

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»**

М.М. Орлова, О.А. Щербина

КОМП'ЮТЕРНІ МЕРЕЖІ КУРСОВИЙ ПРОЄКТ

*Рекомендовано Методичною радою КПІ ім. Ігоря Сікорського
як навчальний посібник для здобувачів ступеня бакалавра за освітньою
програмою «Системне програмування і спеціалізовані комп'ютерні системи»
спеціальності 123 «Комп'ютерна інженерія»*

Київ
«КПІ ім. Ігоря Сікорського»

2021

Рецензент: Кулаков Ю.О., д-р техн. наук, проф., професор
кафедри обчислювальної техніки КПІ ім. Ігоря
Сікорського

Відповідальн
ий редактор Тарасенко В. П., д-р техн. наук, проф., професор
кафедри системного програмування і
спеціалізованих комп'ютерних систем КПІ ім. Ігоря
Сікорського

*Гриф надано Методичною радою КПІ ім. Ігоря Сікорського (протокол № 2 від
09.12.2021 р.)*

*за поданням Вченої ради факультету прикладної математики (протокол №
2 від 27.09.2021 р.)*

Електронне мережне навчальне видання
Орлова Марія Миколаївна, канд. техн. наук, доц.
Щербина Олександр Андрійович, канд. техн. наук, доц.

КОМП'ЮТЕРНІ МЕРЕЖІ

КУРСОВИЙ ПРОЄКТ

Комп'ютерні мережі: Курсовий проєкт [Електронний ресурс]: навч.
посібн. для студ. спеціальності 123 «Комп'ютерна інженерія» / М.М.
Орлова, О.А. Щербина; КПІ ім. Ігоря Сікорського. – Електронні текстові
дані (1 файл: 2,73 Мбайт). – Київ: КПІ ім. Ігоря Сікорського, 2021. – 76 с.

Навчальний посібник розроблено для виконання курсового
проєкту з дисципліни «Комп'ютерні мережі», а саме розробки
системи моделювання функціонування мережі, визначення
найбільш оптимального маршруту передачі та аналізу
отриманих результатів. Посібник містить постановку завдання,
вимоги до виконання курсового проєкту, оформлення його
результатів, необхідні теоретичні відомості, вимоги аналізу
отриманих результатів. Навчальне видання призначене для
студентів, які навчаються за спеціальністю 123 «Комп'ютерна
інженерія» факультету прикладної математики КПІ ім. Ігоря
Сікорського.

© М. М. Орлова, О. А. Щербина, 2021
© КПІ ім. Ігоря Сікорського, 2021

ЗМІСТ

ВСТУП.....	5
1. МЕТА ТА ЗАВДАННЯ КУРСОВОГО ПРОЄКТУ.....	6
2. ЗАВДАННЯ НА КУРСОВИЙ ПРОЄКТ.....	8
2.1 Типове завдання на курсовий проєкт.....	8
2.2. Основні вимоги до курсового проєкту.....	10
2.3. Вимоги до мови програмування.....	12
3. СТРУКТУРА, ОБСЯГ ТА ОФОРМЛЕННЯ КУРСОВОГО ПРОЄКТУ.....	13
3.1. Структура курсового проєкту.....	13
3.2. Вимоги до оформлення курсового проєкту.....	14
3.3. Вимоги до оформлення текстової частини.....	15
4. ТЕОРЕТИЧНІ ВІДОМОСТІ ДЛЯ ВИКОНАННЯ.....	24
КУРСОВОГО ПРОЄКТУ.....	24
4.1. Ідентифікація модулів на прикладі IP-адресації.....	24
4.2. Алгоритми маршрутизації потоків даних.....	28
4.3. Вибір найкоротшого шляху.....	29
4.4. Метод заливки.....	31
4.5. Маршрутизація за вектором відстаней.....	32
4.6. Маршрутизація з врахуванням стану лінії.....	35
4.7. Ієрархічна маршрутизація.....	35
4.8. Широкомовна маршрутизація.....	36
4.9. Класифікація протоколів динамічної маршрутизації.....	37
4.10. Дистанційно-векторні протоколи маршрутизації.....	44
4.11. Протоколи маршрутизації з урахуванням стану каналу.....	49
5. ЕТАПИ ВИКОНАННЯ І ПОРЯДОК ЗАХИСТУ КУРСОВОГО ПРОЄКТУ...55	
5.1. Графік виконання курсового проєкту.....	55
5.2. Порядок розгляду та захисту курсового проєкту.....	55

5.3. Рейтингова система оцінювання курсового проєкту.....	57
СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ.....	62
ДОПОМІЖНА ЛІТЕРАТУРА.....	63
ДОДАТКИ.....	65
ДОДАТОК А. ЗРАЗОК ОФОРМЛЕННЯ ТИТУЛЬНОГО АРКУША КУРСОВОГО ПРОЄКТУ.....	65
ДОДАТОК Б. ЗАВДАННЯ НА КУРСОВИЙ ПРОЄКТ.....	66
ДОДАТОК В. ПРИКЛАД ОФОРМЛЕННЯ АНОТАЦІЇ.....	68
ДОДАТОК Г. ПРАВИЛА ОФОРМЛЕННЯ СПИСКУ ЛІТЕРАТУРИ.....	69
ДОДАТОК Д. ТЕХНІЧНЕ ЗАВДАННЯ.....	72
ДОДАТОК Е. Перший аркуш пояснювальної записки.....	1

ВСТУП

Дисципліна «Комп'ютерні мережі» є базовою дисципліною циклу загальної підготовки бакалаврів за спеціальністю 123 «Комп'ютерна інженерія» освітньої програми бакалаврів «Системне програмування та спеціалізовані комп'ютерні системи», яка ознайомлює студентів з проблемами, які стоять перед системним програмуванням, і основними напрямками їх вирішення. Одним з важливих видів індивідуальних завдань, що передбачені навчальною програмою дисципліни «Комп'ютерні мережі» та виконуються студентами при її вивченні, є курсовий проєкт (КП).

Курсовий проєкт є творчим індивідуальним завданням і виконується студентами самостійно під керівництвом викладача згідно із завданням на основі набутих знань та умінь з даної дисципліни та суміжних дисциплін: «Структури даних та алгоритми», «Теорія інформації та кодування», «Теорія ймовірностей», «Комп'ютерні системи», «Архітектура комп'ютерів». Виконання курсового проєкту сприяє розширенню та поглибленню отриманих теоретичних знань щодо архітектури та структурної організації комп'ютерних мереж, принципів взаємодії робочих станцій, режимів передачі повідомлень в мережах передачі даних, формуванню вмінь аналізувати отримані результати, самостійного вирішення поставленої задачі оптимальним шляхом та оформлення документів курсового проєкту відповідно до чинних правил і стандартів.

Посібник містить загальну постановку завдання та зміст типового індивідуального завдання, рекомендації щодо методики виконання завдання, вимоги до виконання завдання, оформлення пояснювальної записки та всього курсового проєкту та термінів і процедури захисту курсового проєкту, рейтингової системи оцінювання тощо.

1. МЕТА ТА ЗАВДАННЯ КУРСОВОГО ПРОЄКТУ

Метою курсового проєкту з дисципліни «Комп'ютерні мережі» є закріплення та поглиблення знань щодо структурної організації комп'ютерних мереж (КМ), режимів передачі повідомлень в мережах передачі даних (МПД) комп'ютерної мережі, набуття практичних навичок у дослідженні особливостей структури та функціонування КМ, режимів передачі даних, алгоритмів маршрутизації повідомлень, показників різних типів трафіків тощо.

Завданням даного курсового проєкту є розробка системи моделювання роботи комп'ютерної мережі, вивчення та реалізація алгоритмів визначення найкоротшого маршруту передачі повідомлень різних розмірів та аналіз службового та інформаційного трафіка в мережі при різних умовах передачі. В результаті виконання **курсowego проєкту** студент має:

ЗНАТИ:

- способи побудови комунікаційних мереж передачі даних;
- алгоритми вибору мінімальних маршрутів передачі в мережі та маршрутів з мінімальною кількістю транзитних ділянок залежно від параметрів передачі, встановлених відправником повідомлення;
- особливості застосування протоколів маршрутизації та алгоритмів пошуку найбільш ефективних маршрутів передачі відповідно до встановлених параметрів передачі;
- методи організації каналів передачі даних та пошуку найбільш ефективних маршрутів передачі з врахуванням параметрів якості обслуговування, які сформовані відправником повідомлення;
- режими передачі в МПД та фрагментації повідомлень в комп'ютерній мережі залежно від параметрів каналів зв'язку та умов передачі;

- методики проектування структури мережі передачі даних та організації каналів зв'язку з різними параметрами передачі;

УМІТИ:

- самостійно працювати з навчально-методичною, технічною, періодичною літературою й іншими інформаційними джерелами за тематикою розробки;
- практично використовувати отримані знання для самостійного вирішення поставленого завдання;
- аналізувати вимоги до параметрів та особливостей передачі повідомлень, аналізувати структури каналів передачі та глобальної мережі;
- оцінювати параметри передачі даних з використанням різних способів, алгоритмів та протоколів передачі;
- узагальнювати отримані результати досліджень та формулювати рекомендації щодо найбільш ефективної структури мережі передачі даних та особливостей передачі повідомлень при заданих параметрах якості обслуговування;

НАПРАЦЮВАТИ ДОСВІД:

- проектування структури комп'ютерної мережі та способів передачі в ній відповідно до вимог, які сформовані відправником;
- розроблення алгоритмів найбільш ефективної передачі повідомлень при заданих параметрах якості обслуговування;
- застосування сучасних способів моделювання роботи комп'ютерної мережі.

2. ЗАВДАННЯ НА КУРСОВИЙ ПРОЄКТ

Тема курсового проєкту – Маршрутизація в мережі передачі даних.

Основним завданням курсового проєкту є розробка системи моделювання функціонування комп'ютерної мережі та проведення дослідження особливостей передачі повідомлень різного розміру з визначенням розміру інформаційного та службового трафіку, кількості переданих пакетів та оптимального розміру інформаційних пакетів для заданих умов передачі.

