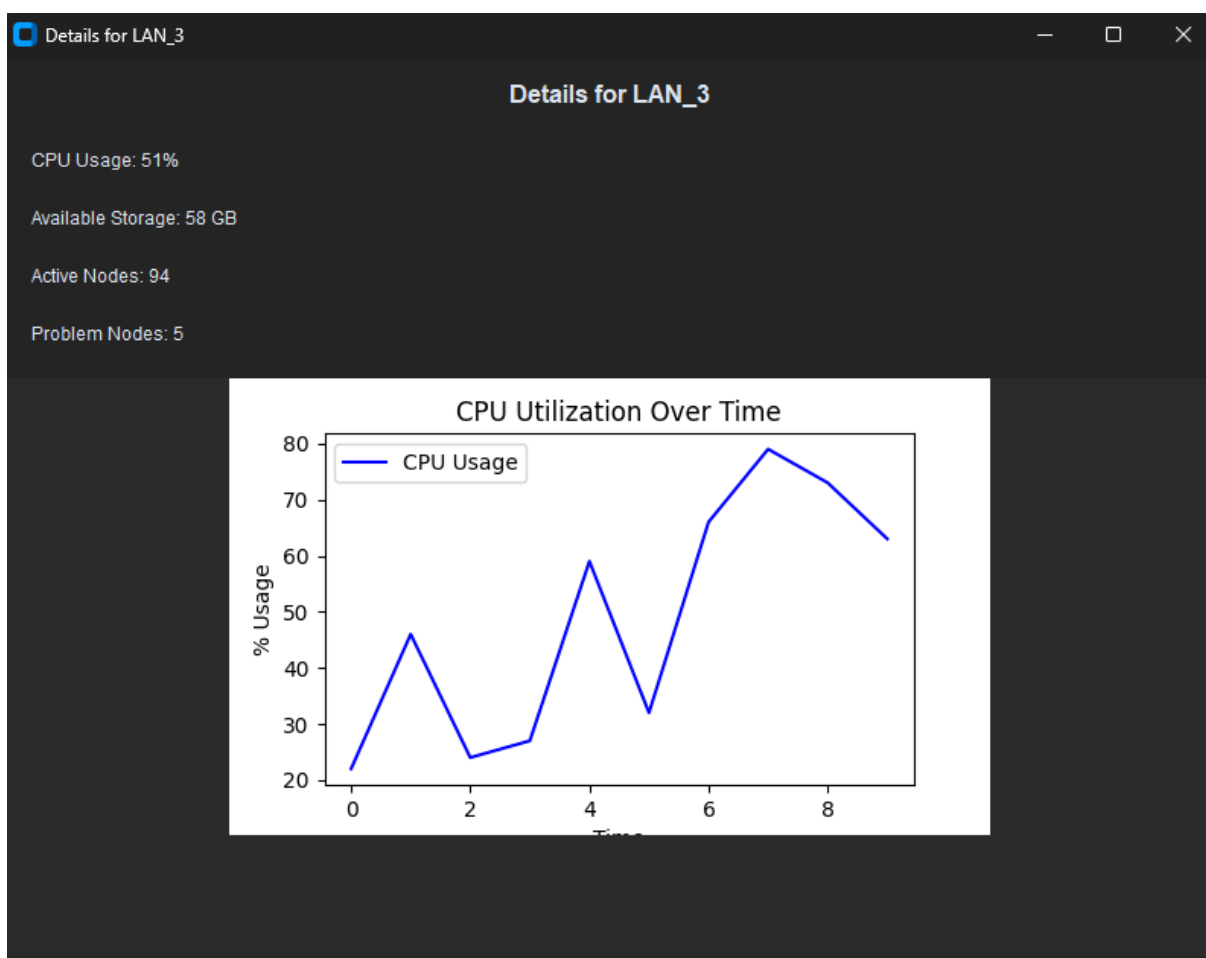


Рисунок 4.2 – Інтерфейс вікна підмереж

Логічна структура програми для вікі-сайту організована таким чином, щоб забезпечити зручне управління сторінками, можливість їх перегляду, додавання нових сторінок та їх редагування. Вона включає в себе кілька основних компонентів, які забезпечують функціональність веб-застосунку. Усі ці компоненти взаємодіють один з одним для забезпечення ефективної роботи сайту.

Структура файлів сайту зображена на рисунку 4.3.

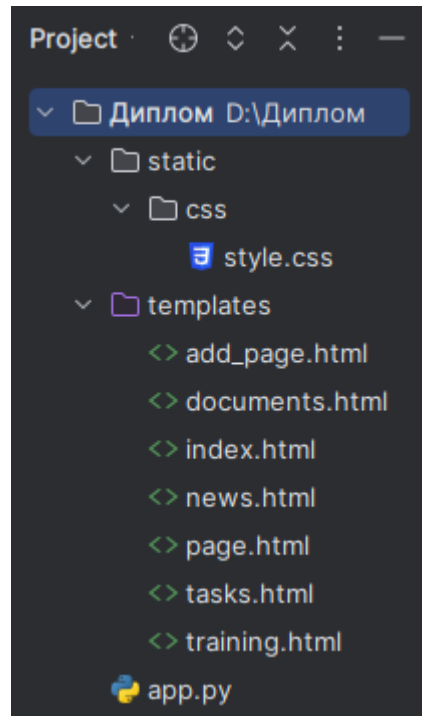


Рисунок 4.3 – Структура файлів сайту

Графічний інтерфейс користувача побудований за допомогою шаблонів HTML, CSS. Інтерфейс складається з кількох сторінок. Веб-сторінки генеруються за допомогою Flask та Jinja2, де кожна сторінка може бути інтерпретована сервером і відображена користувачеві.

Інтерфейс включає навігаційне меню для переходу між основними сторінками сайту, а також форму для додавання нових сторінок до вікі.

Графічний інтерфейс зображено на рисунках 4.4 – 4.5.

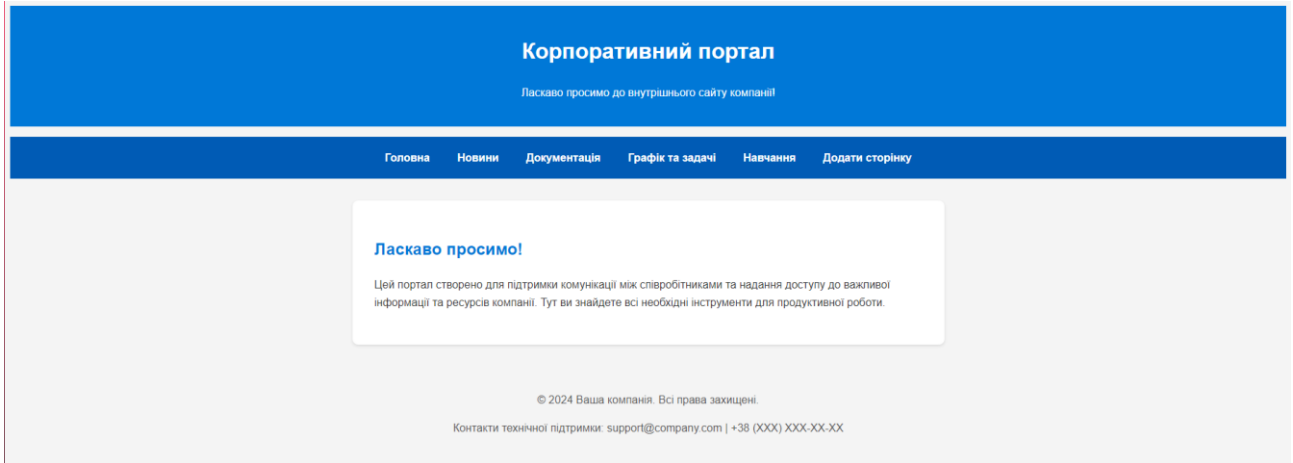


Рисунок 4.4 – Головна сторінка сайту

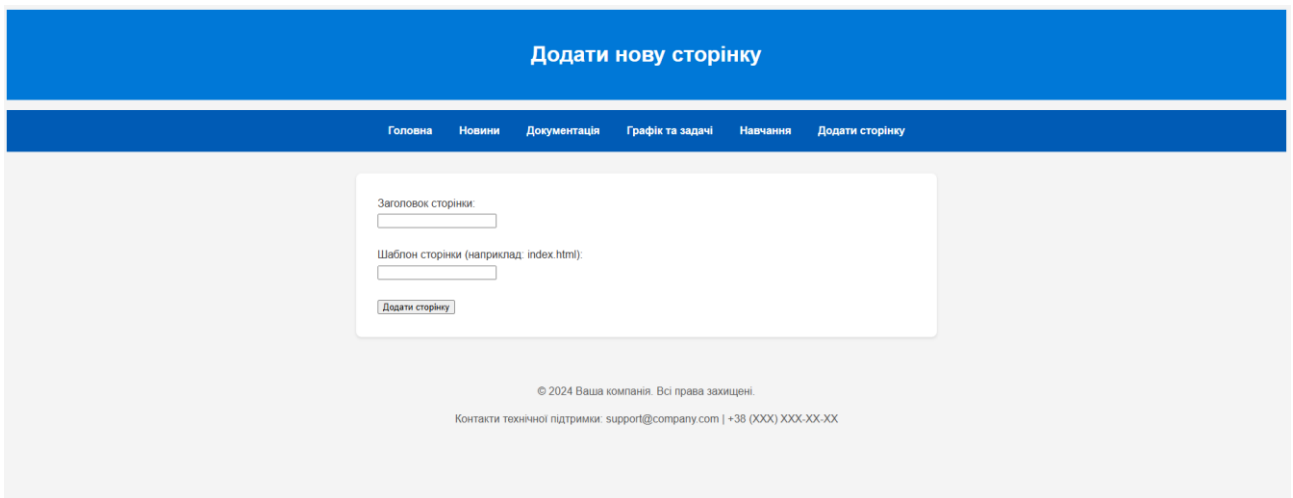


Рисунок 4.5 – Сторінка додавання нових сторінок

## 4.4 Загальні відомості

а) Позначення і найменування програми:

- 1) Програма: "Система моніторингу мережі"
- 2) Позначення: NetworkMonitor

б) Програмне забезпечення, необхідне для функціонування програми:

- 1) Python: Основна мова програмування для реалізації програми.
- 2) Tkinter: Бібліотека для створення графічного інтерфейсу користувача.
- 3) CustomTkinter: Модуль для поліпшеного дизайну віджетів в Tkinter.

4) Matplotlib: Бібліотека для побудови графіків та діаграм.

5) Threading: Стандартна бібліотека Python для створення багатозадачності.

6) Datetime: Модуль для роботи з датою і часом, необхідний для логування подій.

в) Мови програмування, на яких написана програма:

1) Python: Використовується для реалізації основної логіки програми, роботи з бібліотеками та обробки даних.

а) Позначення і найменування програми:

1) Програма: "Вікі-сайт корпоративного порталу"

2) Позначення: WikiApp

б) Програмне забезпечення, необхідне для функціонування програми:

1) Python: Основна мова програмування для розробки програми.

2) Flask: Веб-фреймворк, використовуваний для розробки веб-застосунку[7].

3) Jinja2: Шаблонізатор, що входить до складу Flask для рендерингу HTML-шаблонів.

4) Werkzeug: Бібліотека для роботи з HTTP-запитами та іншими веб-технологіями, також частина Flask.

5) Браузер: Для перегляду веб-сторінок та взаємодії з користувачем.

в) Мови програмування, на яких написана програма:

1) Python: Мова програмування, на якій написаний серверний код програми.

2) HTML: Мова розмітки для створення сторінок сайту.

3) CSS: Мова для стилізації HTML-елементів, що забезпечує вигляд веб-сторінок.

#### **4.5 Виклик і завантаження**

Система моніторингу мережі: програма запускається через головний Python-скрипт. Після запуску користувач отримує графічний інтерфейс, що



відображає статус пристроїв та дозволяє здійснювати тестування системи, оновлювати дані та переглядати лог-файли.

Локальний вікі-сайт: для запуску локального вікі-сайту використовується Python-бібліотека Flask. Вікі-сайт запускається через запуск Flask-серверу, який обслуговує різні сторінки сайту. Сервер приймає запити на головну сторінку та сторінки окремих розділів, а також підтримує додавання нових сторінок через форму.

## 5 ЕКСПЕРЕМЕНТАЛЬНИЙ РОЗДІЛ

### 5.1 Розробка математичної моделі мережі як замкнутої системи масового обслуговування

Відповідно до архітектури комп'ютерної мережі та імітаційної моделі розроблено математичну модель комп'ютерної мережі як закритої колективної системи обслуговування, яка зображена на рисунку 5.1.

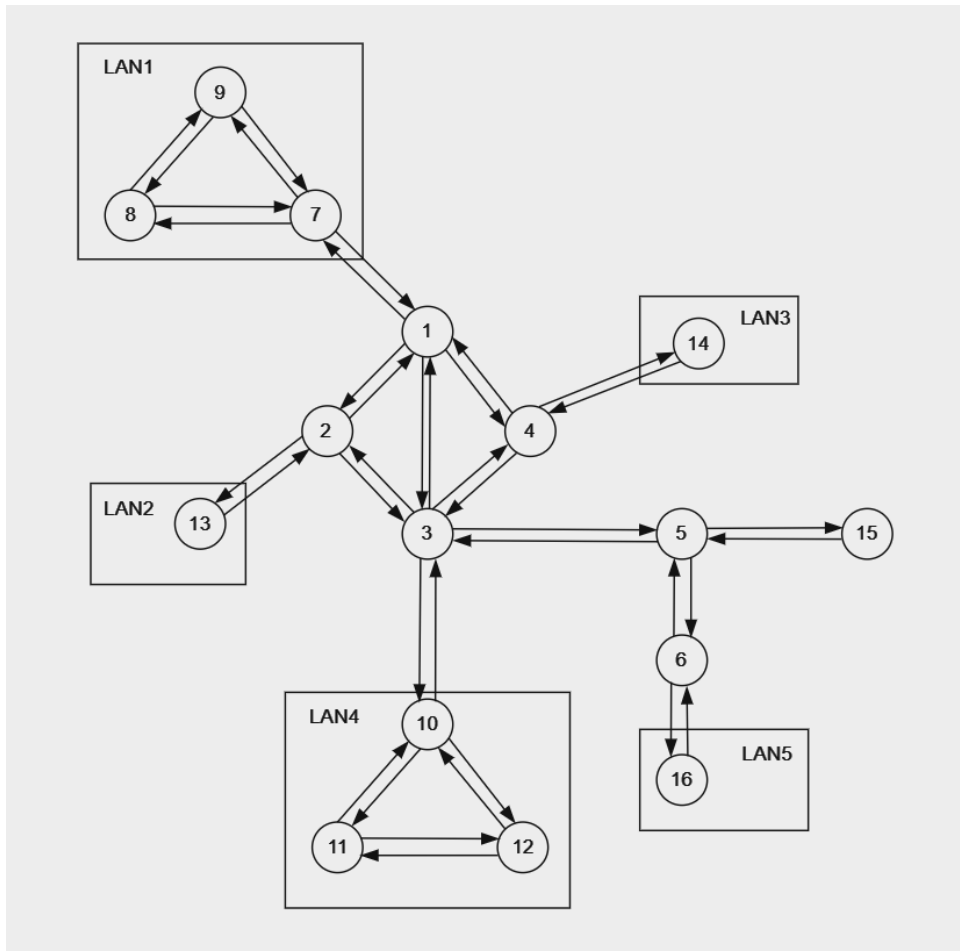


Рисунок 5.1 – Конструкція математичної моделі комп'ютерної мережі

В конструкції математичної моделі комп'ютерної мережі вузли 7, 8, 9, 10, 11, 12, 13, 14, 15, 16 виконують функції комутаторів, які обслуговують локальні мережі. Вузли 1, 2, 3, 4, 5, 6 є маршрутизаторами.

Зв'язки між елементами структури визначаються ймовірністю передачі пакета від одного вузла до іншого. Кожен вузол є системою масового обслуговування. Ймовірність з'єднання вузла з самим собою дорівнює нулю. Ймовірність з'єднання вузлів між собою задається матрицею маршрутів.

$$Pr := \begin{pmatrix} 0 & 0.2 & 0.4 & 0.2 & 0 & 0 & 0.2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.2 & 0 & 0.6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2 & 0 & 0 \\ 0.15 & 0.15 & 0 & 0.15 & 0.4 & 0 & 0 & 0 & 0 & 0.15 & 0 & 0 & 0 & 0 & 0 \\ 0.2 & 0 & 0.6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2 & 0 \\ 0 & 0 & 0.3 & 0 & 0 & 0.4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.3 \\ 0 & 0 & 0 & 0 & 0.7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.3 \\ \hline 0.6 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2 & 0.2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.6 & 0 & 0.4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.6 & 0.4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2 & 0.2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.6 & 0 & 0.4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.6 & 0.4 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Рисунок 5.2 – Маршрутна матриця

Стовпці матриці показують час обробки одного повідомлення у відповідному вузлі:

$$\tau_i :=$$

$20 \cdot 10^{-6}$
$20 \cdot 10^{-6}$
$20 \cdot 10^{-6}$
$20 \cdot 10^{-6}$
$20 \cdot 10^{-6}$
$20 \cdot 10^{-6}$
$20 \cdot 10^{-6}$
$20 \cdot 10^{-6}$
$20 \cdot 10^{-6}$
$20 \cdot 10^{-6}$
$20 \cdot 10^{-6}$
$20 \cdot 10^{-6}$
$20 \cdot 10^{-6}$
$20 \cdot 10^{-6}$
$20 \cdot 10^{-6}$
$20 \cdot 10^{-6}$
$20 \cdot 10^{-6}$
$20 \cdot 10^{-6}$
$20 \cdot 10^{-6}$

Рисунок 5.3 – Час обробки одного повідомлення на вузлі

## 5.2 Розрахунок параметрів мережі по її моделі

Потім за допомогою методу Гауса обчислюється матриця стовпчиків коефіцієнтів передачі:

$$e_m = \begin{pmatrix} 1 \\ 0.833 \\ 3.111 \\ 0.833 \\ 4.148 \\ 2.37 \\ 0.333 \\ 0.111 \\ 0.111 \\ 0.778 \\ 0.259 \\ 0.259 \\ 0.167 \\ 0.167 \\ 1.244 \\ 0.711 \end{pmatrix}$$

Рисунок 5.4 – Розрахунок передаточних коефіцієнтів

Елемент вказує на кількість конвеєрів обробки пакетів у кожному вузлі системи масового обслуговування. Для цілей підрахунку передбачається, що кожен пристрій містить лише один конвеєр обробки пакетів.

-----	
	0
0	1
1	1
2	1
3	1
4	1
5	1
6	1
m = 7	1
8	1
9	1
10	1
11	1
12	1
13	1
14	1
15	1

Рисунок 5.5 – Кількість конвеєрів

Матриця В визначає ймовірність того, що у вузлі (позначеному рядком) є пакети, які очікують на обробку (позначені стовпчиком).

	0	1	2	3	4
0	0.833	0.143	0.021	$2.36 \cdot 10^{-3}$	$1.529 \cdot 10^{-4}$
1	0.861	0.123	0.015	$1.379 \cdot 10^{-3}$	$7.363 \cdot 10^{-5}$
2	0.481	0.291	0.152	0.061	0.014
3	0.861	0.123	0.015	$1.379 \cdot 10^{-3}$	$7.363 \cdot 10^{-5}$
4	0.308	0.287	0.226	0.134	0.045
5	0.605	0.263	0.099	0.029	$4.824 \cdot 10^{-3}$
6	0.944	0.053	$2.518 \cdot 10^{-3}$	$9.092 \cdot 10^{-5}$	$1.88 \cdot 10^{-6}$
7	0.981	0.018	$2.867 \cdot 10^{-4}$	$3.414 \cdot 10^{-6}$	$2.321 \cdot 10^{-8}$
8	0.981	0.018	$2.867 \cdot 10^{-4}$	$3.414 \cdot 10^{-6}$	$2.321 \cdot 10^{-8}$
9	0.87	0.116	0.013	$1.127 \cdot 10^{-3}$	$5.602 \cdot 10^{-5}$
10	0.957	0.042	$1.536 \cdot 10^{-3}$	$4.298 \cdot 10^{-5}$	$6.881 \cdot 10^{-7}$
11	0.957	0.042	$1.536 \cdot 10^{-3}$	$4.298 \cdot 10^{-5}$	$6.881 \cdot 10^{-7}$
12	0.972	0.027	$6.45 \cdot 10^{-4}$	$1.159 \cdot 10^{-5}$	$1.189 \cdot 10^{-7}$
13	0.972	0.027	$6.45 \cdot 10^{-4}$	$1.159 \cdot 10^{-5}$	$1.189 \cdot 10^{-7}$
14	0.792	0.171	0.032	$4.472 \cdot 10^{-3}$	$3.662 \cdot 10^{-4}$
15	0.881	0.119	0.012	$9.033 \cdot 10^{-4}$	$3.908 \cdot 10^{-5}$

Рисунок 5.6 – Ймовірність очікування пакетів

Середнє значення розраховується для кожного вузла мережі за булевим алгоритмом.

Інтенсивність вхідних потоків пакетів на кожному вузлі:

	0
0	$8.194 \cdot 10^3$
1	$6.826 \cdot 10^3$
2	$2.549 \cdot 10^4$
3	$6.826 \cdot 10^3$
4	$3.399 \cdot 10^4$
5	$1.942 \cdot 10^4$
6	$2.729 \cdot 10^3$
7	909.564
8	909.564
9	$6.375 \cdot 10^3$
10	$2.122 \cdot 10^3$
11	$2.122 \cdot 10^3$
12	$1.368 \cdot 10^3$
13	$1.368 \cdot 10^3$
14	$1.019 \cdot 10^4$
15	$5.826 \cdot 10^3$

Рисунок 5.7 – Розрахунок інтенсивності вхідного потоку

Середнє число пакетів що чекають на обробку в кожному вузлі:

	0
0	0.222
1	0.179
2	1.063
3	0.179
4	1.791
5	0.69
6	0.066
L = 7	0.021
8	0.021
9	0.166
10	0.05
11	0.05
12	0.032
13	0.032
14	0.289
15	0.169

Рисунок 5.8 – Розрахунок пакетів, що чекають на обробку

Середній час обробки пакета в вузлі:

	0
0	$2.707 \cdot 10^{-5}$
1	$2.625 \cdot 10^{-5}$
2	$4.168 \cdot 10^{-5}$
3	$2.625 \cdot 10^{-5}$
4	$5.269 \cdot 10^{-5}$
5	$3.553 \cdot 10^{-5}$
6	$2.401 \cdot 10^{-5}$
t = 7	$2.312 \cdot 10^{-5}$
8	$2.312 \cdot 10^{-5}$
9	$2.599 \cdot 10^{-5}$
10	$2.371 \cdot 10^{-5}$
11	$2.371 \cdot 10^{-5}$
12	$2.334 \cdot 10^{-5}$
13	$2.334 \cdot 10^{-5}$
14	$2.835 \cdot 10^{-5}$
15	$2.897 \cdot 10^{-5}$

Рисунок 5.9 – Розрахунок середнього часу обробки пакета для вузлів

### 5.2.1 Параметри роботи мережі в звичайному режимі

Ступінь навантаження каналів у мережі, дорівнює 6. Час обробки пакетів на всіх вузлах однаковий-5 одиниць часу. Кожен вузол мережі має один конвеєр для обробки пакетів. На основі цих вихідних даних було створено графік, який показує середню продуктивність кожного вузла. Графіки зображені на рисунках 5.10 – 5.12.