Курсові проєкти з дисципліни «Комп'ютерні мережі» можуть бути виконані за індивідуальною тематикою, визначеної рамками навчального плану. В цьому випадку студент може запропонувати свою тему, не передбачену тематикою кафедри, обґрунтувавши при цьому доцільність її розробки. Запропоновані теми курсового проєкту обов'язково узгоджуються з керівником проєкту не пізніше кінця четвертого тижня з початку семестра. При виборі теми студент повинен враховувати свої можливості, знання в обраному напрямку діяльності, науковий доробок, отриманий в ході практичної діяльності або при виконання інших проєктів. Тема курсового проєкту повинна бути актуальною, відображати сучасний стан проєктування, потребу підприємств і організацій.

На курсове проєктування студенти отримують типове стандартне завдання з індивідуальними вхідними даними за варіантом, який обрається за номером в списку групи. Типове завдання на розробку містить 5 завдань, останнє з яких є додатковим, і при його виконанні оцінюється додатковими заохочувальними балами.

2.1 Типове завдання на курсовий проєкт

Розробити програму, яка дозволяла б моделювати процес визначення маршруту передачі повідомлень в мережі передачі даних заданої конфігурації та передачу повідомлень довільного розміру в режимах:

- віртуального каналу;

- дейтаграмному режимі.

Завдання 1. Ввести конфігурацію мережі передачі даних (згідно з варіантом завдання), що має відповідну кількість комунікаційних вузлів і ступінь (середню кількість каналів, що підключені до комунікаційного вузла). Кожний канал передачі (канал зв'язку між комунікаційними вузлами) має сукупність параметрів: вага лінії (комплексний показник, який враховує різні характеристики каналу), тип (дуплексний, напівдуплексний) та ймовірність виникнення помилок. До кожного i -го комунікаційного вузла ($i = 1, 2, 3, 4, \dots$) підключена робоча станція.

Завдання 2. Використовуючи заданий алгоритм визначити найкоротший маршрут між однією робочою станцією мережі та всіма іншими робочими станціями мережі. Визначити маршрут (маршрути) з мінімальною кількістю транзитних ділянок. Представити таблиці відстаней та маршрутів у кожному комунікаційному вузлі мережі передачі даних.

Завдання 3. Провести серію тестів (не менше 10) з використанням створеної моделі мережі.

- Визначити час доставки повідомлень **різного розміру** при передачі пакетів **різної довжини** при організації віртуального каналу та при передачі в дейтаграмному режимі.
- **Обґрунтувати** вибрані розміри інформаційних і службових (управляючих) пакетів.
- Визначити кількість управляючих та інформаційних пакетів, необхідних для передачі повідомлень **різного розміру при різних значеннях розміру пакету**, при встановленні логічного з'єднання між вузлами мережі, організації віртуального каналу та при передачі в дейтаграмному режимі.
- Визначити загальний розмір управляючого та інформаційного трафіків.
- Визначити затримку передачі між найбільш територіально віддаленими робочими станціями.

- Визначити, як впливає ймовірність виникнення помилок на розмір службового трафіку.
- Представити залежність службового трафіку від розміру інформаційного пакету.

Завдання 4. Представити отримані результати у вигляді графіків (таблиць, діаграм тощо). Порівняти отримані результати за загальною кількістю переданих пакетів (інформаційних і службових) та розміру інформаційного та службового трафіків. **Зробити обґрунтовані висновки!**

- На основі проведеного аналізу визначити оптимальний розмір пакету для даних умов (конфігурації мережі передачі даних).

Завдання 5 (додаткове). Визначити середню інтенсивність запитів на передачу повідомлень від кінцевих вузлів, при якій час очікування його обслуговування не буде перевищувати заданий час T .

2.2. Основні вимоги до курсового проєкту

Курсовий проєкт складається з теоретичної, практичної (розробки програми та проведення відповідних тестів) та дослідницької частин.

Вимоги до **структури** курсового проєкту:

1. Теоретична частина повинна складатись з таких розділів:

- постановка задачі, теоретичні відомості;
- опис та аналіз методів маршрутизації при передачі повідомлень;
- алгоритм пошуку найкоротшого маршруту передачі за метрикою та кількістю транзитних ділянок;
- опис і аналіз запропонованого алгоритму при різних розмірах повідомлення та параметрах передачі.

*2. Опис розробленого **програмного продукту** повинен включати:*

- призначення та структура розробленого програмного продукту (ПП);
- опис функцій та процедур, які реалізовано в програмі;
- результати функціонування розробленого ПП;
- обґрунтування розміру інформаційного та службових пакетів.

3. Дослідницька частина повинна включати:

- опис процесу тестування передачі повідомлень різних розмірів при різних значеннях розміру пакетів та при різних параметрах передачі;
- представлення результатів тестування у вигляді таблиць, графіків, діаграм тощо);
- порівняльний аналіз отриманих результатів;
- повнота, коректність та конкретність порівняльного аналізу результатів у висновках.

Вимоги до **програмного продукту** курсового проєкту:

- швидке завдання вручну компонентів топології мережі <мишкою> - вузлів і каналів (повний дуплекс та напівдуплекс);
- випадкова генерація структури мережі з заданими політиками створення;
- політики для каналів: випадкового вибору ваги в заданих межах та константне значення із заданої множини значень;
- політики для буферів каналів: випадкового вибору ваги в заданих межах та константне значення з заданої множини значень;
- реалізацію основних звичних для користувача елементів управління: додавання, видалення, виділення, перетаскування вузлів і каналів;
- при захопленні об'єкта мишкою відображається основна інформація про об'єкт (наприклад, таблиця маршрутизації, вага каналів, завантаження буферів тощо);
- можливість відключати, включати обрані вузли і канали;
- перегляд покрокового виконання алгоритмів;
- генерування випадкового трафіка повідомлень;
- меню відправлення конкретних повідомлень із однієї в іншу робочу станцію мережі з зазначенням їх розміру.

Завдання на виконання курсового проєкту студенти отримують не пізніше четвертого тижня семестра, в якому передбачено виконання курсового проєкту. У завданні сформульована загальна постановка завдання, індивідуальне завдання, визначені основні вихідні дані, вимоги до роботи та очікувані результати, а також календарний план-графік виконання курсового проєкту та процедура його захисту. В окремих випадках, пункти завдання (або завдання в цілому) можуть бути змінені студентом за згодою викладача.

2.3. Вимоги до мови програмування

Мова програмування для створення системи моделювання обирається студентом самостійно та узгоджується з викладачем.

3. СТРУКТУРА, ОБСЯГ ТА ОФОРМЛЕННЯ КУРСОВОГО ПРОЄКТУ

3.1. Структура курсового проєкту

Курсовий проєкт має відповідати вимогам до таких документів і мати наступні розділи, які наведено в порядку їх розташування в готовій зброшурованій роботі.

- Титульний аркуш (зразок наведено в додатку А).
- Завдання на розробку (згідно визначеного варіанту, додаток Б).
- Анотація українською та англійською мовами (з 5-15 ключовими словами) – кожна анотація на окремому аркуші (зразок оформлення в додатку В).
- Технічне завдання (зразок оформлення в додатку Д).
- Пояснювальна записка, яка в свою чергу містить наступні розділи, підрозділи та параграфи.
 - Зміст.
 - Перелік скорочень, умовних позначень, термінів.
 - Вступ.

Розділ 1. Теоретична частина повинна, яка повинна складатись з таких підрозділів:

- постановка задачі, теоретичні відомості;
- опис та аналіз методів маршрутизації при передачі повідомлень;
- алгоритм пошуку найкоротшого маршруту передачі за метрикою та кількістю транзитних ділянок;
- опис і аналіз запропонованого алгоритму при різних розмірах повідомлення та параметрах передачі.

Розділ 2. Опис розробленого програмного продукту, який повинен включати:

- призначення та структура розробленого програмного продукту (ПП);

- опис функцій та процедур, які реалізовано в програмі;
- результати функціонування розробленого ПП.

Розділ 3. Дослідницька частина повинна включати:

- опис процесу тестування передачі повідомлень різних розмірів при різних значеннях розміру пакетів та при різних параметрах передачі;
 - представлення результатів тестування;
 - аналіз і порівняння отриманих результатів.
- Висновки (повинні бути повними, коректними і базуватись на даних порівняльного аналізу отриманих результатів).
- Список використаних літературних джерел (за останніми стандартами оформлення літературних посилань – додаток Г).
 - Додатки.

Висновки по роботі повинні бути достатньо **повними та обґрунтованими і базуватись на результатах, які отримано саме в процесі роботи і проведеного дослідження**. Всі пункти висновків мають бути основані на порівняльному аналізі отриманих результатів та підтверджуватись відповідними графіками або числовими значеннями.

3.2. Вимоги до оформлення курсового проєкту

Курсовий проєкт оформлюється відповідно до вимог ДСТУ 3008:2015 «Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання».

Текстова частина курсового проєкту повинна бути написана **грамотно, ТЕХНІЧНОЮ** мовою!!!

P.S. Не допускайте орфографічних помилок і помилок пунктуації! Все гарне враження від відмінної роботи дуже псується, якщо зустрічаються помилки. Перш ніж віддавати роботу на перевірку викладачеві, уважно прочитайте його від початку до кінця декілька разів або дайте прочитати будь-кому, хто грамотніший за Вас.

Найбільш важливі положення цього стандарту полягають в наступному.

Вимоги до обсягу та текстової частини курсового проєкту

- текстова частина: 25 – 50 сторінок (але не менше 25) сторінок формату А4 пояснювальної записки (тільки пояснювальної записки, без титульного аркуша, анотацій тощо);
- мова: українська;
- формат аркушів: А4 (210x297 мм);
- шрифт: Times New Roman 14 пунктів;
- міжрядковий інтервал: 1,5 рядка, без відступів «до» та «після» тексту;
- параметри сторінки: ліве поле – 3 см, верхнє та нижнє поля – 2 см, праве поле – 1 см;
- нумерація сторінок: в правому верхньому куті сторінки без крапки в кінці (крім титульної сторінки);
- вирівнювання тексту – по ширині;
- абзацний відступ повинен бути однаковим по всій роботі та дорівнювати 1,25 см.