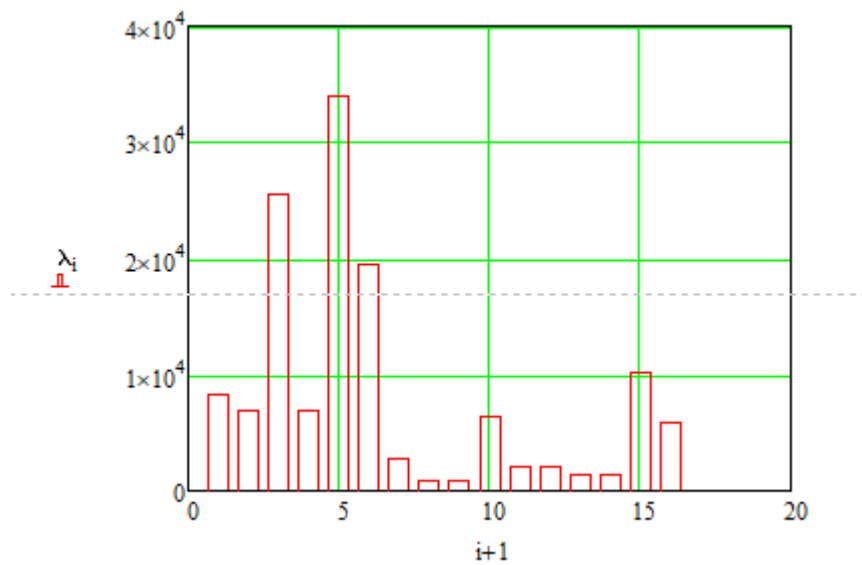


Рисунок 5.10 – Інтенсивність потоку, що входить у вузол

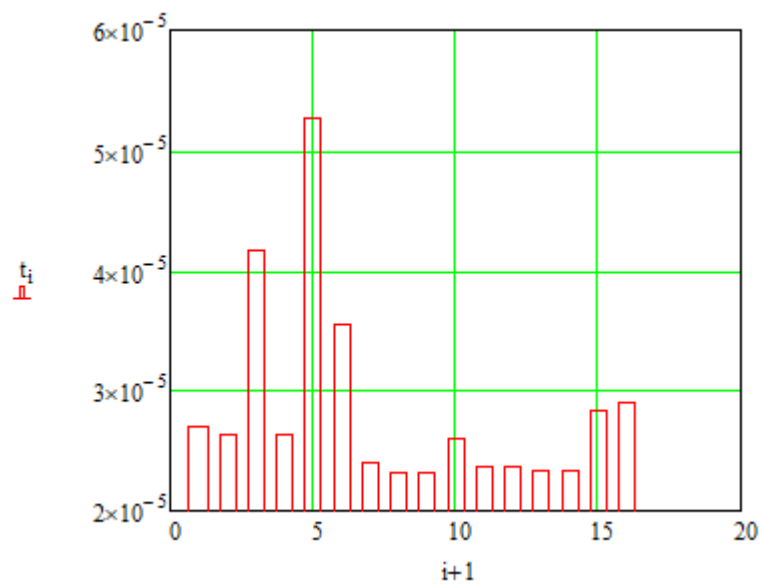


Рисунок 5.11 – Середній час перебування пакета у вузлі

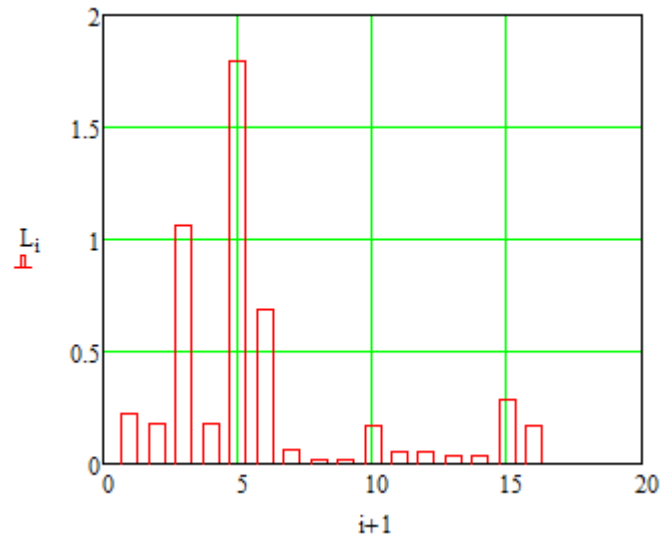
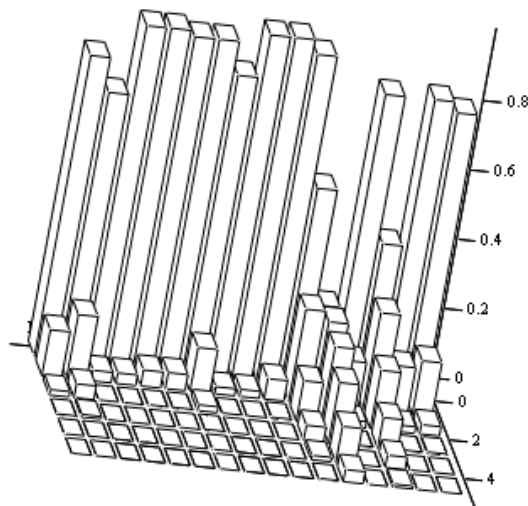


Рисунок 5.12 – Середня кількість пакетів які знаходяться у вузлі

Як бачимо, загалом для всіх вузлів мережі, які є комутаторами, середній показник показує, що всі повідомлення обробляються швидко і без затримок. Виняток становлять вузли 2, 4, 5, 6 і 7, які відповідають за маршрутизацію в мережі.

Рисунок 5.13 показує з якою вірогідністю у вузлах мережі буде черга.



В

Рисунок 5.13 – Вірогідність черги у вузлах мережі

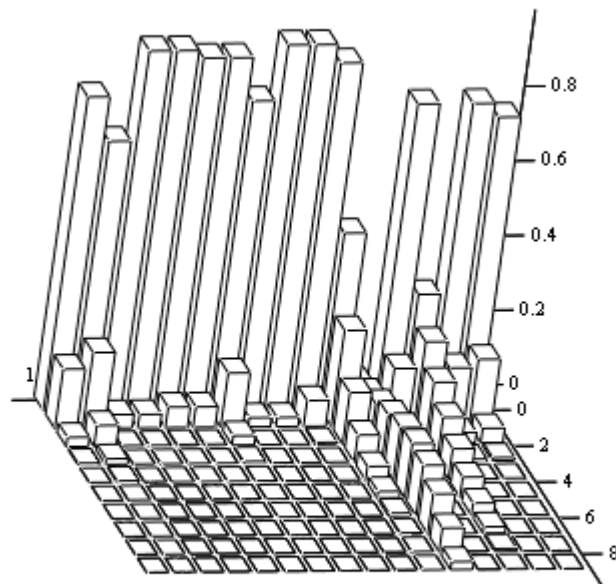


За цими параметрами можна зробити попередній висновок, що найбільш проблемними є мережеві маршрутизатори.

### 5.2.2 Параметри роботи мережі при граничній пропускній здатності

Коли мережа працює на максимальній потужності, комп'ютери використовують значну частину своїх ресурсів для обробки запитів і передачі даних. Навантаження на мережу і обсяг трафіку залежить від характеру виконуваної задачі та інтенсивності обміну даними. Наприклад, при виконанні стандартних операцій трафік зводиться до мінімуму, але при обробці великої кількості запитів або передачі великих обсягів даних навантаження на мережу може досягати критичного рівня, що впливає на стабільність і продуктивність.

Щоб змодельювати цю ситуацію, було встановлено  $N = 10$ . Графіки приведені на рисунках 5.14 – 5.16.



В

Рисунок 5.14 – Імовірність постановки в чергу на вузлах, якщо мережа працює на граничній пропускній здатності

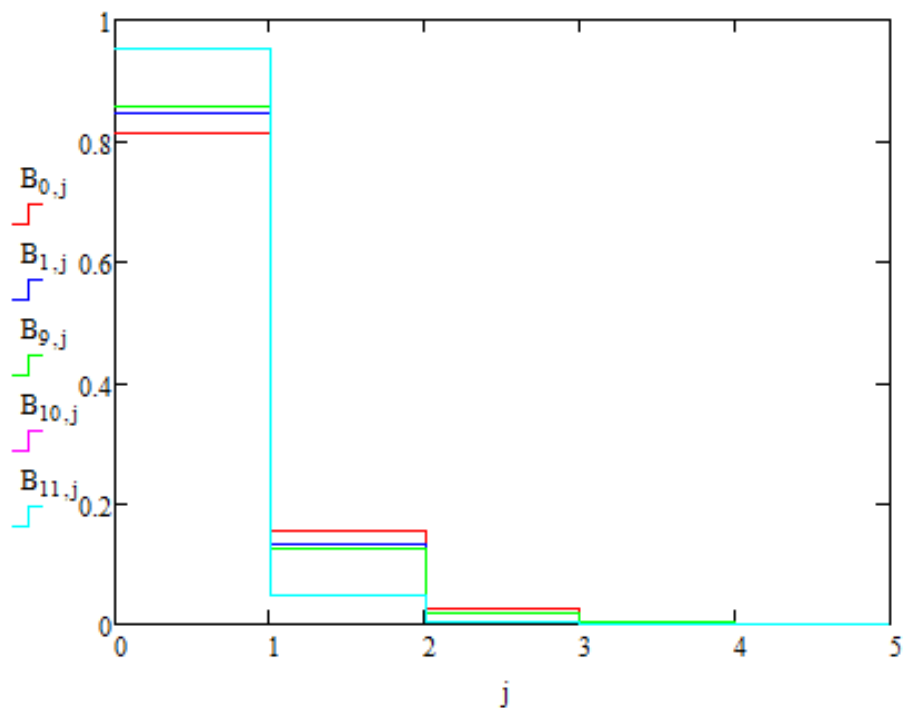


Рисунок 5.15 – Імовірність того, що вузол має чергу, при звичайному режимі роботи

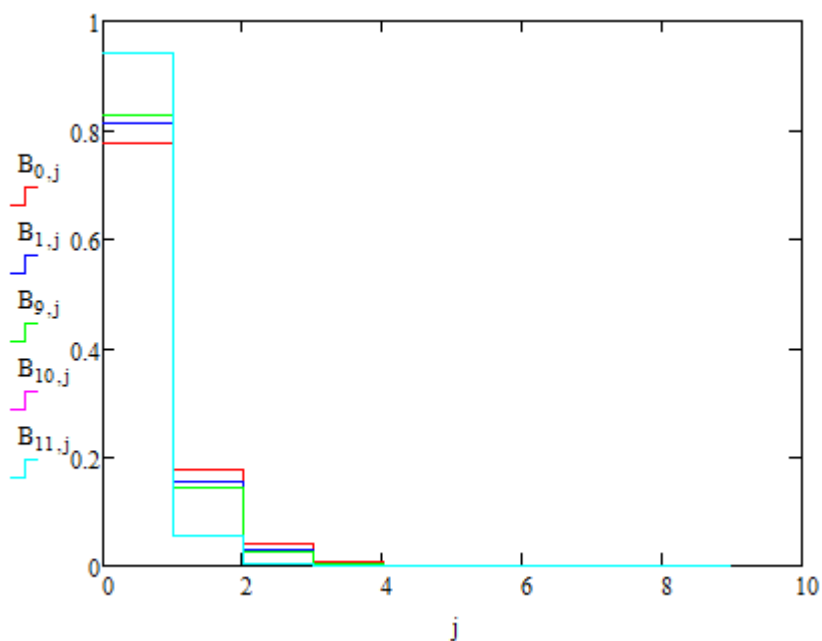


Рисунок 5.16 – Імовірність постановки в чергу на вузлах, при граничній пропускній здатності

Результати моделювання показують, що подвоєння кількості пакетів, які циркулюють у мережі, збільшує ймовірність того, що два пакети опиняться в черзі, але тільки для вузлів на комутаторах №7, 8, 9, 10, 11, 12, 13, 14, 15, 16.

### 5.3 Корекція характеристик вузлів

Корекція характеристик вузлів №3, 5, 6 здійснюється шляхом збільшення швидкості обробки пакетів.

$\tau_i :=$

$20 \cdot 10^{-6}$
$20 \cdot 10^{-6}$
$10 \cdot 10^{-6}$
$20 \cdot 10^{-6}$
$10 \cdot 10^{-6}$
$10 \cdot 10^{-6}$
$20 \cdot 10^{-6}$
$20 \cdot 10^{-6}$
$20 \cdot 10^{-6}$
$20 \cdot 10^{-6}$
$20 \cdot 10^{-6}$
$20 \cdot 10^{-6}$
$20 \cdot 10^{-6}$
$20 \cdot 10^{-6}$
$20 \cdot 10^{-6}$
$20 \cdot 10^{-6}$
$20 \cdot 10^{-6}$
$20 \cdot 10^{-6}$

Рисунок 5.17 – Скореговані значення часу обробки повідомлення

Відповідно до варіацій були розраховані як середні, так і ймовірнісні характеристики. Графіки приведені на рисунках 5.18 – 5.20.