Всі аркуші курсового проєкту (крім Титульного аркуша, Завдання та Анотацій) виконуються на аркушах формату А4, які містять так звану «рамку» або «штамп». Зразки цих аркушів наведено в Додатках Д - Е.

3.3. Вимоги до оформлення текстової частини

Структурними елементами курсового проєкту є розділи, підрозділи, пункти та підпункти. При виконанні курсового проєкту треба дотримуватись наступних правил:

- кожний розділ та спеціальні структурні елементи необхідно починати з нової сторінки;
- підрозділи, пункти та підпункти повинні починатися на поточній сторінці;

- не допускається наводити заголовок підрозділу або пункту в нижній частині сторінки, якщо після нього розміщується **тільки один рядок** тексту;
- відстань між заголовком і наступним або попереднім текстом повинна бути **два рядки**; між двома заголовками пропусків рядків **не повинно бути**;
- заголовки (назви) спеціальних структурних елементів необхідно розміщувати посередині рядка і наводити **прописними (великими)** літерами без крапки наприкінці, **не підкреслюючи та не виділяючи напівжирним шрифтом**;
- у розділів, підрозділів, пунктів і підпунктів повинні бути заголовки;
- у розділів повинна бути порядкова нумерація в рамках всієї роботи, яка позначаються арабськими цифрами без крапки (наприклад, 1, 2, 3 і т.д.);
- заголовки розділів необхідно розміщувати посередині рядка і наводити **прописними (великими)** літерами без крапки наприкінці, **не підкреслюючи та не виділяючи напівжирним шрифтом**;
- якщо заголовок містить два або більше речень, вони розділяються крапкою;
- нумерація підрозділів арабськими цифрами в рамках кожного розділу; номер підрозділу складається з номера розділу та порядкового номеру підрозділу, що розділені крапкою, причому після номеру підрозділу крапка не ставиться (наприклад, 1.1, 1.2, 1.3 і т.д.);
- заголовки підрозділів необхідно починати з абзацного відступу і друкувати маленькими літерами, крім першої великої без крапки в кінці, **не підкреслюючи та не виділяючи напівжирним шрифтом**;
- у пунктів повинна бути порядкова нумерація арабськими цифрами в рамках кожного підрозділу; номер пункту складається з номера розділу, номера підрозділу та порядкового номера пункту, що

розділяються крапками, причому після номеру пункту крапка не ставиться (наприклад, 1.1.1, 1.1.2, 1.1.3 і т.д.).

Перелік умовних позначень, скорочень і термінів

В цьому структурному елементі курсового проєкту необхідно навести і розшифрувати скорочення термінів і понять, які зустрічаються в тексті більше двох разів (крім загальноприйнятих). При цьому в самому тексті після першого згадування відповідного терміну або поняття необхідно навести в дужках його скорочення.

Скорочення необхідно наводити за абеткою спочатку кирилицею, а потім латиницею.

Якщо в тексті КП відсутні скорочення, даний структурний елемент **можна не наводити**.

Зміст

Зміст тексту необхідно оформлювати відповідно до ДСТУ 3008-2015 (Зміст даних методичних рекомендацій оформлено відповідно до цих вимог).

Рисунки

При наведені рисунків в тексті курсового проєкту необхідно притримуватись наступних вимог:

- рисунки необхідно розміщувати безпосередньо **після тексту**, де він згадується вперше (на цій же сторінці), або на наступній сторінці;
- на кожний рисунок має бути **посилання в тексті** курсового проєкту обов'язково з зазначенням його номера;
- рисунки повинні мати нумерацію арабськими цифрами та назву, яка обов'язково починається з **великої** літери; нумерація виконується:
 - або наскрізно по всій роботі;

- або номер рисунка складається з номера розділу, в якому міститься цей рисунок, та порядкового номера рисунка в цьому розділі, відділених крапкою;
- номер та назва рисунка розміщується **під ним посередині рядка**, наприклад, «Рисунок 2.3 – Структура обчислювального модуля»;
- блок-схеми алгоритмів (при необхідності їх наведення в КП) повинні оформлюватись як рисунки відповідно до ГОСТ 19.701-90 «Единая система программной документации. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения»;
- вище і нижче кожного рисунка повинен бути залишений **один вільний** (незаповнений) рядок.

Таблиці

Цифрові дані, отримані в результаті моделювання, обчислення тощо або інші результати проведеного дослідження бажано наводити у вигляді таблиць, які в загальному випадку мають вигляд, наведений на рисунку 3.1.

Таблица 1 – Назва таблиці

Головка таблиці						

Боковик
(колонка для
заголовків рядків)

Колонки

Заголовки
колонок
Підзаголовки
колонок

Рисунок 3.1 – Приклад таблиці

При наведені таблиць в тексті курсового проєкту необхідно притримуватись наступних вимог:

- таблиці необхідно розміщувати безпосередньо **після тексту**, де вони згадуються вперше, або на наступній сторінці;
- на кожну таблицю має бути **посилання в тексті** курсового проєкту обов'язково з зазначенням її номера;
- таблиці повинні мати нумерацію арабськими цифрами та назву, яка обов'язково починається з **великої** літери; нумерація виконується:
 - або наскрізно по всьому документу;
 - або номер таблиці складається з номера розділу, в якому міститься ця таблиця, та порядкового номера таблиці в цьому розділі, відокремлених крапкою;
- номер та назва таблиці розміщується **перед** таблицею **по центру рядка**, наприклад, «Таблиця 3.4 – Результати моделювання»;
- якщо всі рядки таблиці не вміщуються на одній сторінці, то таблиця розділяється на частини, при цьому частина таблиці переноситься на наступну сторінку, шапка таблиці повторюється в кожній частині таблиці (допускається замінювати шапку таблиці на наступній сторінці на нумерацію стовпців, які наводяться в першій частині таблиці арабськими цифрами; у випадку переносу таблиці над її частинами, що перенесені на наступну сторінку, **ліворуч без абзацного відступу** пишеться «Продовження таблиці 3.4» (тобто вказується номер перенесеної таблиці);
- **заголовки колонок** таблиці починають з **великої** літери, а **підзаголовки** – з **малої** літери, якщо вони становлять одне речення із заголовком;
- підзаголовки, які мають самостійне значення, подають з великої літери;
- у кінці заголовків і підзаголовків таблиць **крапки не ставлять**;
- переважна форма іменників у заголовках – однина;

- вище і нижче кожної таблиці повинен бути залишений **один вільний** (незаповнений) рядок.

Формули та рівняння

Формули та рівняння подають посередині рядка симетрично тексту окремим рядком безпосередньо після тексту, в якому їх згадано.

Вище та нижче кожних формули чи рівняння має бути **один пустий рядок** від попереднього та до наступного тексту.

Нумерують лише ті формули та/чи рівняння, на які є посилання в тексті КП або додатків.

Формули та рівняння у звіті, крім формул і рівнянь у додатках, треба нумерувати наскрізно арабськими цифрами. Дозволено їх нумерувати в межах кожного розділу аналогічно нумерації рисунків і таблиць.

Номер формули чи рівняння наводять на їх рівні праворуч у крайньому положенні в круглих дужках, наприклад, (3). У багаторядкових формулах або рівняннях їхній номер проставляють **на рівні останнього** рядка.

Пояснення параметрів або позначень, які входять до формули чи рівняння, треба подавати безпосередньо під формулою або рівнянням у тій послідовності, в якій їх наведено у формулі або рівнянні.

Пояснення позначень треба подавати без абзацного відступу з нового рядка, починаючи зі слова «де» без двокрапки. Познаки, яким встановлюють визначення чи пояснення, рекомендовано вирівнювати у вертикальному напрямку.

Приклад оформлення математичної формули.

Генеруючий многочлен $g(x)$ коду Ріда–Соломона має вигляд:

$$g(x) = (x - \alpha)(x - \alpha^2) \dots (x - \alpha^r), \quad (1)$$

де α - примітивний елемент поля $GF(N_\Omega)$;

r - кількість кодовекторів корекції спотворень у кодових словах.

Кілька наведених і не відокремлених текстом формул пишуть одну під одною і розділяють комами.

Приклад.

$$f_1(x,y) = S_1, \quad (2)$$

$$f_2(x,y) = S_2. \quad (3)$$

Числові значення величин з допусками наводять наступним чином:

мінус 5 В, плюс 5 В (або + 5 В);

$(17 \pm 3) \%$;

$75 \text{ мм} \pm 2 \text{ мм}$ або $(75 \pm 2) \text{ мм}$.

Діапазон чисел фізичних величин наводять, використовуючи прикметники «від» і «до». Наприклад, «від 3 мм до 7 мм» (а не від 3 до 7 мм).

У формулах і/чи рівняннях верхні та нижні індекси, а також показники степеня, в усьому тексті пояснювальної записки мають бути однакового розміру, але меншими за букву чи символ, якого вони стосуються.

Формули у вигляді рисунків не допускаються!

Переліки

Переліки можна наводити у будь-яких структурних елементах курсового проєкту (розділах, підрозділах, пунктах і підпунктах). Перед переліком ставлять двокрапку (крім пояснювальних переліків на рисунках).

Якщо використовуються переліки одного рівня підпорядкованості, на які в роботі немає посилань, то перед кожним із переліків ставлять знак «тире».

Якщо у КП використовуються переліки розвиненої та складної ієрархії, зазвичай використовуються ресурси текстових редакторів автоматичного створення нумерації переліків (наприклад, цифра – літера – тире (або літера)). При цьому текст кожної позиції необхідно починати з малої літери з абзацного відступу відносно попереднього рівня підпорядкованості.

Наприклад:

(1)_____;

(2)_____;

(3) _____:

(a) _____;

(b) _____;

(c) _____:

(i) _____;

(4) _____.

Посилання

У тексті пояснювальної записки курсового проєкту можна робити посилання на структурні елементи самого КП та інші джерела.

У разі посилання на структурні елементи самого документу зазначають відповідно номери розділів, підрозділів, пунктів, рисунків, формул, рівнянь, таблиць, додатків.

Посилання виконують, використовуючи такі вирази: «у розділі 1», «див. 2.1», «відповідно до 1.2.1», «(рисунок 2.5)», «відповідно до таблиці 3.4», «згідно з формулою (2.1)», «у рівняннях (1.2) – (1.5)», «(Додаток Г)» тощо.

Посилання на джерело інформації, наведене в переліку джерел посилання, рекомендовано подавати наступним чином: номер у квадратних дужках, за яким це джерело зазначено в переліку джерел посилання, наприклад, «у роботах [2] – [5]», «у роботах [2], [7], [21]».

Список використаних джерел

Список використаних джерел формується автором роботи за його вибором одним із таких способів:

- у порядку появи посилань у тексті;
- в алфавітному порядку прізвищ перших авторів або заголовків;
- у хронологічному порядку.

Бібліографічний опис списку використаних джерел оформлюється магістрантом відповідно до Національного стандарту України ДСТУ 8302:2015 «Інформація та документація. Бібліографічне посилання. Загальні положення та

правила складання», основні правила якого та приклади оформлення наведені у додатку Г.

Не допускається виконувати посилання на джерела, що не мають наукової цінності (наприклад, сторінки Вікіпедії тощо).

На всі джерела, вказані в списку, повинні бути посилання в основному тексті курсового проєкту. Всі публікації, які згадуються та цитуються в тексті проєкту, повинні бути в списку використаних джерел.

Бібліографічний опис використаного джерела може обмежуватися обов'язковою інформацією, необхідною для однозначної ідентифікації цього джерела.

Додатки

Додатки необхідно оформлювати на сторінках формату А4 і розташовувати після списку використаних джерел.

Кожний додаток повинен починатися з нової сторінки.

Додатки позначають послідовно **великими літерами української абетки**, крім літер Г, Є, З, І, Ї, Й, О, Ч, Ь, наприклад, ДОДАТОК А, ДОДАТОК Б. Якщо в роботі є тільки один додаток, він позначається як ДОДАТОК А.

Кожний додаток повинен мати заголовок, який друкується **зверху по центру великими літерами**. Посередині рядка над заголовком **великими літерами** необхідно надрукувати слово «ДОДАТОК ____» і великою літерою, що визначає саме цей додаток.

Лістинги програм

Лістинги розроблених програм в даному курсовому проєкті **наводити не потрібно**.

4. ТЕОРЕТИЧНІ ВІДОМОСТІ ДЛЯ ВИКОНАННЯ

КУРСОВОГО ПРОЄКТУ

4.1. Ідентифікація модулів на прикладі IP-адресації

У кожного хоста і маршрутизатора в Internet є IP-адреса, що складається з двох логічних частин: ідентифікатор мережі та ідентифікатор хоста-вузла. У заголовку IP-пакета для зберігання IP-адрес відводяться 2 поля довжиною по 4 байти (32 біти).

Найчастіше IP-адреси записують у вигляді чотирьох десяткових чисел, що представляють значення кожного байту, і розділених крапками, наприклад, 192.168.0.1.

Треба відзначити, що запис адреси не передбачає спеціального розмежовуючого знака між ідентифікатором мережі і хоста. Але при маршрутизації виникає необхідність розділити адресу на ці дві частини. Для цього використовуються такі наступні підходи.

- Використання фіксованої границі; при цьому 32-бітне поле адреси завчасно ділиться на дві частини фіксованої довжини, в одній з яких завжди розміщується ідентифікатор мережі, в іншій - вузла. Але в даному випадку, оскільки поле, що відводиться для зберігання номера вузла, має фіксовану довжину, всі мережі будуть мати однакове максимальне число вузлів. Такий підхід не дозволяє диференційно використовувати адресний простір, тому сьогодні майже не використовується.
- Використання маски, що дозволяє максимально гнучко встановлювати границі між ідентифікатором мережі і вузла. Тут адресний простір можна використовувати для створення багатьох мереж різного розміру. Маска - це 32 бітне число, що використовується разом з IP-адресою, причому двійковий запис маски містить безперервну послідовність одиниць в тих розрядах, що повинні в IP-адресі інтерпретуватися як номер мережі. Межа між послідовностями одиниць і нулів у масці

відповідає межі між ідентифікатором мережі і вузла в IP-адресі. Кількість одиниць в масці називають префіксом. Наприклад, масці 255.0.0.0 відповідає префікс 8.

- Використання класів адрес (повнокласова адресація). Вводиться п'ять класів адрес: А, В, С, D, Е. Три з них (А, В, С) використовуються для адресації мереж, а два (D, Е) мають спеціальне призначення. Для кожного класу мережних адрес визначено положення межі між номером мережі і номером вузла.

Ознакою, на основі якої IP-адреси відносять до того або іншого класу, є значення кількох перших біт адреси (ключа). На рисю 4.1 представлено структура IP-адреси.

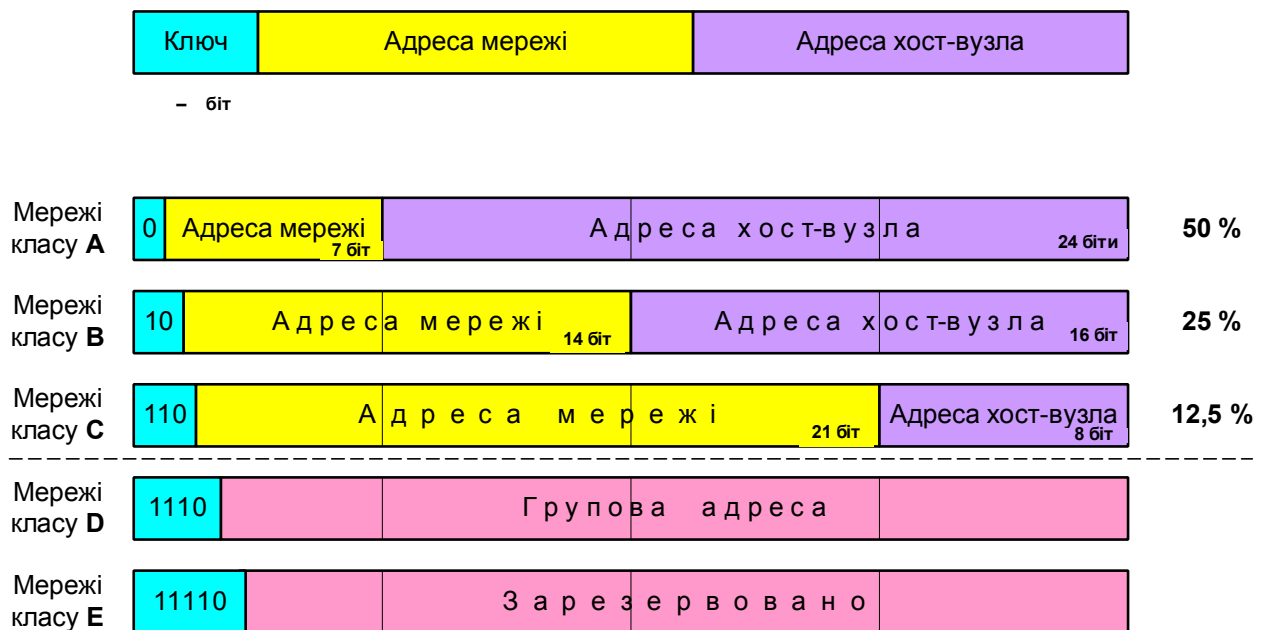


Рисунок 4.1. - Структура IP-адреси різних класів

У табдиці 4.1. наведено параметрів для IP-адрес всіх класів.

Щоб отримати із IP-адреси ідентифікатор мережі і вузла, треба розділити адресу на дві відповідні частини і доповнити кожен з них нулями до повних 4 байт. Наприклад, для адреси класу А 10.120.200.1 перший байт ідентифікує мережу, а останні три - вузол. Таким чином, ідентифікатор мережі є адреса 10.0.0.0, а вузла - адреса 0.120.200.1.

При призначені IP-адрес ідентифікатор мережі і вузла не можуть складатися лише з одних двійкових нулів й одиниць. Тому у табл. 4.1 максимальна кількість вузлів зменшена на 2, адреси 0 і 255 заборонено використовувати для адресації мережних інтерфейсів.

Таблиця 4.1 - Класи IP-адрес

Клас	Перші біти	Найменший-найбільший номер мережі	Маска	Кількість вузлів у мережі
A	0	1.0.0.0-126.0.0.0 (0 — не використовується, 127 — зарезервовано)	255.0.0.0	224—2, поле 3 байти
B	10	128.0.0.0-191.255.0.0	255.255.0.0	216—2, поле 2 байти
C	110	192.0.0.0-223.255.255.0	255.255.255.0	28—2, поле 1 байт
D	1110	224.0.0.0-239.255.255.255		Групові адреси
E	11110	240.0.0.0-247.255.255.255		Зарезервовано

Тут також треба пам'ятати, якщо у полі номер мережі стоять лише нулі, то за замовчуванням вважається, що вузол призначення належить тій же мережі, що і вузол-відправник. Така адреса може бути використана лише як адреса відправника. Якщо всі двійкові розряди IP-адреси дорівнюють 1, то пакет з такою адресою призначення має розсилатися всім вузлах, що знаходяться у цій же мережі, що і відправник пакету. Така адреса називається обмеженою широкомовною (limited broadcast). Обмеженість означає, що пакет не вийде за межі даної мережі. Якщо ж поле адреси отримувача у розрядах, що відповідають номеру хоста, містять лише 1, то пакет, що має таку адресу, розсилається всім вузлам мережі, номер якої вказано в адресі отримувача. Наприклад, пакет з адресою 192.168.1.255 буде відправлено всім вузлах мережі 192.168.1.0. такий тип адреси називається широкомовним (broadcast). При цьому треба розуміти, що у протоколі IP нема розуміння широкомовлення у тому змісті, як воно використовується в протоколах канального рівня, коли дані

повинні бути доставлені всім вузлах мережі. Як обмежена, так і звичайна широкомовна розсилка мають границі поширення у підмережі: вони обмежуються або мережею, до якої належить відправник пакету, або мережею, ідентифікатор якої вказано як отримувача. Тому поділ мережі на частини з допомогою маршрутизаторів локалізує широкомовний шторм межами однієї із підмереж лише тому, що нема способу адресувати пакет одночасно всім вузлам всіх підмереж мережі.