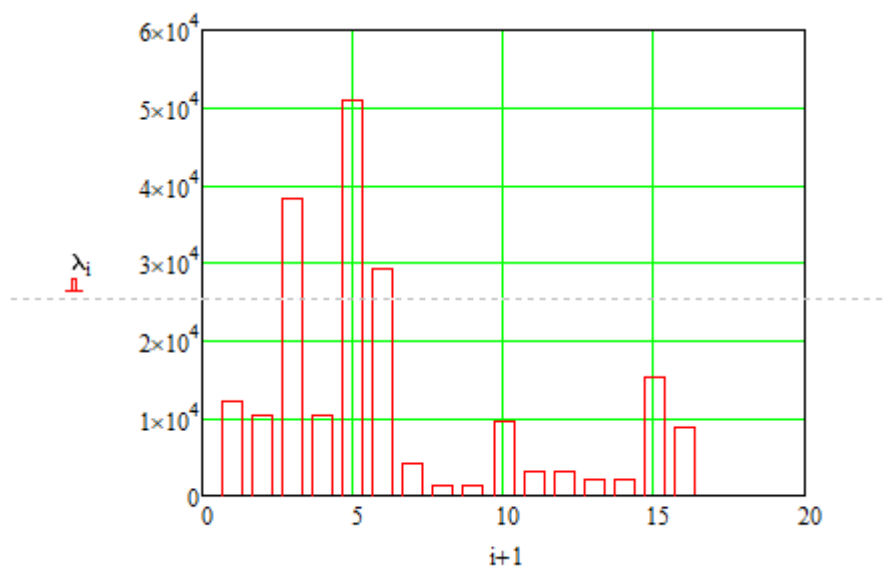


Рисунок 5.18 – Інтенсивність потоку, що надходить у вузол

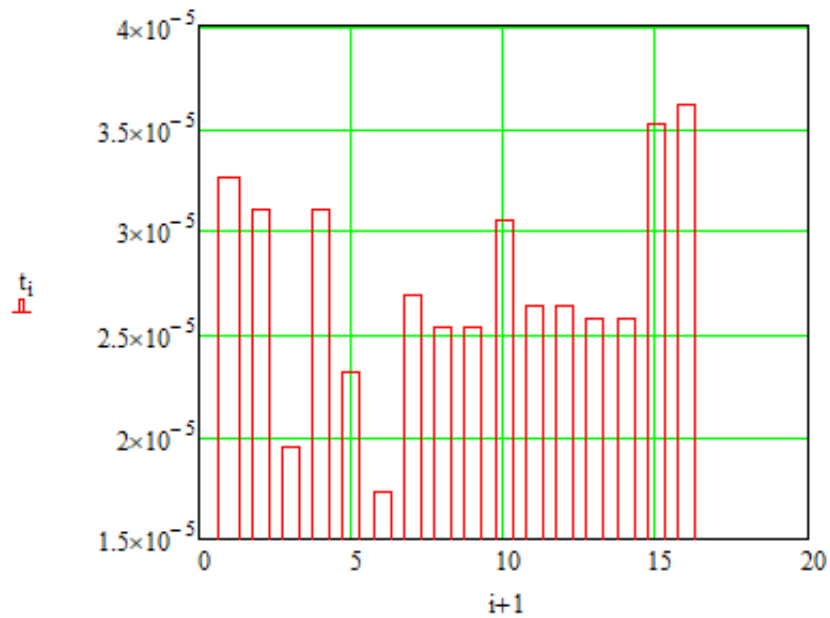


Рисунок 5.19 – Середній час, протягом якого пакет залишається у вузлі

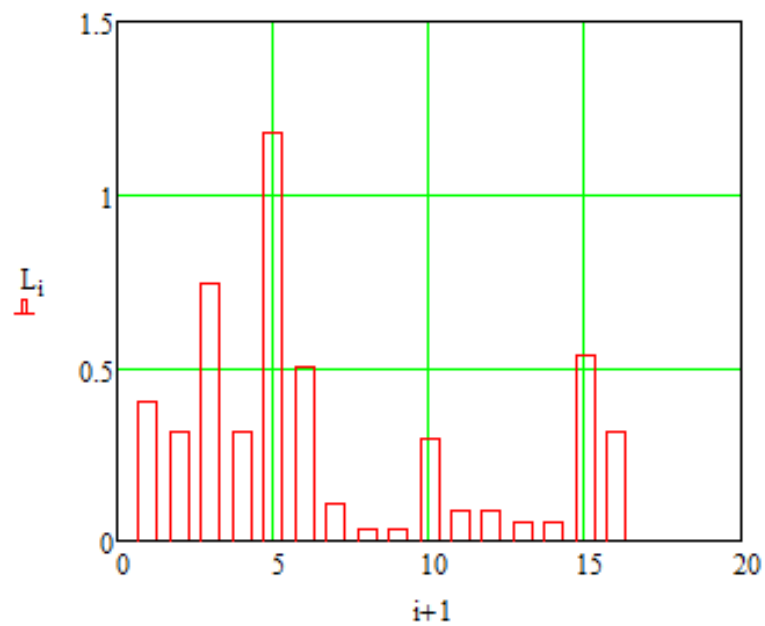
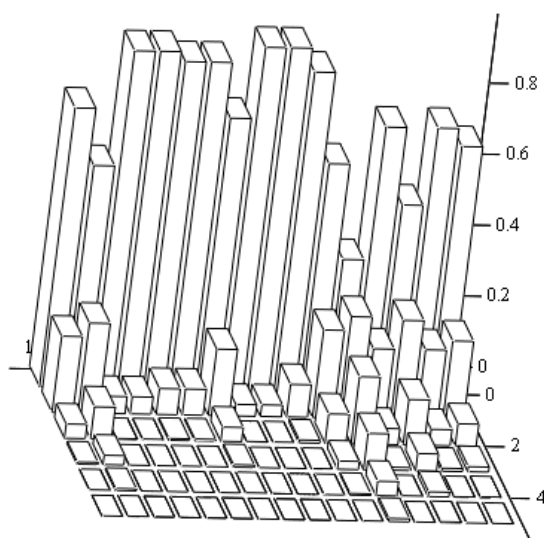


Рисунок 5.20 – Середня кількість пакетів у вузлі

Це може дозволити отримати доступ до лінії в сильно обмежених зонах мережі. Ефективність каналу по трафіку зображена на рисунку 5.21.

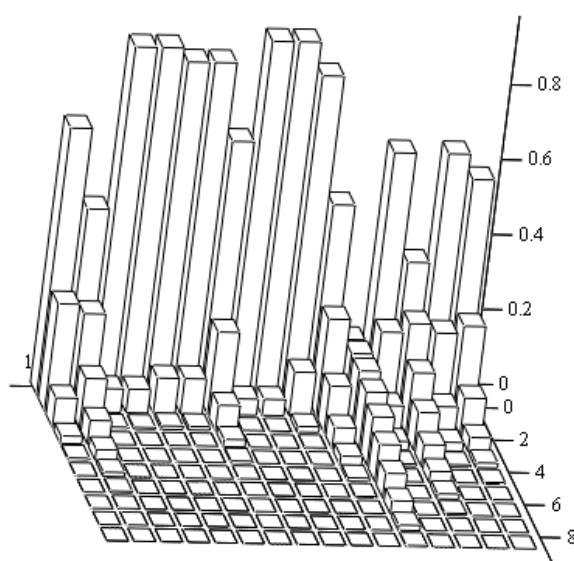


В

Рисунок 5.21 – Ефективність каналу по трафіку, при звичайному режимі роботи

Циркуляція пакетів у мережі зі скоригованими характеристиками зменшує ймовірність того, що вузол не потрапить у чергу до маршрутизатора. Пропускна здатність каналу зображена на рисунку 5.22.

Підвищення швидкості обробки пакетів на ділянках з найнижчою толерантністю до перевантажень покращило загальну продуктивність мережі.



В

Рисунок 5.22 – Пропускна здатність каналу, при граничній пропускній здатності

Аналіз даних, отриманих під час дослідження стану мережі при роботі на гранично допустимій здатності, показує, що через стохастичну природу фазової діаграми, яка описує мережу, навантаження на вузли мережі зростає лінійно, але в деяких вузлах ключові характеристики можуть змінюватися не лінійно.

Порівнюючи характеристики комп'ютерних мереж у досліджуваних станах, можна зробити висновок, що найбільш негативний вплив на продуктивність мережі має робота при граничній пропускній здатності[3].

## ВИСНОВКИ

У межах виконаної кваліфікаційної роботи вирішено наукове завдання, яке полягає у розробці оптимальної структури та визначенні параметрів мережевих пристроїв для підтримки комп'ютерної системи компанії, що займається інтернет-продажами. Основна ідея дослідження полягала в обґрунтуванні підходів до вдосконалення інфраструктури, спрямованих на підвищення надійності, швидкості та безпеки внутрішніх інформаційних процесів. Наукове значення роботи полягає у розробці методичних рекомендацій із застосування сучасних підходів до моніторингу та управління інформаційними ресурсами.

Було розроблено та впроваджено додаток внутрішнього моніторингу та локального вікі-сайту.

Результати дослідження підтвердили, що запропоновані методологічні підходи та технічні рішення дозволяють значно покращити роботу інформаційної системи компанії, сприяти підвищенню її конкурентоспроможності та забезпечити стійкість до можливих загроз.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Комплексна оцінка ефективності інтернет-магазинів в електронній комерції. ВІСНИК ХАРКІВСЬКОГО НАЦІОНАЛЬНОГО університету імені В. Н. КАРАЗІНА. 2019. Т. 1, № 96. URL: <https://doi.org/10.26565/2311-2379-2019-96-05> (дата звернення: 11.12.2024).
2. Mkansi M., Nsakanda A. Leveraging the physical network of stores in e-grocery order fulfilment for sustainable competitive advantage. research in transportation economics. sciencedirect. URL: <https://www.sciencedirect.com/science/article/pii/S0739885919303026?via=ihub> (date of access: 30.10.2024).
3. Пасічний С. І. Курсова робота з дисципліни «Імітаційне моделювання комп'ютерних систем і мереж». НТУ «Дніпровська політехніка». Дніпро 2023р.
4. Пасічний С. І. Кваліфікаційна робота бакалавра на тему: «Комп'ютерна система Інтернет-магазину товарів для сну з детальним опрацюванням побудови, налаштування та безпеки корпоративної мережі». НТУ «Дніпровська політехніка». Дніпро 2023р.
5. Python Documentation. Python Docs. URL: <https://docs.python.org/3/> (date of access: 22.11.2024).
6. Литвин А. Освіта та інформаційні технології : Посібник. 25-те вид. Львів, 2019. URL: <https://doi.org/10.1007/s10639-019-09966-4> (дата звернення: 12.12.2024).
7. Фурда О. Курс Flask. ITdvn. URL: <https://itvdn.com/ua/video/flask-ua> (date of access: 01.12.2024).



**ДОДАТОК А****ТЕКСТ ПРОГРАМИ МОНІТОРИНГУ І ВІКІ-САЙТУ**

**Міністерство освіти і науки України**  
**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ**  
**«ДНІПРОВСЬКА ПОЛІТЕХНІКА»**

**ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ**  
**ПРОГРАМИ МОНІТОРИНГУ І ВІКІ-САЙТУ**  
**КОМПАНІЇ ІНТЕРНЕТ-ПРОДАЖІВ**

Текст програми

804.02070743.24028-01 12 01

Листів 6

## АНОТАЦІЯ

Дана програма містить в собі програмний код системи моніторингу для управління та контролю робочих процесів. Програма призначена для збору даних, обробки інформації у реальному часі та надання можливості візуалізації ключових показників системи користувачам. Програма написана мовою програмування Python з використанням Flask.

Даний веб-портал є внутрішньою корпоративною системою для управління інформацією компанії. Портал містить інтерфейс для перегляду корпоративних новин, доступу до документації, керування задачами та графіками, а також організації навчання і професійного розвитку співробітників.

Вікі-сайт розроблено з використанням мови розмітки HTML5 та стилів CSS3, що забезпечує адаптивний дизайн та зручність користування на різних пристроях. Структура сайту дозволяє легко розширювати функціонал, додаючи нові сторінки чи модулі.

Призначення порталу – спрощення комунікації між відділами компанії, підвищення ефективності доступу до необхідної інформації та створення інтерактивного простору для взаємодії співробітників.