Використання масок при IP-адресації. Якщо разом з IP-адресою використовувати маску, то це дозволить виконувати адресацію більш гнучко і відмовитися від класової адресації. При цьому частину мережі називають підмережею. Для виділення підмережі маршрутизатору необхідно «накласти» маску підмережі на IP-адресу, що дозволить виділити в адресі ідентифікатори мережі, підмережі та кінцевого хост-вузла.

Наприклад, нехай для IP-адреси 172.16.155.5 задано маску 255.255.192.0. Двійковий вид відповідно такий:

- IP-адреса 10101100.00010000.10011011.00000101;
- маска 11111111.11111111.11000000.00000000.

Якщо ігнорувати маску й інтерпретувати адресу 172.16.155.5 на основі класів, то ідентифікатор мережі є 172.16.0.0, а вузла - 0.0.155.5 (оскільки адреса відноситься до класу B).

Якщо використовувати маску, то 18 послідовних одиниць в масці 255.255.192.0 після «накладання» на IP-адресу 172.16.155.5 ділять її на дві частини:

- ідентифікатор мережі 10101100.00010000.10;
- ідентифікатор вузла 011011.00000101.

У десятковій формі запис номера мережі і вузла після доповнення до 32 біт мають відповідно такий вигляд: 172.16.128.0 і 0.0.27.5.

Виділення з допомогою маски адреси мережі і хоста можна інтерпретувати як виконання логічної операції I (AND).

Для запису масок використовуються також такі формати запису з використанням префіксу, а саме: наприклад, 172.16.155.5/18 - 18 означає префікс, що вказує на кількість одиниць на початку маски.

4.2. Алгоритми маршрутизації потоків даних

Важливою функцією мережного рівня є вибір маршрута для передачі пакетів від початкового до кінцевого вузла. У більшості мереж пакетам необхідно проходити через декілька маршрутизаторів. Алгоритми вибору маршрута і структури даних, що ними використовуються, є головною метою при проєктуванні мережного рівня.

Алгоритм маршрутизації реалізується тією частиною програмного забезпечення мережного рівня, яка відповідає за вибір вихідної лінії для відправки вхідного пакету. Також треба розуміти, що маршрутизація і пересилання — це різні процеси. Пересилання полягає в опрацюванні вхідних пакетів і виборі для них вихідної лінії відповідно до таблиці маршрутизації. Маршрутизація відповідає за заповнення і оновлення таблиць маршрутизації, використовуючи для цього алгоритми маршрутизації.

Алгоритм вибору маршруту характеризують за такими властивостями: коректністю, простотою, надійністю (має коректно реагувати на зміни топології мережі та трафіка), стійкістю, справедливістю та оптимальністю. Всі алгоритми вибору маршрута можна поділити на два основних класи:

- адаптивні;
- неадаптивні.

Неадаптивні алгоритми при виборі маршрута не враховують топологію, поточний стан мережі і не змінюють трафік в каналах передачі. Вибір маршруту для пари хостів виконується заздалегідь, в автономному режимі, і список маршрутів завантажується в таблиці маршрутизаторів під час завантаження мережі.

Адаптивні алгоритми приймають рішення про вибір маршрутів при зміні топології і також в залежності від завантаженості каналів передачі. Адаптивні алгоритми відрізняються джерелами отримання інформації, моментами зміни

маршрутів (наприклад, через визначені однакові інтервали часу, при зміні навантаження на канал або при зміні структури зв'язку між комунікаційними вузлами) і даними, що використовуються для оптимізації (відстань, кількість транзитних частин або очікуваний час пересилання).

При виборі шляху від одного кінцевого вузла комп'ютерної мережі до іншого має використовуватися оптимальний маршрут. В одній мережі може бути декілька оптимальних маршрутів. Множина оптимальних маршрутів від всіх відправників до отримувачів називають вхідним деревом. Завдання всіх алгоритмів вибору маршрута є обчислення і використання вхідного дерева для всіх маршрутизаторів. Вхідне дерево не містить петель, тому кожен пакет доставляється отримувачу за обмежене число пересилань. Однак у реальних умовах лінії зв'язку і маршрутизатори можуть виходити з ладу під час виконання певних операцій. Тому різні маршрутизатори можуть мати різне уявлення про поточну топологію мережі.

4.3. Вибір найкоротшого шляху

Алгоритми вибору найкоротшого маршрута передачі використовуються у різних формах завдяки своїй простоті. Ідея полягає в побудові графу підмережі, у якого кожен вузол буде відповідати маршрутизатору, а кожна дуга - лінії зв'язку. Для вибору маршрута між двома маршрутизаторами алгоритм знаходить найкоротший шлях між ними на графі. Під найкоротшим шляхом зазвичай розуміють або кількість транзитних ділянок, або фізичну довжину лінії, або середню довжину черги і час затримки пересилання, або пропускну спроможність.

Відомо кілька алгоритмів знаходження найкоротшого шляху між двома вузлами графу. Один з них було запропоновано Едсгером Вібе Дейкстрою у 1959 р. Кожен вузол «помічається» відстанню до нього від вузла відправника за найменшим відомим шляхом. Спочатку шляхи невідомі, тому всі вузли вважаються недосяжними. Після того як починає працювати алгоритм, і знаходяться відстані, помітки вузлів змінюються і вказують на оптимальний

шлях. Після того як підтверджується, що помітка відповідає найкоротшому шляху, вона стає постійною і надалі не змінюється.

Для прикладу розглянемо зважений ненаправлений граф, що наведено на рис. 4.2, а. Вагові коефіцієнти для ребер відповідають відстані між маршрутизаторами. Потрібно знайти найкоротший шлях від A до G . Починаємо з вузла A , який позначимо чорним кругом. Далі досліджуємо всі зв'язані з ним вузли і вказуємо біля них відстань до вузла A . Якщо знайдеться більш короткий шлях до будь-якого вузла, то разом з відстанню у помітці біля вузла зміниться назва вузла, через який буде проходити найкоротший шлях. Таким чином, можна буде відновити весь шлях. Після розгляду всіх сусідів A помічаємо постійним той вузол, що є найближче. Це буде вузол B , він стає новим робочим вузлом. Відповідні позначки показано на рис. 4.2, б.

Далі повторюємо таку ж процедуру з вузлом B — досліджуємо всіх його сусідів. Якщо сума відстані від вузла B і значення помітки у вузлі B (відстань від A до B) виявиться меншим, ніж значення помітки у вузла, що досліджується (відстань до A , яку знайдено іншим шляхом), то це означає, що знайдено короткий шлях. Помітка вузла буде змінюватися.

Після дослідження всіх сусідніх з робочим вузлів і зміні тимчасових поміток, по всьому графу шукається вузол з найменшою поміткою. Цей вузол помічається як постійний і він стає поточним робочим вузлом. На рис. 4.2 показано перші 5 етапів роботи алгоритму.

Для прикладу розглянемо рис. 4.2, в. На даному етапі вузол D було помічено постійним. Нехай, наприклад, існує короткий шлях ніж ABD , наприклад, через уявну вершину U , тобто $AХUD$. У даному випадку можливі два випадки: або вузол U вже є постійним, або ще ні. Якщо це так, то вузол E вже перевірявся, тоді вузол U було призначено постійним і робочим вузлом. Тобто шлях $AХUD$ вже досліджувався.

Якщо вузол U помічено як тимчасовий, то помітка вузла U або більше чи дорівнює помітці вузла D , або менше неї. У першому випадку шлях $AХUD$ не може бути коротшим ніж шлях ABD . Якщо ж помітка вузла U менше помітки

вузла $Д$, тоді вузол $У$ повинен був стати постійним раніше вузла $Д$, і вузол $Д$ перевірявся б з вузла $У$.