## ЗМІСТ

1 Програма системи моніторингу .....	85
1.1 Файл monitoring.py .....	85
2 Програма вікі-сайту .....	89
2.1 Файл app.py .....	89
2.2 Файл index.html .....	90
2.3 Файл add_page.html .....	91
2.4 Файл documents.html .....	92
2.5 Файл news.html .....	93
2.6 Файл tasks.html .....	94
2.7 Файл training.html .....	95
2.8 Файл style.css .....	96

# 1 ПРОГРАМА СИСТЕМИ МОНІТОРИНГУ

## 1.1 Файл `monitoring.py`

```

import customtkinter as ctk
import random
from datetime import datetime
import threading
import time
import matplotlib.pyplot as plt
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
import tkinter as tk
from tkinter import ttk

class NetworkMonitor:
    def __init__(self, root):
        self.root = root
        self.root.title("Network Monitor GUI")
        self.root.geometry("900x600")

        self.devices = [
            {"name": "LAN_1", "count": "77", "ip": "172.23.169.0", "status": "Online",
            "traffic": 0},
            {"name": "LAN_2", "count": "81", "ip": "172.23.168.128", "status":
            "Online", "traffic": 0},
            {"name": "LAN_3", "count": "57", "ip": "172.23.170.0", "status": "Online",
            "traffic": 0},
            {"name": "LAN_4", "count": "71", "ip": "172.23.169.128", "status":
            "Online", "traffic": 0},
            {"name": "LAN_5", "count": "96", "ip": "172.23.168.0", "status": "Online",
            "traffic": 0},
        ]

        self.setup_gui()
        self.start_monitoring()

    def setup_gui(self):
        # Header
        header = ctk.CTkLabel(self.root, text="Network Monitoring System",
font=("Arial", 18, "bold"), text_color="white")
        header.pack(side=ctk.TOP, fill=ctk.X, pady=10)

        # Device TreeView
        self.tree_frame = ctk.CTkFrame(self.root)

```

```

self.tree_frame.pack(pady=20, padx=10, fill=ctk.BOTH, expand=True)

# Use the standard tkinter Treeview
columns = ("Name", "Node Count", "IP Address", "Status", "Traffic")
self.tree = ttk.Treeview(self.tree_frame, columns=columns, show="headings")

self.tree.heading("Name", text="Name")
self.tree.heading("Node Count", text="Node Count")
self.tree.heading("IP Address", text="IP Address")
self.tree.heading("Status", text="Status")
self.tree.heading("Traffic", text="Traffic")

for col in columns:
    self.tree.column(col, width=150, anchor=tk.CENTER)

self.tree.bind("<Double-1>", self.show_details)
self.tree.pack(fill=ctk.BOTH, expand=True)

# Control Buttons
self.button_frame = ctk.CTkFrame(self.root)
self.button_frame.pack(pady=10)

self.refresh_btn = ctk.CTkButton(self.button_frame, text="Refresh Data",
command=self.update_device_tree)
self.refresh_btn.grid(row=0, column=0, padx=10)

self.alert_btn = ctk.CTkButton(self.button_frame, text="Test Alerts",
command=self.test_alerts)
self.alert_btn.grid(row=0, column=1, padx=10)

# Logs
log_label = ctk.CTkLabel(self.root, text="Event Logs:", font=("Arial", 12,
"bold"))
log_label.pack(anchor=ctk.W, padx=10)

self.log_text = ctk.CTkTextbox(self.root, height=10, font=("Courier", 10))
self.log_text.pack(padx=10, fill=ctk.BOTH, expand=True)
self.log_text.insert(ctk.END, "[System] Network Monitor Started\n")

self.update_device_tree()

def update_device_tree(self):
    for row in self.tree.get_children():
        self.tree.delete(row)

for device in self.devices:
    device["traffic"] = random.randint(10, 1000) # Random traffic in KB/s

```

```

        self.tree.insert("", "end", values=(device["name"], device["count"],
device["ip"],
                                device["status"], f"{device['traffic']} KB/s"))

```

```

self.log_event("Device data refreshed.")

```

```

def start_monitoring(self):

```

```

    thread = threading.Thread(target=self.monitor_devices)

```

```

    thread.daemon = True

```

```

    thread.start()

```

```

def monitor_devices(self):

```

```

    while True:

```

```

        time.sleep(5)

```

```

        for device in self.devices:

```

```

            if random.random() < 0.1:

```

```

                device["status"] = "Offline"

```

```

                self.log_event(f"ALERT: {device['name']} is Offline!")

```

```

            else:

```

```

                device["status"] = "Online"

```

```

            self.update_device_tree()

```

```

def log_event(self, message):

```

```

    timestamp = datetime.now().strftime("%Y-%m-%d %H:%M:%S")

```

```

    self.log_text.insert(ctk.END, f"[{timestamp}] {message}\n")

```

```

    self.log_text.see(ctk.END)

```

```

def test_alerts(self):

```

```

    ctk.CTkMessageBox.show_warning("Test Alert", "This is a test alert to simulate
notifications.")

```

```

    self.log_event("Test alert triggered.")

```

```

def show_details(self, event):

```

```

    selected_item = self.tree.selection()

```

```

    if not selected_item:

```

```

        return

```

```

    item = self.tree.item(selected_item)

```

```

    values = item["values"]

```

```

    name = values[0]

```

```

    detail_window = ctk.CTkToplevel(self.root)

```

```

    detail_window.title(f"Details for {name}")

```

```

    detail_window.geometry("800x600")

```

```

    ctk.CTkLabel(detail_window, text=f"Details for {name}", font=("Arial", 16,
"bold")).pack(pady=10)

```

```

cpu_usage = random.randint(10, 90)
available_space = random.randint(50, 500)
active_nodes = random.randint(40, 100)
problem_nodes = random.randint(0, 10)

    ctk.CTkLabel(detail_window, text=f"CPU Usage: {cpu_usage}%",
font=("Arial", 12)).pack(anchor=ctk.W, padx=20, pady=5)
    ctk.CTkLabel(detail_window, text=f"Available Storage: {available_space} GB",
font=("Arial", 12)).pack(anchor=ctk.W, padx=20, pady=5)
    ctk.CTkLabel(detail_window, text=f"Active Nodes: {active_nodes}",
font=("Arial", 12)).pack(anchor=ctk.W, padx=20, pady=5)
    ctk.CTkLabel(detail_window, text=f"Problem Nodes: {problem_nodes}",
font=("Arial", 12)).pack(anchor=ctk.W, padx=20, pady=5)

chart_frame = ctk.CTkFrame(detail_window)
chart_frame.pack(fill=ctk.BOTH, expand=True, pady=10)

figure = plt.Figure(figsize=(5, 3), dpi=100)
ax = figure.add_subplot(111)
ax.plot([random.randint(10, 90) for _ in range(10)], label="CPU Usage",
color="blue")
ax.set_title("CPU Utilization Over Time")
ax.set_ylabel("% Usage")
ax.set_xlabel("Time")
ax.legend()

canvas = FigureCanvasTkAgg(figure, chart_frame)
canvas.get_tk_widget().pack()

if __name__ == "__main__":
    root = ctk.CTk()

    app = NetworkMonitor(root)
    root.mainloop()

```



## 2 ПРОГРАМА ВІКІ-САЙТУ

### 2.1 Файл app.py

```
from flask import Flask, render_template, request, redirect, url_for

app = Flask(__name__)

pages = [
    {"title": "Головна", "template": "index.html"},
    {"title": "Новини", "template": "news.html"},
    {"title": "Документація", "template": "documents.html"},
    {"title": "Завдання", "template": "tasks.html"},
    {"title": "Навчання", "template": "training.html"}
]

@app.route('/')
def index():
    return render_template('index.html')

@app.route('/news')
def about():
    return render_template('news.html')

@app.route('/documents')
def documents():
    return render_template('documents.html')

@app.route('/tasks')
def tasks():
    return render_template('tasks.html')

@app.route('/training')
def training():
    return render_template('training.html')

@app.route('/add', methods=['GET', 'POST'])
def add_page():
    if request.method == 'POST':
```

```

title = request.form['title']
template = request.form['template']

# Додаємо нову сторінку до списку
pages.append({"title": title, "template": template})

# Перенаправляємо на головну сторінку після додавання
return redirect(url_for('index'))

# Якщо GET запит, повертаємо форму для додавання сторінки
return render_template('add_page.html')

@app.route('/page/<title>')
def page(title):
    for p in pages:
        if p["title"].lower() == title.lower():
            return render_template(p["template"]) # Завантажуємо шаблон сторінки
    return "Page not found", 404

if __name__ == '__main__':
    app.run(debug=True)

```

## 2.2 Файл index.html

```

<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="../static/css/style.css">
  <title>Корпоративний портал</title>
</head>
<body>
  <header>
    <h1>Корпоративний портал</h1>
    <p>Ласкаво просимо до внутрішнього сайту компанії!</p>
  </header>

  <nav>
    <a href="/">Головна</a>
    <a href="/news">Новини</a>

```

```

    <a href="/documents">Документація</a>
    <a href="/tasks">Графік та задачі</a>
    <a href="/training">Навчання</a>
    <a href="/add">Додати сторінку</a>
</nav>

<section>
  <h2>Ласкаво просимо!</h2>
  <p>Цей портал створено для підтримки комунікації між
співробітниками та надання доступу до важливої інформації та ресурсів
компанії. Тут ви знайдете всі необхідні інструменти для продуктивної
роботи.</p>
</section>

<footer>
  <p>&copy; 2024 Ваша компанія. Всі права захищені.</p>
  <p>Контакти технічної підтримки: support@company.com | +38 (XXX)
XXX-XX-XX</p>
</footer>
</body>
</html>

```

## 2.3 Файл add\_page.html

```

<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="../static/css/style.css">
  <title>Додати сторінку</title>
</head>
<body>
  <header>
    <h1>Додати нову сторінку</h1>
  </header>

  <nav>
    <a href="/">Головна</a>
    <a href="/news">Новини</a>
    <a href="/documents">Документація</a>
    <a href="/tasks">Графік та задачі</a>
    <a href="/training">Навчання</a>
    <a href="/add">Додати сторінку</a>

```

```

</nav>
<section>
  <form action="/add" method="POST">
    <label for="title">Заголовок сторінки:</label><br>
    <input type="text" id="title" name="title" required><br><br>

    <label for="template">Шаблон сторінки (наприклад:
index.html):</label><br>
    <input type="text" id="template" name="template" required><br><br>

    <input type="submit" value="Додати сторінку">
  </form>
</section>

<footer>
  <p>&copy; 2024 Ваша компанія. Всі права захищені.</p>
  <p>Контакти технічної підтримки: support@company.com | +38 (XXX)
XXX-XX-XX</p>
</footer>
</body>
</html>

```

## 2.4 Файл documents.html

```

<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="../static/css/style.css">
  <title>Документація та ресурси</title>
</head>
<body>
  <header>
    <h1>Документація та ресурси</h1>
    <p>Важлива інформація для співробітників</p>
  </header>

  <nav>
    <a href="/">Головна</a>
    <a href="/news">Новини</a>
    <a href="/documents">Документація</a>
    <a href="/tasks">Графік та задачі</a>
    <a href="/training">Навчання</a>

```

```

    <a href="/add">Додати сторінку</a>
</nav>

<section>
  <h2>Корисні ресурси</h2>
  <p>Ви можете завантажити посібники, інструкції та інші матеріали
для вашої роботи прямо тут. Це допоможе вам швидше орієнтуватися в
наших процесах та інструментах.</p>
</section>

<footer>
  <p>&copy; 2024 Ваша компанія. Всі права захищені.</p>
  <p>Контакти технічної підтримки: support@company.com | +38 (XXX)
XXX-XX-XX</p>
</footer>
</body>
</html>

```

## 2.5 Файл news.html

```

<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="../static/css/style.css">
  <title>Новини компанії</title>
</head>
<body>
  <header>
    <h1>Новини компанії</h1>
    <p>Останні новини та оновлення</p>
  </header>

  <nav>
    <a href="/">Головна</a>
    <a href="/news">Новини</a>
    <a href="/documents">Документація</a>
    <a href="/tasks">Графік та задачі</a>
    <a href="/training">Навчання</a>
    <a href="/add">Додати сторінку</a>
  </nav>

  <section>

```

```

<h2>Останні новини</h2>
<p><strong>Оновлення відділу маркетингу</strong>: Наша команда маркетингу успішно завершила кампанію з запуску нового продукту. Вона отримала схвальні відгуки від клієнтів та медіа.</p>
<p><strong>Нове корпоративне навчання</strong>: Всі співробітники компанії запрошуються на курс з особистої ефективності та управління часом. Зареєструватися можна через портал навчання.</p>
<p><strong>Інтеграція нових інструментів</strong>: В компанії будуть впроваджені нові інструменти для автоматизації робочих процесів, що значно полегшить роботу відділів.</p>
</section>

<footer>
<p>&copy; 2024 Ваша компанія. Всі права захищені.</p>
<p>Контакти технічної підтримки: support@company.com | +38 (XXX) XXX-XX-XX</p>
</footer>
</body>
</html>

```

## 2.6 Файл tasks.html

```

<!DOCTYPE html>
<html lang="uk">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<link rel="stylesheet" href="../static/css/style.css">
<title>Графік та задачі</title>
</head>
<body>
<header>
<h1>Графік та задачі</h1>
<p>Керуйте своїми задачами та часом</p>
</header>

<nav>
<a href="/">Головна</a>
<a href="/news">Новини</a>
<a href="/documents">Документація</a>
<a href="/tasks">Графік та задачі</a>
<a href="/training">Навчання</a>
<a href="/add">Додати сторінку</a>
</nav>