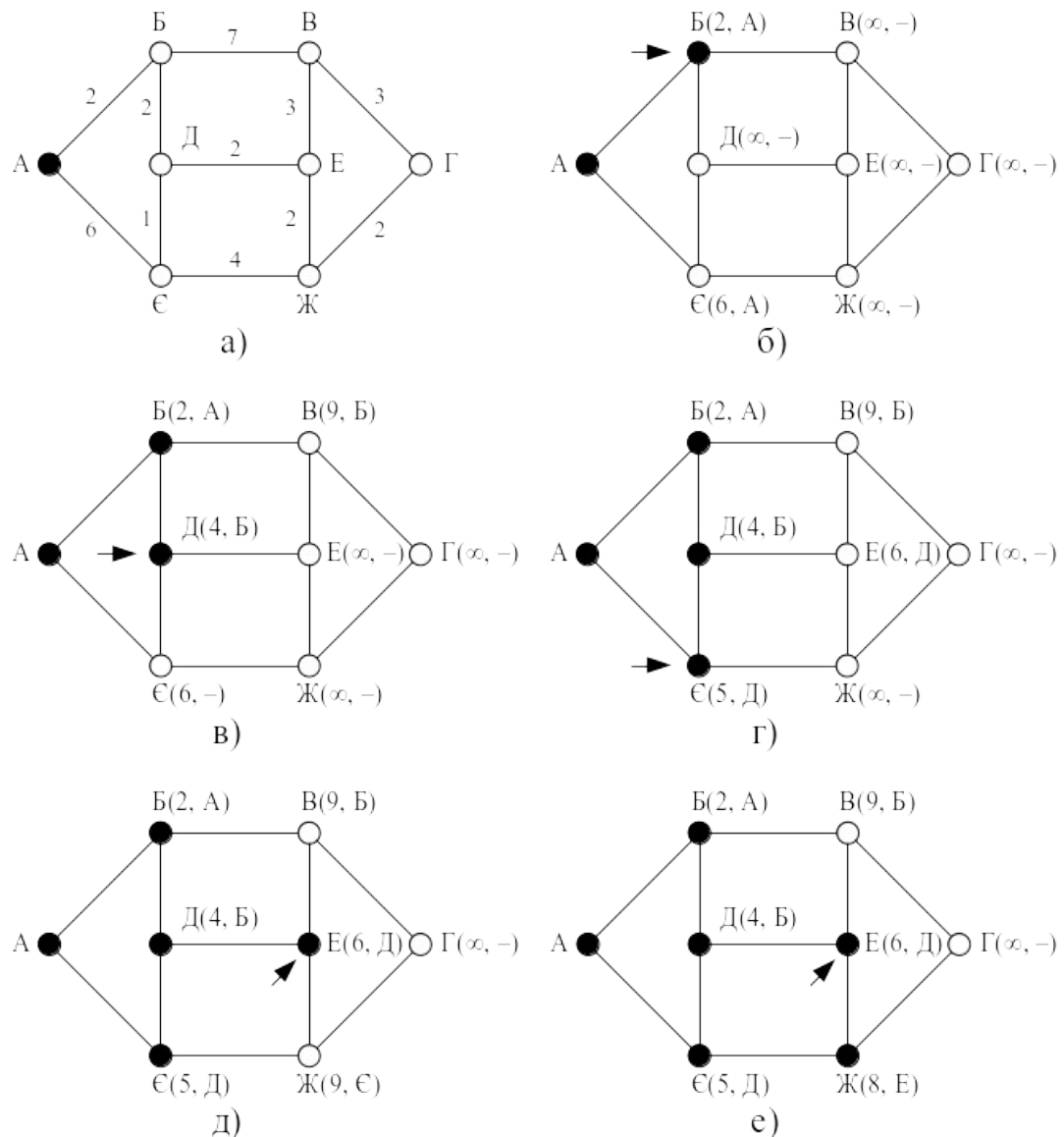


Рисунок 4.2 - Приклад обчислення найкоротшого шляху від А до Г з використанням алгоритму Дейкстри

4.4. Метод заливки

Цей алгоритм є статичним алгоритмом, при якому кожний пакет, що приходить маршрутизатор, відправляється на всі вихідні лінії, крім тієї, з якої він прийшов. Даний алгоритм породжує велику кількість дублікатів пакетів, якщо не використовувати ніяких додаткових засобів. Один з таких полягає в розміщенні в заголовку пакета лічильника, що буде рахувати кількість пройдених ділянок і зменшуватися на кожному маршрутизаторі. Коли значення

цього лічильника стає нульовим, пакет знищується. В ідеальному випадку даний лічильник має встановлюватися таким, що дорівнює довжині шляху від відправника до отримувача. Якщо відправник не знає відстані до отримувача, він може встановити значення лічильника, яке дорівнює довжині максимального шляху (діаметру) в даній підмережі.

Альтернативний спосіб обмеження кількості дублікатів полягає в підрахунку пакетів, що проходять через маршрутизатор. Це дозволяє не відправляти їх повторно. Наприклад, кожен маршрутизатор записує у кожний отриманий від своїх хостів пакет порядковий номер. Всі маршрутизатори мають список маршрутизаторів-джерел, у якому зберігаються всі порядкові номери пакетів, які їм зустрічаються. Якщо пакет від даного джерела з таким порядковим номером вже є у списку, то він далі не поширюється і знищується.

Для неможливості збільшення списку можна забезпечити для всіх списків лічильник k , що буде показувати, що всі порядкові номери включно з k вже зустрічалися. І коли приходить пакет, можна легко перевірити, чи він не є дублікатом.

На практиці зустрічається варіант даного алгоритму під назвою вибіркової заливки. У даному випадку маршрутизатори посилають пакети не на всі канали, а лише на ті, які приблизно йдуть у потрібному напрямку.

У більшості випадків даний алгоритм не використовується. Однак він застосовується там, де потрібна висока надійність, яка забезпечується надлишковим пересиланням пакетів.

4.5. Маршрутизація за вектором відстаней

Сучасні комп'ютерні мережі зазвичай використовують не статичні, а динамічні методи маршрутизації, оскільки статичні не враховують поточне завантаження мережі. Найбільш популярні два методи:

- маршрутизація за вектором відстаней;
- маршрутизація з врахуванням стану каналу.

Перший метод працює на основі таблиць (векторів), що підтримуються всіма маршрутизаторами і містять найкращі відомі шляхи для кожного з

можливих адресатів. Для оновлення даних у таблицях виконується обмін інформацією між сусідніми маршрутизаторами.

Алгоритм маршрутизації за вектором відстаней ще називають іменами його авторів Беллмана-Форда і Форда-Фалкерсона.

У даному випадку таблиці, з якими працюють і які оновлюють маршрутизатори, містять записи про кожен маршрутизатор підмережі. Кожен запис складається з двох частин: найпріоритетніший номер каналу для даного отримувача і відстань (або час проходження пакета) до даного отримувача. Одиницею вимірювання може слугувати число транзитних ділянок, мілісекунди, кількість пакетів, що чекають у черзі в даному напрямку тощо.

Вважається, що маршрутизаторам відома відстань до кожного з сусідів. Якщо за одиницю вимірювання використовується кількість транзитних ділянок, то відстань дорівнює одній транзитній ділянці. Якщо відстань вимірюється в одиницях часу затримки, то маршрутизатор може її визначати з допомогою спеціального ЕСНО (ехо) пакета, в який отримувач час отримання і який відправляється назад якомога швидше.

Нехай, наприклад, як одиниці вимірювання використовується час затримки, і для даного маршрутизатора цей параметр відомий для всіх сусідів. Через кожні T мілісекунд всі маршрутизатори відправляють своїм сусідам список із затримками для кожного отримувача. Кожен з них отримує такий список від всіх своїх сусідів. Наприклад, одна з таких таблиць прийшла від сусіда X , і в ній вказано, що час поширення від маршрутизатора X до маршрутизатора i дорівнює X_i . Якщо маршрутизатор знає, що час пересилання до маршрутизатора X дорівнює m , то затримка при передачі пакету маршрутизатору i через X дорівнює $X_i + m$. Після виконання таких обчислень для всіх своїх сусідів маршрутизатор може вибрати найкращий шлях і розмістити відповідний запис у нову таблицю. При цьому стара таблиця в обрахунках не використовується.

Процес оновлення таблиці показано на рис. 4.3. Зліва показано підмережу. Перші чотири стовпці показують вектори затримок, що отримує

маршрутизатор L від своїх сусідів. Маршрутизатор A вважає, що час передачі від нього до B дорівнює 12 мс, 25 мс - до маршрутизатора B і т.д. Нехай, маршрутизатор L оцінив затримки до своїх сусідів A , K , I і M відповідно як 8, 10, 12 і 6 мс.

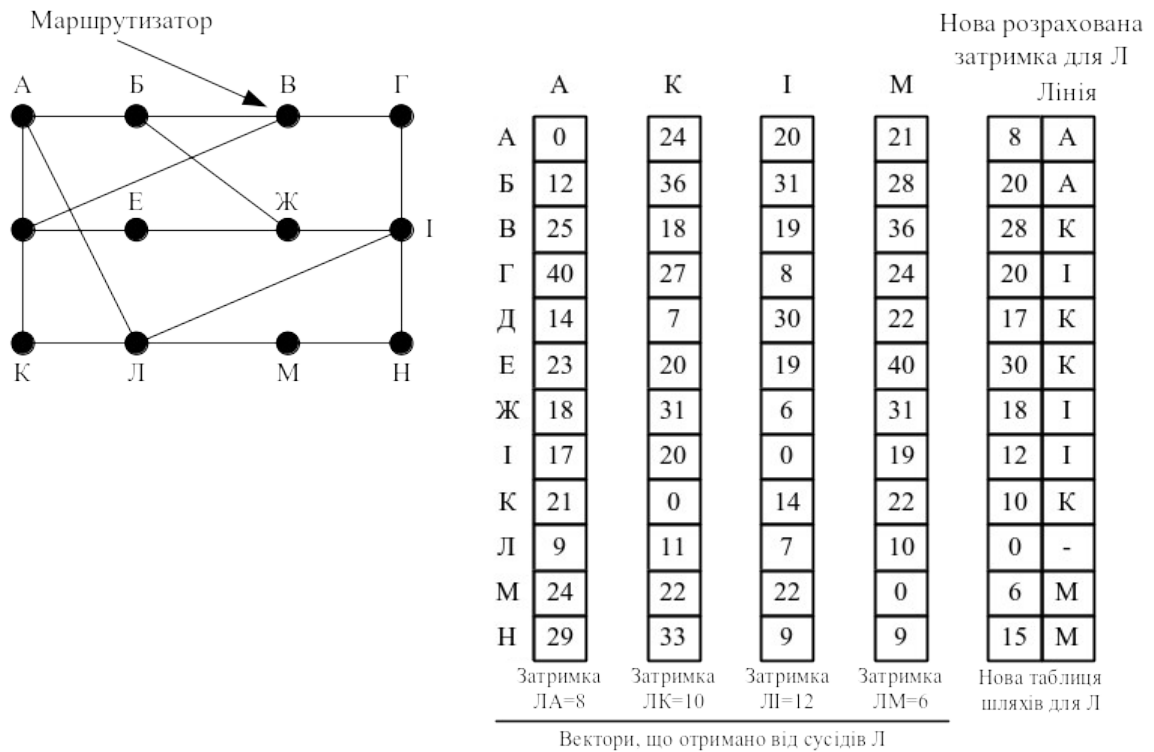


Рисунок 4.3 - Приклад виконання алгоритму Беллмана-Форда

Тепер розглянемо, як вузол L розраховує новий маршрут до маршрутизатора $Ж$. Він знає, що затримка до A складає 8 мс, A вважає, що від нього до $Ж$ дані буде доставлено за 18 мс. Таким чином, L знає, що коли він буде відправляти пакети для $Ж$ через A , то затримка складатиме 26 мс. Аналогічно він обчислює значення затримок для маршрутів від нього до $Ж$, що проходять через інших сусідів (K , I і M), й отримує відповідно 41 ($31+10$), 18 ($6+12$) і 37 ($31+6$) мс. Найкраще значення - це 18 мс, тому саме воно поміщається у стрічку таблиці для отримувача $Ж$. Разом з числом записується значення лінії, через яку проходить найкоротший шлях до $Ж$, тобто це I . Даний метод повторюється для всіх інших адресатів, і при цьому отримується нова таблиця, що показана як правий стовпець на рисунку 4.3.

Алгоритм маршрутизації за вектором відстаней працює теоретично, але має серйозний недолік на практиці: хоча правильна відповідь вкінці кінців знаходиться, процес її пошуку може зайняти дуже багато часу. Він швидко перераховує таблиці, якщо отримує «гарні» новини (наприклад, затримка стала меншою), але повільно після отримання «поганих» новин (наприклад, затримка стала більшою). Погані новини повільно поширюються - жоден маршрутизатор не може встановити значення відстані таким, що більше ніж на одиницю перевищує мінімальне значення цієї відстані, що зберігається у сусідів. Таким чином, всі маршрутизатори будуть до нескінченності збільшувати значення відстані до маршрутизатора, що «зник» у підмережі.

4.6. Маршрутизація з врахуванням стану лінії

Відмовитися від алгоритму на основі векторів відстаней довелося у 1979 р. з двох причин: по-перше, не враховувалася пропускна спроможність каналу, по-друге, алгоритм буде довго приходити у стійкий стан (проблема рахунку до нескінченності). Основою нового алгоритму була проста ідея, що ґрунтується на основі п'яти вимог до маршрутизатора, а саме кожен маршрутизатор повинен:

- знаходити своїх сусідів і дізнаватися їх мережні адреси;
- вимірювати затримку або вартість зв'язку з кожним зі своїх сусідів;
- створювати пакет, що містить всю зібрану інформацію;
- відправляти цей пакет всім маршрутизаторам;
- обчислювати найкоротший шлях до всіх маршрутизаторів.

У результаті кожному маршрутизатору відправляються повна топологія і всі виміряні значення затримок. Після цього для знаходження найкоротшого шляху до кожного маршрутизатора може використовуватися алгоритм Дейкстри.

4.7. Ієрархічна маршрутизація

Розмір таблиць маршрутизації, що підтримуються маршрутизаторами, збільшується пропорційно збільшенню розміру мережі. При цьому необхідно

мати значний об'єм пам'яті для її зберігання, більше часу для її опрацювання, крім того збільшується кількість службових пакетів. У певний момент мережа може збільшитися до таких розмірів, що зберігати на маршрутизаторах всю необхідну інформацію буде неможливо. Тому у великих мережах маршрутизація повинна виконуватися ієрархічно.

При використанні ієрархічної маршрутизації маршрутизатори розбиваються на окремі регіони. Кожен маршрутизатор знає всі деталі вибору маршрута в межах свого регіону, але нічого не знає про структуру інших регіонів.

У великих мережах може використовувати не лише дворівнева ієрархія. Може бути необхідне групування регіонів у кластери, кластерів у зони, зон у групи і т.д. Доведено, що оптимальна кількість рівнів ієрархії для підмереж, що складаються з N маршрутизаторів, дорівнює $\ln N$. При цьому потрібно $e \ln N$ записів для кожного маршрутизатора. Доведено, що збільшення довжини ефективного середнього шляху, що є наслідком використання ієрархічної маршрутизації, досить мале і зазвичай є прийнятним.

4.8. Широкомовна маршрутизація

У деяких застосуваннях хостам треба відправляти повідомлення на деяку кількість (групу) хостів або відразу на всі. Найбільш ефективно поширювати відповідні дані широкомовним способом, що надає можливість всім зацікавленим хостам отримувати їх. Широкомовною називається розсилка пакетів всім пунктам призначення. Для її реалізації використовуються різні методи.

Один з методів виконує розсилку окремих пакетів у всіх напрямках. Але це призводить до завантаження каналу. Крім того, відправнику треба мати повний список всіх хостів-отримувачів.

Тому використовується алгоритм багатоадресної маршрутизації. При використанні даного методу у кожному пакеті міститься або список адресатів, або бітова карта, що вказує на хости-отримувачі. Маршрутизатор, що отримує

такий пакет, перевіряє список, що міститься у пакеті, і визначає набір вихідних ліній. Далі створює копії пакету для кожної вихідної лінії. Весь список розсилки розподіляється між вихідними лініями. Після певної кількості пересилань кожен пакет буде містити лише одну адресу призначення.

Наступний алгоритм широкомовної маршрутизації використовує кореневе дерево або інше зв'язне дерево. Зв'язне дерево представляє собою підмножину підмережі, що містить у собі всі маршрутизатори, але не містить замкнутих шляхів. Якщо кожен маршрутизатор знає, які з його ліній належать зв'язному дереву, він може відправити вхідний пакет на всі лінії зв'язного дерева, крім тієї, з якої його отримав. Такий метод оптимально використовує пропускну спроможність мережі, утворює мінімальну кількість пакетів. Єдиною умовою є наявність у всіх маршрутизаторів інформації про зв'язне дерево.

Існує подібний широкомовний алгоритм, в якого може не використовуватися зв'язне дерево. У його основу покладено ідею просування зустрічним шляхом. Коли приходить широкомовний пакет, маршрутизатор перевіряє, чи використовується для передачі джерелу широкомовлення лінія, по якій прийшов пакет. Якщо відповідь позитивна, то велика ймовірність того, що широкомовний пакет прийшов по найкращому маршруту і є першою копією, яку отримав маршрутизатор. Тоді маршрутизатор розсилає цей пакет по всіх лініях крім тієї, по якій він його отримав. Але якщо пакет отримано від того ж джерела по іншій лінії, він відкидається як потенційна копія.

4.9. Класифікація протоколів динамічної маршрутизації

Як було зазначено раніше, маршрутизація представляє собою вибір напрямків (маршрутів) передачі даних від однієї мережі до іншої, від станції-відправника до станції-отримувача. Маршрути можуть задаватись безпосередньо адміністратором (статичні маршрути) або створюватись динамічно на основі інформації, отриманої від інших маршрутизаторів (динамічні маршрути). Інформація про маршрути зберігається в **таблиці**

маршрутизації. Основною перевагою динамічної маршрутизації є можливість оперативного реагування на стан мережі: появу нових сегментів, відмову каналів зв'язку і навіть вихід із ладу окремих маршрутизаторів. Якщо при налаштуванні невеликої мережі, що містить два або три маршрутизатори, статична маршрутизація є цілком прийнятним рішенням, то в мережах з десятками та сотнями маршрутизаторів альтернатива динамічній маршрутизації відсутня.

Так на рис. 4.4 показано мережу, що містить три маршрутизатори, нехай адміністратор налаштував обладнання таким чином, що маршрут з мережі А в мережу Б проходить через перший та другий маршрутизатори. Якщо канал між Router1 та Rt2 відмовить, то проходження даних за цим маршрутом стане неможливим, хоча, як видно з рисунку, існує альтернативний маршрут через Rt3. Однак для того, щоб цей маршрут став доступним адміністратор має внести зміни в маршрутні таблиці на Router1 та Rt2. При використанні динамічної маршрутизації зміни будуть зроблені автоматично протягом кількох хвилин, або навіть секунд.

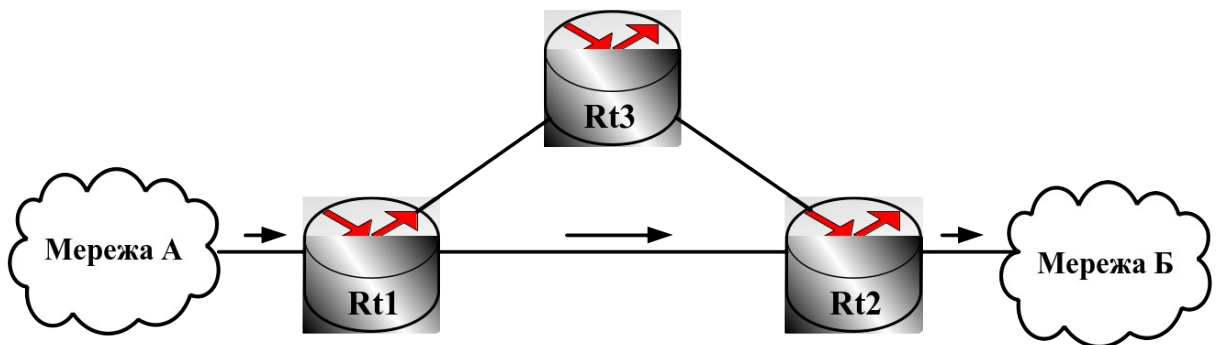


Рисунок 4.4 - Статичний маршрут

Успішне функціонування динамічної маршрутизації залежить від виконання маршрутизатором двох основних функцій:

- підтримки таблиці маршрутизації в актуальному стані;
- своєчасному розповсюдженні інформації про стан того чи іншого фрагменту мережі.

Для розповсюдження інформації про мережу використовується один або кілька протоколів маршрутизації. Протокол маршрутизації визначає набір правил, що використовуються маршрутизатором при здійсненні зв'язку з сусідніми маршрутизаторами, зокрема він визначає:

- яким чином розсилаються оновлення маршрутів;
- яка інформація міститься в оновленнях;
- з якою періодичністю відсилаються оновлення;
- яким чином визначаються адресати оновлень.

При формуванні таблиць маршрутизації та їх оновленні постає задача вибору найкращого маршруту. Кожний алгоритм маршрутизації використовує свій власний спосіб вибору найкращого маршруту. Для цього з кожним маршрутом асоціюється певне значення, яке називається **метрикою**. Як правило, менше значення метрики вказує на кращий маршрут.