```

```

<section>
  <h2>Ваші задачі на сьогодні</h2>
  <p>Ось список завдань, які потрібно виконати сьогодні: 1. Оновити документ А. 2. Завершити презентацію В. 3. Провести зустріч з командою...</p>
</section>

<footer>
  <p>&copy; 2024 Ваша компанія. Всі права захищені.</p>
  <p>Контакти технічної підтримки: support@company.com | +38 (XXX) XXX-XX-XX</p>
</footer>
</body>
</html>

```

## 2.7 Файл training.html

```

<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="../static/css/style.css">
  <title>Навчання та розвиток</title>
</head>
<body>
  <header>
    <h1>Навчання та розвиток</h1>
    <p>Можливості для професійного розвитку</p>
  </header>

  <nav>
    <a href="/">Головна</a>
    <a href="/news">Новини</a>
    <a href="/documents">Документація</a>
    <a href="/tasks">Графік та задачі</a>
    <a href="/training">Навчання</a>
    <a href="/add">Додати сторінку</a>
  </nav>

  <section>
    <h2>Онлайн курси та тренінги</h2>
    <p>Ми пропонуємо курси та тренінги для розвитку нових навичок, щоб

```

підвищити вашу кваліфікацію та здатність до вирішення складних завдань.</p>

</section>

<footer>

<p>&copy; 2024 Ваша компанія. Всі права захищені.</p>

<p>Контакти технічної підтримки: support@company.com | +38 (XXX) XXX-XX-XX</p>

</footer>

</body>

</html>

## 2.8 Файл style.css

```
/* General Reset */
```

```
{
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}
```

```
/* Body Styles */
```

```
body {
  font-family: 'Arial', sans-serif;
  line-height: 1.6;
  background-color: #f4f4f4;
  color: #333;
}
```

```
/* Header Styles */
```

```
header {
  background: #0078d7;
  color: white;
  padding: 20px 0;
  text-align: center;
}
```

```
header h1 {
  color: white;
}
```

```
section {
  max-width: 800px;
```



```
margin: 2rem auto;
background: white;
padding: 2rem;
border-radius: 8px;
box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
}
```

```
/* Navigation Styles */
```

```
nav {
  background: #005bb5;
  display: flex;
  justify-content: center;
  margin: 15px 0;
  padding: 0.5rem 0;
}
```

```
nav a {
  color: white;
  text-decoration: none;
  padding: 10px 20px;
  font-weight: bold;
  transition: background 0.3s;
}
```

```
nav a:hover {
  background: #45a1d6;
  text-decoration: underline;
}
```

```
/* Footer Styles */
```

```
footer {
  color: #555;
  text-align: center;
  margin: 2rem 0;
  padding: 20px 0;
}
```

```
h1, h2 {
  color: #0078d7;
}
```

**ДОДАТОК Б****ТЕКСТ ПРОГРАМ РЕАЛІЗАЦІЇ МАТЕМАТИЧНОЇ МОДЕЛІ  
КОМП'ЮТЕРНОЇ МЕРЕЖІ**

**Міністерство освіти і науки України**  
**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ**  
**«ДНІПРОВСЬКА ПОЛІТЕХНІКА»**

**ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ**  
**ПРОГРАМ РЕАЛІЗАЦІЇ МАТЕМАТИЧНОЇ МОДЕЛІ**  
**КОМП'ЮТЕРНОЇ МЕРЕЖІ**

Текст програми

804.02070743.24028-01 12 01

Листів 19

## АНОТАЦІЯ

Даний документ містить ПЗ реалізації математичної моделі комп'ютерної мережі.

Тексти програм реалізовані в середовищі Mathcad середовищі операційної системи Windows 11.

Програма реалізує рекурентний метод Бузена для розрахунку параметрів комп'ютерної мережі як замкнутої системи масового обслуговування.

## ЗМІСТ

1 Перелік використаних змінних .....	102
2 Текст програми .....	103
3 Результат розрахунку .....	115

## 1 ПЕРЕЛІК ВИКОРИСТАНИХ ЗМІННИХ

$N_n$  – кількість вузлів мережі.

$\tau$  – час обробки одного пакета у вузлі.

$P_r$  – матриця перехідних ймовірностей.

$e$  – матриця перехідних коефіцієнтів.

$m$  – кількість конвеєрів у вузлах.

$N$  – ступінь завантаженості каналів.

$B$  – матриця ймовірностей черги у вузлах.

$\lambda$  – середня інтенсивність запитів на вході у вузол.

$L$  – середня черга пакетів у вузлі.

$t$  – середній час перебування пакета у вузлі.



	0
0	$2 \cdot 10^{-5}$
1	$2 \cdot 10^{-5}$
2	$2 \cdot 10^{-5}$
3	$2 \cdot 10^{-5}$
4	$2 \cdot 10^{-5}$
5	$2 \cdot 10^{-5}$
6	$2 \cdot 10^{-5}$
7	$2 \cdot 10^{-5}$
8	$2 \cdot 10^{-5}$
9	$2 \cdot 10^{-5}$
10	$2 \cdot 10^{-5}$
11	$2 \cdot 10^{-5}$
12	$2 \cdot 10^{-5}$
13	$2 \cdot 10^{-5}$
14	$2 \cdot 10^{-5}$
15	$2 \cdot 10^{-5}$

Розрахунок інтенсивності обробки запитів в вузлах мережі

$$\mu_i := \frac{1}{\tau_i}$$

	0
0	$5 \cdot 10^4$
1	$5 \cdot 10^4$
2	$5 \cdot 10^4$
3	$5 \cdot 10^4$
4	$5 \cdot 10^4$
5	$5 \cdot 10^4$
6	$5 \cdot 10^4$
7	$5 \cdot 10^4$
8	$5 \cdot 10^4$
9	$5 \cdot 10^4$
10	$5 \cdot 10^4$
11	$5 \cdot 10^4$
12	$5 \cdot 10^4$
13	$5 \cdot 10^4$
14	$5 \cdot 10^4$
15	$5 \cdot 10^4$



## Матриця імовірностей передачі

Pr :=	0	0.2	0.4	0.2	0	0	0.2	0	0	0	0	0	0	0	0	0
	0.2	0	0.6	0	0	0	0	0	0	0	0	0	0.2	0	0	0
	0.15	0.15	0	0.15	0.4	0	0	0	0	0	0.15	0	0	0	0	0
	0.2	0	0.6	0	0	0	0	0	0	0	0	0	0	0.2	0	0
	0	0	0.3	0	0	0.4	0	0	0	0	0	0	0	0	0.3	0
	0	0	0	0	0.7	0	0	0	0	0	0	0	0	0	0	0.3
	0.6	0	0	0	0	0	0	0.2	0.2	0	0	0	0	0	0	0
	0	0	0	0	0	0	0.6	0	0.4	0	0	0	0	0	0	0
	0	0	0	0	0	0	0.6	0.4	0	0	0	0	0	0	0	0
	0	0	0.6	0	0	0	0	0	0	0	0.2	0.2	0	0	0	0
	0	0	0	0	0	0	0	0	0	0.6	0	0.4	0	0	0	0
	0	0	0	0	0	0	0	0	0	0.6	0.4	0	0	0	0	0
	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

Перевірка правильності заповнення передаточної матриці

$$\text{SumPr}_i := \sum_{j=0}^{Nn} \text{Pr}_{i,j}$$

	0
0	1
1	1
2	1
3	1
4	1
5	1
6	1
7	1
8	1
9	1
10	1
11	1
12	1
13	1
14	1
15	1

## Визначення коефіцієнтів передачі

$$P := P r^T$$

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0.2	0.15	0.2	0	0	0.6	0	0	0	0	0	0	0	0	0
1	0.2	0	0.15	0	0	0	0	0	0	0	0	0	1	0	0	0
2	0.4	0.6	0	0.6	0.3	0	0	0	0	0.6	0	0	0	0	0	0
3	0.2	0	0.15	0	0	0	0	0	0	0	0	0	0	1	0	0
4	0	0	0.4	0	0	0.7	0	0	0	0	0	0	0	0	1	0
5	0	0	0	0	0.4	0	0	0	0	0	0	0	0	0	0	1
6	0.2	0	0	0	0	0	0	0.6	0.6	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0.2	0	0.4	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0.2	0.4	0	0	0	0	0	0	0	0
9	0	0	0.15	0	0	0	0	0	0	0	0.6	0.6	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0.2	0	0.4	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0.2	0.4	0	0	0	0	0
12	0	0.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0.2	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0.3	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0.3	0	0	0	0	0	0	0	0	0	0

$$D := \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$P1 := P - D$$

$$j := 1..Nn$$

$$i := 0..Nn$$

$$P2_{(j-1),i} := P1_{0,i} + P1_{j,i}$$

P1 =

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	-1	0.2	0.15	0.2	0	0	0.6	0	0	0	0	0	0	0	0	0
1	0.2	-1	0.15	0	0	0	0	0	0	0	0	0	1	0	0	0
2	0.4	0.6	-1	0.6	0.3	0	0	0	0	0.6	0	0	0	0	0	0
3	0.2	0	0.15	-1	0	0	0	0	0	0	0	0	0	1	0	0
4	0	0	0.4	0	-1	0.7	0	0	0	0	0	0	0	0	1	0
5	0	0	0	0	0.4	-1	0	0	0	0	0	0	0	0	0	1
6	0.2	0	0	0	0	0	-1	0.6	0.6	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0.2	-1	0.4	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0.2	0.4	-1	0	0	0	0	0	0	0
9	0	0	0.15	0	0	0	0	0	0	-1	0.6	0.6	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0.2	-1	0.4	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0.2	0.4	-1	0	0	0	0
12	0	0.2	0	0	0	0	0	0	0	0	0	0	-1	0	0	0
13	0	0	0	0.2	0	0	0	0	0	0	0	0	0	-1	0	0
14	0	0	0	0	0.3	0	0	0	0	0	0	0	0	0	-1	0
15	0	0	0	0	0	0.3	0	0	0	0	0	0	0	0	0	-1

P2 =

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	-0.8	-0.8	0.3	0.2	0	0	0.6	0	0	0	0	0	1	0	0	0
1	-0.6	0.8	-0.85	0.8	0.3	0	0.6	0	0	0.6	0	0	0	0	0	0
2	-0.8	0.2	0.3	-0.8	0	0	0.6	0	0	0	0	0	0	1	0	0
3	-1	0.2	0.55	0.2	-1	0.7	0.6	0	0	0	0	0	0	0	1	0
4	-1	0.2	0.15	0.2	0.4	-1	0.6	0	0	0	0	0	0	0	0	1
5	-0.8	0.2	0.15	0.2	0	0	-0.4	0.6	0.6	0	0	0	0	0	0	0
6	-1	0.2	0.15	0.2	0	0	0.8	-1	0.4	0	0	0	0	0	0	0
7	-1	0.2	0.15	0.2	0	0	0.8	0.4	-1	0	0	0	0	0	0	0
8	-1	0.2	0.3	0.2	0	0	0.6	0	0	-1	0.6	0.6	0	0	0	0
9	-1	0.2	0.15	0.2	0	0	0.6	0	0	0.2	-1	0.4	0	0	0	0
10	-1	0.2	0.15	0.2	0	0	0.6	0	0	0.2	0.4	-1	0	0	0	0
11	-1	0.4	0.15	0.2	0	0	0.6	0	0	0	0	0	-1	0	0	0
12	-1	0.2	0.15	0.4	0	0	0.6	0	0	0	0	0	0	-1	0	0
13	-1	0.2	0.15	0.2	0.3	0	0.6	0	0	0	0	0	0	0	-1	0
14	-1	0.2	0.15	0.2	0	0.3	0.6	0	0	0	0	0	0	0	0	-1

$j := 0..Nn - 1$

$i := 0..Nn - 1$

$PP2_{j,i} := P2_{j,i+1}$

PP2 =

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	-0.8	0.3	0.2	0	0	0.6	0	0	0	0	0	1	0	0	0
1	0.8	-0.85	0.8	0.3	0	0.6	0	0	0.6	0	0	0	0	0	0
2	0.2	0.3	-0.8	0	0	0.6	0	0	0	0	0	0	1	0	0
3	0.2	0.55	0.2	-1	0.7	0.6	0	0	0	0	0	0	0	1	0
4	0.2	0.15	0.2	0.4	-1	0.6	0	0	0	0	0	0	0	0	1
5	0.2	0.15	0.2	0	0	-0.4	0.6	0.6	0	0	0	0	0	0	0
6	0.2	0.15	0.2	0	0	0.8	-1	0.4	0	0	0	0	0	0	0
7	0.2	0.15	0.2	0	0	0.8	0.4	-1	0	0	0	0	0	0	0
8	0.2	0.3	0.2	0	0	0.6	0	0	-1	0.6	0.6	0	0	0	0
9	0.2	0.15	0.2	0	0	0.6	0	0	0.2	-1	0.4	0	0	0	0
10	0.2	0.15	0.2	0	0	0.6	0	0	0.2	0.4	-1	0	0	0	0
11	0.4	0.15	0.2	0	0	0.6	0	0	0	0	0	-1	0	0	0
12	0.2	0.15	0.4	0	0	0.6	0	0	0	0	0	0	-1	0	0
13	0.2	0.15	0.2	0.3	0	0.6	0	0	0	0	0	0	0	-1	0
14	0.2	0.15	0.2	0	0.3	0.6	0	0	0	0	0	0	0	0	-1

$Q_{j,0} := P2_{j,0}$

	0
0	-0.8
1	-0.6
2	-0.8
3	-1
4	-1
5	-0.8
6	-1
7	-1
8	-1
9	-1
10	-1
11	-1
12	-1
13	-1
14	-1

Q =

E := Isolve(PP2,Q)

	0
0	-0.833
1	-3.111
2	-0.833
3	-4.148
4	-2.37
5	-0.333
6	-0.111
7	-0.111
8	-0.778
9	-0.259
10	-0.259
11	-0.167
12	-0.167
13	-1.244
14	-0.711

E =



	0
0	1
1	1
2	1
3	1
4	1
5	1
6	1
m = 7	1
8	1
9	1
10	1
11	1
12	1
13	1
14	1
15	1

Розрахунок значень функції A

$$A_{m,j} := \begin{cases} (j!) & \text{if } m_1 \geq N - 1 \\ 1 & \text{if } m_1 = 1 \\ (j!) & \text{if } 1 < m_1 < N - 1 \wedge j \leq m_1 \\ m_1! \binom{m_1}{j}^{j-m_1} & \text{if } (1 < m_1 < N - 1 \wedge j > m_1) \end{cases}$$

	0	1	2	3	4	5
0	1	1	1	1	1	1
1	1	1	1	1	1	1
2	1	1	1	1	1	1
3	1	1	1	1	1	1
4	1	1	1	1	1	1
5	1	1	1	1	1	1
A = 6	1	1	1	1	1	1
7	1	1	1	1	1	1
8	1	1	1	1	1	1
9	1	1	1	1	1	1
10	1	1	1	1	1	1
11	1	1	1	1	1	1
12	1	1	1	1	1	1
13	1	1	1	1	1	1
14	1	1	1	1	1	1
15	1	1	1	1	1	1

$$X_i := \frac{e_i}{\mu_i}$$

X =

	0
0	$2 \cdot 10^{-5}$
1	$1.666 \cdot 10^{-5}$
2	$6.222 \cdot 10^{-5}$
3	$1.666 \cdot 10^{-5}$
4	$8.296 \cdot 10^{-5}$
5	$4.74 \cdot 10^{-5}$
6	$6.66 \cdot 10^{-6}$
7	$2.22 \cdot 10^{-6}$
8	$2.22 \cdot 10^{-6}$
9	$1.556 \cdot 10^{-5}$
10	$5.18 \cdot 10^{-6}$
11	$5.18 \cdot 10^{-6}$
12	$3.34 \cdot 10^{-6}$
13	$3.34 \cdot 10^{-6}$
14	$2.488 \cdot 10^{-5}$
15	$1.422 \cdot 10^{-5}$

Обчислення матриці констант T

$$T_{i,j} := \frac{(X_i)^j}{A_{i,j}}$$

$$T_{i,0} := 1$$

T =

	0	1	2	3	4
0	1	$2 \cdot 10^{-5}$	$4 \cdot 10^{-10}$	$8 \cdot 10^{-15}$	0
1	1	$1.666 \cdot 10^{-5}$	$2.776 \cdot 10^{-10}$	$4.624 \cdot 10^{-15}$	0
2	1	$6.222 \cdot 10^{-5}$	$3.871 \cdot 10^{-9}$	$2.409 \cdot 10^{-13}$	0
3	1	$1.666 \cdot 10^{-5}$	$2.776 \cdot 10^{-10}$	$4.624 \cdot 10^{-15}$	0
4	1	$8.296 \cdot 10^{-5}$	$6.882 \cdot 10^{-9}$	$5.71 \cdot 10^{-13}$	0
5	1	$4.74 \cdot 10^{-5}$	$2.247 \cdot 10^{-9}$	$1.065 \cdot 10^{-13}$	0
6	1	$6.66 \cdot 10^{-6}$	$4.436 \cdot 10^{-11}$	0	0
7	1	$2.22 \cdot 10^{-6}$	$4.928 \cdot 10^{-12}$	0	0
8	1	$2.22 \cdot 10^{-6}$	$4.928 \cdot 10^{-12}$	0	0
9	1	$1.556 \cdot 10^{-5}$	$2.421 \cdot 10^{-10}$	$3.767 \cdot 10^{-15}$	0
10	1	$5.18 \cdot 10^{-6}$	$2.683 \cdot 10^{-11}$	0	0
11	1	$5.18 \cdot 10^{-6}$	$2.683 \cdot 10^{-11}$	0	0
12	1	$3.34 \cdot 10^{-6}$	$1.116 \cdot 10^{-11}$	0	0
13	1	$3.34 \cdot 10^{-6}$	$1.116 \cdot 10^{-11}$	0	0
14	1	$2.488 \cdot 10^{-5}$	$6.19 \cdot 10^{-10}$	$1.54 \cdot 10^{-14}$	0
15	1	$1.422 \cdot 10^{-5}$	$2.022 \cdot 10^{-10}$	$2.875 \cdot 10^{-15}$	...

Розраховуємо константи для другого і наступних вузлів

$$i := 1..Nn$$

$$k := 0..N - 1$$

$$G_{0,j} := T_{0,j}$$

$$G_{i,k} := \sum_{j=0}^k (T_{i,j} \cdot G_{i-1,k-j})$$

	0	1	2	3	4
0	1	$2 \cdot 10^{-5}$	$4 \cdot 10^{-10}$	$8 \cdot 10^{-15}$	0
1	1	$3.666 \cdot 10^{-5}$	$1.011 \cdot 10^{-9}$	$2.484 \cdot 10^{-14}$	0
2	1	$9.888 \cdot 10^{-5}$	$7.163 \cdot 10^{-9}$	$4.705 \cdot 10^{-13}$	0
3	1	$1.155 \cdot 10^{-4}$	$9.088 \cdot 10^{-9}$	$6.219 \cdot 10^{-13}$	0
4	1	$1.985 \cdot 10^{-4}$	$2.556 \cdot 10^{-8}$	$2.742 \cdot 10^{-12}$	0
5	1	$2.459 \cdot 10^{-4}$	$3.721 \cdot 10^{-8}$	$4.506 \cdot 10^{-12}$	0
6	1	$2.526 \cdot 10^{-4}$	$3.889 \cdot 10^{-8}$	$4.765 \cdot 10^{-12}$	0
7	1	$2.548 \cdot 10^{-4}$	$3.946 \cdot 10^{-8}$	$4.852 \cdot 10^{-12}$	0
8	1	$2.57 \cdot 10^{-4}$	$4.003 \cdot 10^{-8}$	$4.941 \cdot 10^{-12}$	0
9	1	$2.726 \cdot 10^{-4}$	$4.427 \cdot 10^{-8}$	$5.63 \cdot 10^{-12}$	0
10	1	$2.777 \cdot 10^{-4}$	$4.571 \cdot 10^{-8}$	$5.867 \cdot 10^{-12}$	0
11	1	$2.829 \cdot 10^{-4}$	$4.717 \cdot 10^{-8}$	$6.111 \cdot 10^{-12}$	0
12	1	$2.863 \cdot 10^{-4}$	$4.813 \cdot 10^{-8}$	$6.272 \cdot 10^{-12}$	0
13	1	$2.896 \cdot 10^{-4}$	$4.91 \cdot 10^{-8}$	$6.436 \cdot 10^{-12}$	0
14	1	$3.145 \cdot 10^{-4}$	$5.692 \cdot 10^{-8}$	$7.852 \cdot 10^{-12}$	0
15	1	$3.287 \cdot 10^{-4}$	$6.16 \cdot 10^{-8}$	$8.728 \cdot 10^{-12}$	...

$$B_{Nn,j} := \frac{T_{Nn,j}}{G_{Nn,N-1}} \cdot G_{Nn,N-1-j}$$

$$B_{Nn,0} := 1 - B_{Nn,1}$$

	0	1	2	3	4
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0
5	0	0	0	0	0
6	0	0	0	0	0
7	0	0	0	0	0
8	0	0	0	0	0
9	0	0	0	0	0
10	0	0	0	0	0
11	0	0	0	0	0
12	0	0	0	0	0
13	0	0	0	0	0
14	0	0	0	0	0
15	0.868	0.132	0.016	$1.574 \cdot 10^{-3}$	...



Розрахунок допоміжних коефіцієнтів

$$i := 0..Nn - 1$$

$$j := 1..N - 1$$

$$Gn_{i,0} := 1$$

$$Gn_{i,j} := G_{Nn,j} - \sum_{k=1}^j (T_{i,k} \cdot Gn_{i,j-k})$$

	0	1	2	3	4
0	1	$3.087 \cdot 10^{-4}$	$5.502 \cdot 10^{-8}$	$7.496 \cdot 10^{-12}$	0
1	1	$3.12 \cdot 10^{-4}$	$5.612 \cdot 10^{-8}$	$7.702 \cdot 10^{-12}$	0
2	1	$2.665 \cdot 10^{-4}$	$4.114 \cdot 10^{-8}$	$4.896 \cdot 10^{-12}$	0
3	1	$3.12 \cdot 10^{-4}$	$5.612 \cdot 10^{-8}$	$7.702 \cdot 10^{-12}$	0
4	1	$2.457 \cdot 10^{-4}$	$3.433 \cdot 10^{-8}$	$3.618 \cdot 10^{-12}$	0
5	1	$2.813 \cdot 10^{-4}$	$4.602 \cdot 10^{-8}$	$5.809 \cdot 10^{-12}$	0
6	1	$3.22 \cdot 10^{-4}$	$5.941 \cdot 10^{-8}$	$8.318 \cdot 10^{-12}$	0
7	1	$3.265 \cdot 10^{-4}$	$6.087 \cdot 10^{-8}$	$8.591 \cdot 10^{-12}$	$1.027 \cdot 10^{-15}$
8	1	$3.265 \cdot 10^{-4}$	$6.087 \cdot 10^{-8}$	$8.591 \cdot 10^{-12}$	$1.027 \cdot 10^{-15}$
9	1	$3.131 \cdot 10^{-4}$	$5.648 \cdot 10^{-8}$	$7.77 \cdot 10^{-12}$	0
10	1	$3.235 \cdot 10^{-4}$	$5.989 \cdot 10^{-8}$	$8.409 \cdot 10^{-12}$	$1.001 \cdot 10^{-15}$
11	1	$3.235 \cdot 10^{-4}$	$5.989 \cdot 10^{-8}$	$8.409 \cdot 10^{-12}$	$1.001 \cdot 10^{-15}$
12	1	$3.254 \cdot 10^{-4}$	$6.05 \cdot 10^{-8}$	$8.522 \cdot 10^{-12}$	$1.017 \cdot 10^{-15}$
13	1	$3.254 \cdot 10^{-4}$	$6.05 \cdot 10^{-8}$	$8.522 \cdot 10^{-12}$	$1.017 \cdot 10^{-15}$
14	1	$3.038 \cdot 10^{-4}$	$5.342 \cdot 10^{-8}$	$7.196 \cdot 10^{-12}$	...

$$i := 0..Nn - 1$$

$$j := 0..N - 1$$

$$B_{i,j} := \frac{T_{i,j}}{G_{Nn,N-1}} Gn_{i,N-1-j}$$

$$B_{11,j} := B_{10,j}$$

	1	2	3	4	5
0	0.155	0.027	$3.911 \cdot 10^{-3}$	$4.389 \cdot 10^{-4}$	$2.843 \cdot 10^{-5}$
1	0.133	0.019	$2.306 \cdot 10^{-3}$	$2.136 \cdot 10^{-4}$	$1.14 \cdot 10^{-5}$
2	0.278	0.168	0.088	0.035	$8.286 \cdot 10^{-3}$
3	0.133	0.019	$2.306 \cdot 10^{-3}$	$2.136 \cdot 10^{-4}$	$1.14 \cdot 10^{-5}$
4	0.238	0.221	0.174	0.103	0.035
5	0.266	0.116	0.044	0.013	$2.126 \cdot 10^{-3}$
6	0.058	$3.278 \cdot 10^{-3}$	$1.559 \cdot 10^{-4}$	$5.63 \cdot 10^{-6}$	$1.164 \cdot 10^{-7}$
7	0.02	$3.762 \cdot 10^{-4}$	$5.917 \cdot 10^{-6}$	$7.046 \cdot 10^{-8}$	$4.791 \cdot 10^{-10}$
8	0.02	$3.762 \cdot 10^{-4}$	$5.917 \cdot 10^{-6}$	$7.046 \cdot 10^{-8}$	$4.791 \cdot 10^{-10}$
9	0.126	0.017	$1.891 \cdot 10^{-3}$	$1.631 \cdot 10^{-4}$	$8.105 \cdot 10^{-6}$
10	0.046	$2.005 \cdot 10^{-3}$	$7.397 \cdot 10^{-5}$	$2.07 \cdot 10^{-6}$	$3.314 \cdot 10^{-8}$
11	0.046	$2.005 \cdot 10^{-3}$	$7.397 \cdot 10^{-5}$	$2.07 \cdot 10^{-6}$	$3.314 \cdot 10^{-8}$
12	0.03	$8.448 \cdot 10^{-4}$	$2.003 \cdot 10^{-5}$	$3.598 \cdot 10^{-7}$	$3.693 \cdot 10^{-9}$
13	0.03	$8.448 \cdot 10^{-4}$	$2.003 \cdot 10^{-5}$	$3.598 \cdot 10^{-7}$	$3.693 \cdot 10^{-9}$
14	0.183	0.04	$7.31 \cdot 10^{-3}$	$1.034 \cdot 10^{-3}$	$8.471 \cdot 10^{-5}$
15	0.132	0.016	$1.574 \cdot 10^{-3}$	$1.194 \cdot 10^{-4}$	...

$$i := 0..Nn$$

$$j := 0..N - 1$$

$$\text{SumB}_i := \sum_j B_{i,j}$$

	0
0	1
1	1
2	1
3	1
4	1
5	1
6	1
7	1
8	1
9	1
10	1
11	1
12	1
13	1
14	1
15	1.017

$$L_i := \sum_{n=0}^{N-1} (n \cdot B_{i,n})$$

$$\lambda_i := e_i \cdot \frac{G_{Nn-1, N-2}}{G_{Nn, N-1}}$$

### 3 РЕЗУЛЬТАТ РОЗРАХУНКУ

Інтенсивність вхідного потоку

Середнє число пакетів в вузлах

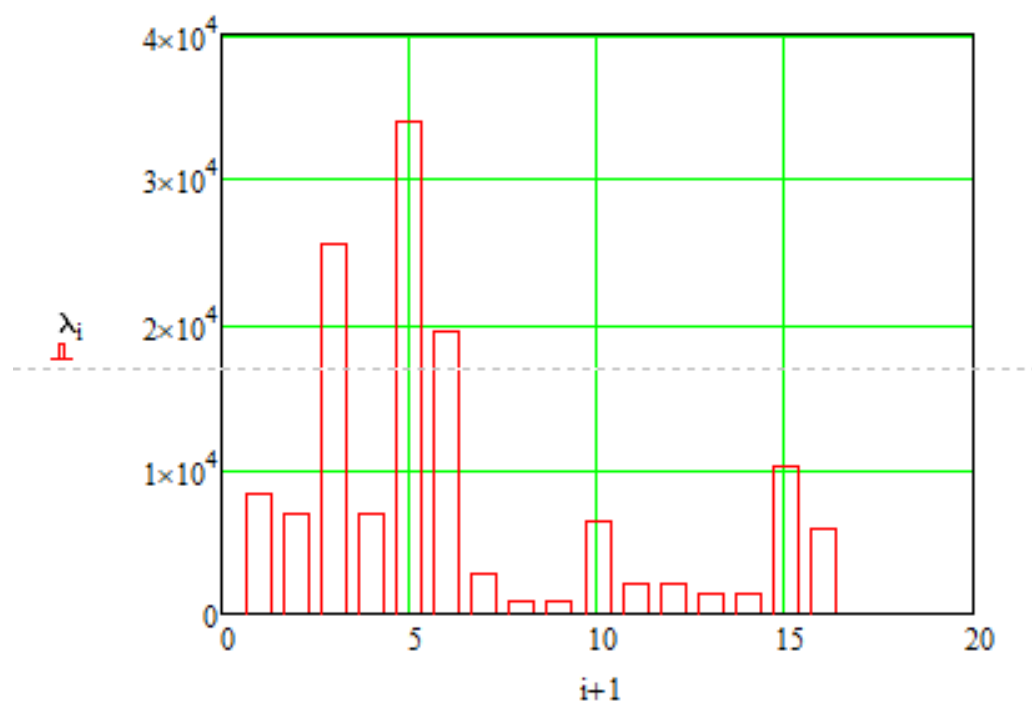
	0
0	$8.194 \cdot 10^3$
1	$6.826 \cdot 10^3$
2	$2.549 \cdot 10^4$
3	$6.826 \cdot 10^3$
-----	
4	$3.399 \cdot 10^4$
5	$1.942 \cdot 10^4$
6	$2.729 \cdot 10^3$
$\lambda =$	7 909.564
	8 909.564
	9 $6.375 \cdot 10^3$
	10 $2.122 \cdot 10^3$
	11 $2.122 \cdot 10^3$
	12 $1.368 \cdot 10^3$
	13 $1.368 \cdot 10^3$
	14 $1.019 \cdot 10^4$
	15 $5.826 \cdot 10^3$

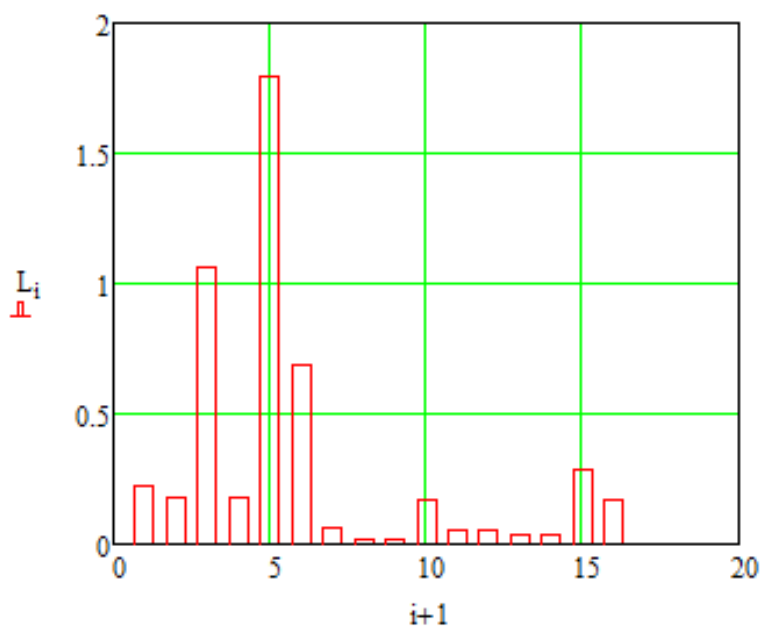
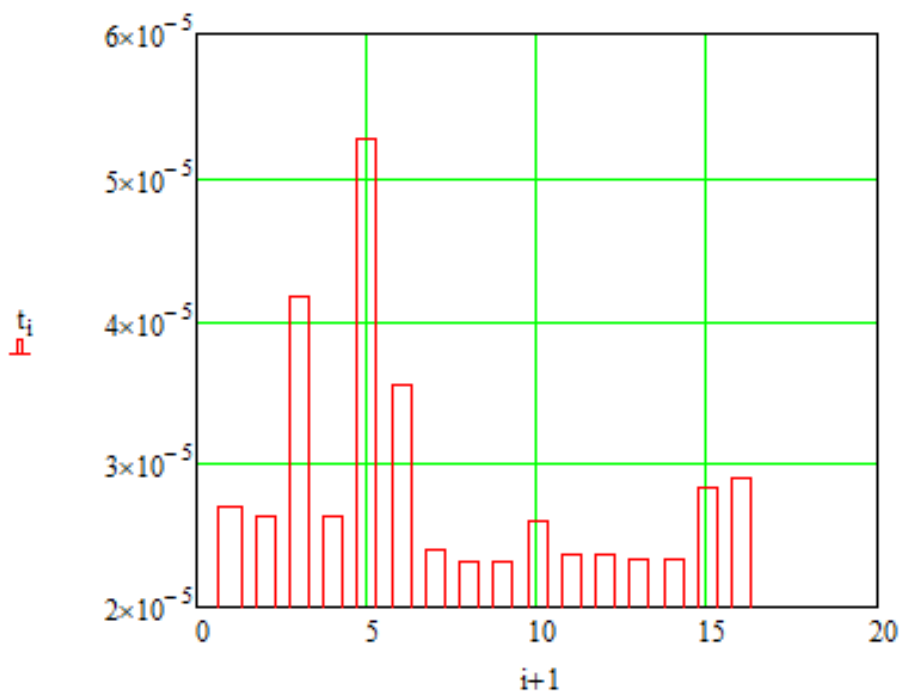
	0
	0 0.222
	1 0.179
	2 1.063
	3 0.179
	4 1.791
	5 0.69
	6 0.066
$L =$	7 0.021
	8 0.021
	9 0.166
	10 0.05
	11 0.05
	12 0.032
	13 0.032
	14 0.289
	15 0.169

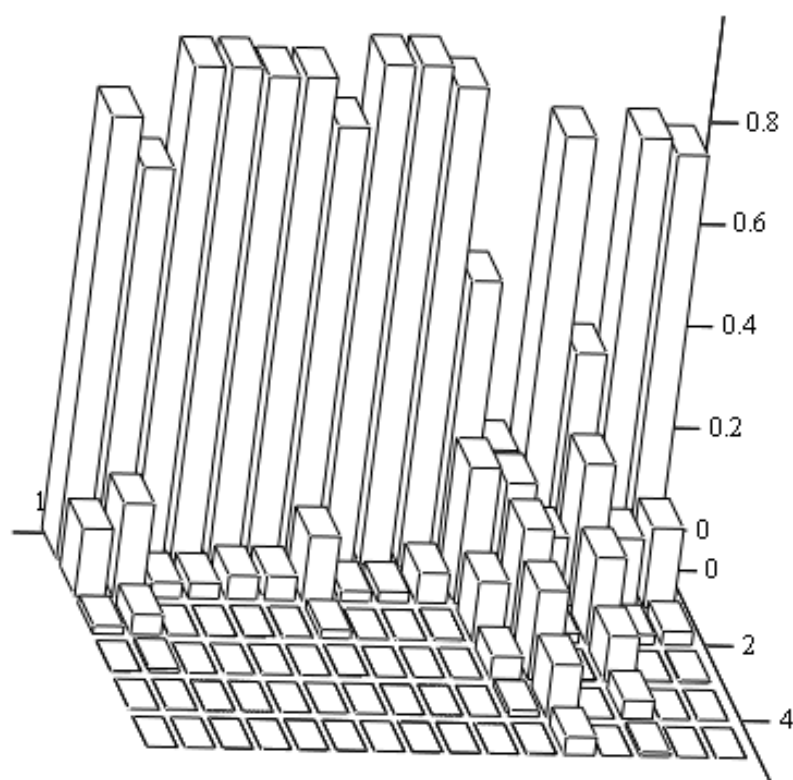
Середній час перебування пакета в вузлі

$$t_i := \frac{L_i}{\lambda_i}$$

	0
0	$2.707 \cdot 10^{-5}$
1	$2.625 \cdot 10^{-5}$
2	$4.168 \cdot 10^{-5}$
3	$2.625 \cdot 10^{-5}$
4	$5.269 \cdot 10^{-5}$
5	$3.553 \cdot 10^{-5}$
6	$2.401 \cdot 10^{-5}$
t = 7	$2.312 \cdot 10^{-5}$
8	$2.312 \cdot 10^{-5}$
9	$2.599 \cdot 10^{-5}$
10	$2.371 \cdot 10^{-5}$
11	$2.371 \cdot 10^{-5}$
12	$2.334 \cdot 10^{-5}$
13	$2.334 \cdot 10^{-5}$
14	$2.835 \cdot 10^{-5}$
15	$2.897 \cdot 10^{-5}$







B