Одні протоколи використовують прості метрики, що розраховуються на основі якогось одного параметру, наприклад, кількості проміжних маршрутизаторів. Існують також протоколи, що використовують так звані композитні метрики, які розраховуються на основі двох або більше параметрів маршруту. Для розрахунку метрик найчастіше використовують наступні параметри: смуга пропускання, затримка, рівень завантаження, надійність, кількість переходів, вартість.

Смуга пропускання (Bandwidth) — характеризує пропускну спроможність каналу, тобто кількість інформації, яка передається в одиницю часу (чим більше пропускну спроможність, тим менше значення метрики). Цей параметр є статичним і не змінюється в процесі роботи мережі.

Затримка (Delay) — час, який потрібен пакету для проходження каналом зв'язку; також статичний параметр і залежить тільки від фізичних особливостей реалізації каналу і параметрів інтерфейсів, що його утворюють.

Рівень завантаження (Load) — ступінь використання ресурсів каналу або маршрутизатора (динамічний параметр).

Надійність (Reliability) — характеризує рівень похибок, що виникають в каналі.

Кількість переходів (Hop count) — кількість маршрутизаторів, через які має пройти пакет для досягнення адресату.

Вартість (Cost) — довільне значення, що розраховується на основі ширини смуги пропускання або встановлюється безпосередньо адміністратором.

Коли мова йде про протоколи маршрутизації, слід чітко відрізнити протоколи маршрутизації від протоколів, що маршрутизуються.

Протокол маршрутизації - це засіб комунікації між маршрутизаторами, який дозволяє пристроям спільно використовувати інформацію про мережі і визначати відстань до окремих вузлів та мереж. Інформація, яку один маршрутизатор отримує від іншого за допомогою протоколу маршрутизації, використовується для створення або оновлення таблиці маршрутизації. Найпоширенішими протоколами маршрутизації є:

- RIP (Routing Information Protocol);
- OSPF (Open Shortest Path First);
- EIGRP (Enhanced Interior Gateway Protocol);
- IS-IS (Intermediate System To Intermediate System).

Базовими характеристиками, за якими адміністратор вибирає той чи інший протокол маршрутизації, є:

- час реакції на зміни в мережі (час конвергенції);
- тип метрики;
- складність налаштування та підтримки;
- протокол є відкритим чи пропрієтарним (належить фірмі розробнику).

Протоколи, **що маршрутизуються**, використовуються для доставки даних користувачів. Їх часто називають протоколами передачі даних. Слід звернути увагу, що протоколи маршрутизації для доставки своїх службових повідомлень використовують протоколи, що маршрутизуються. На

сьогоднішній день найпоширенішим протоколом, що маршрутизується, є протокол IP.

Як було зазначено вище, протокол динамічної маршрутизації повинен підтримувати в актуальному стані таблиці маршрутизації. Для цього він повинен оперативно реагувати на зміни в мережі. Очевидно, що швидкість реакції буде безпосередньо залежити від розміру мережі, яку він обслуговує. Саме тому досить часто адміністратори великих корпоративних мереж поділяють мережу на кілька зон маршрутизації. Більшість сучасних протоколів динамічної маршрутизації підтримують багатозонну маршрутизацію.

В той же час виникає інша задача - об'єднання кількох мереж, що належать різним власникам, і, як правило, використовують різні протоколи динамічної маршрутизації, в одну глобальну мережу з підтримкою маршрутизації між ними. Для вирішення цієї задачі введено поняття автономної системи (АС) та розроблено протоколи маршрутизації між АС.

Автономна система - це набір мереж, які знаходяться під єдиним адміністративним керуванням і в яких використовується єдина стратегія і правила маршрутизації. Як правило, питаннями маршрутизації між автономними системами займаються Internet-провайдери. Надання номерів автономним системам здійснюється централізовано організацією IANA аналогічно IP-адресації. На середину 2011 р. було зареєстровано більше 37 тисяч автономних систем.

Загальну класифікацію протоколів динамічної маршрутизації наведено на рис. 4.5.

За місцем застосування протоколи динамічної маршрутизації поділяються на два класи: **внутрішньої маршрутизації IGP (Interior Gateway Protocols)** та **зовнішньої маршрутизації EGP (Exterior Gateway Protocols)**. Протоколи класу IGP використовуються всередині автономних систем і призначені для розповсюдження інформації про наявні мережі. Інформація про зміни (поява нових та недосяжність існуючих мереж) передається іншим маршрутизаторам автономної системи також за допомогою протоколів IGP.

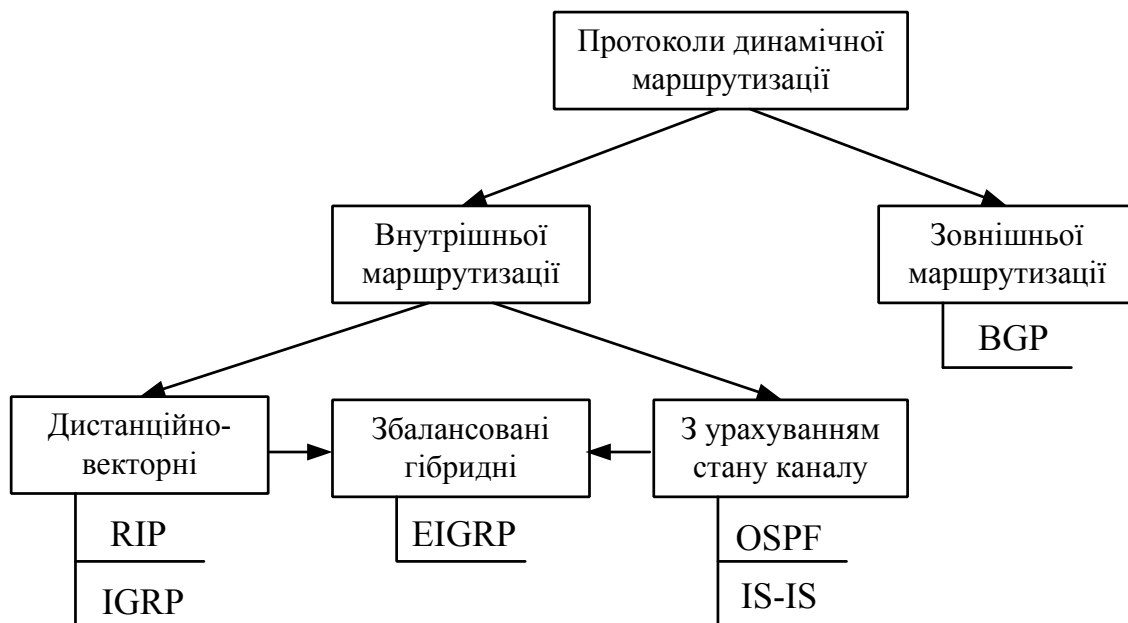


Рисунок 4.5 - Класифікація протоколів динамічної маршрутизації

Для того, щоб маршрутизаторам однієї АС стало відомо про мережі, що розташовані в інших АС, використовуються протоколи класу EGP. Маршрутизатори, що забезпечують зовнішню маршрутизацію, розташовані на границі автономних систем і передають іншим приграничним маршрутизаторам інформацію про наявні в межах АС мережі (рис. 4.6).

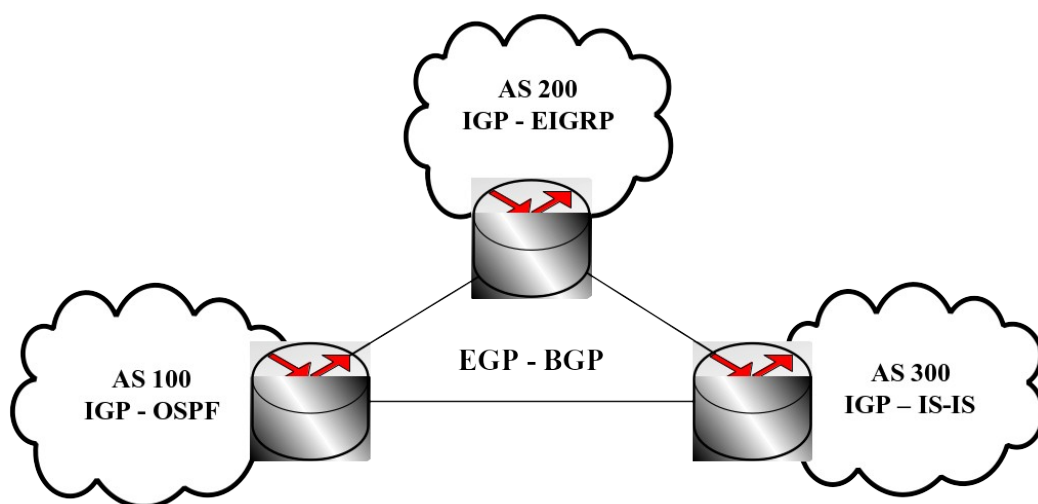


Рисунок 4.6 - Зовнішня та внутрішня маршрутизація

На сьогоднішній день функцію зовнішньої маршрутизації виконує єдиний протокол — BGP (Border Gateway Protocol).

За принципом функціонування протоколи класу IGP можна поділити на 2 базові типи:

- дистанційно-векторні протоколи;
- протоколи з урахуванням стану каналу.

Дистанційно-векторний протокол (distance vector routing protocol) визначає напрямок (вектор) та відстань до кожної мережі. **Протоколи з урахуванням стану каналу** LSR (link-state routing protocol) відтворюють топологію всієї мережі і на основі алгоритму SPF (shortest path first) розраховують оптимальний маршрут до кожної мережі. Існують також комбіновані протоколи, які називаються **збалансованими гібридними протоколами**, які поєднують певні риси базових класів.

Найвідомішим і, певно, найстарішим дистанційно-векторним протоколом є **протокол RIP** (Routing Information Protocol). Цей протокол є відкритим і підтримується мережним обладнанням всіх виробників, досить простий в налаштуванні. Як метрику використовує інформацію про кількість проміжних маршрутизаторів, що іноді призводить до неоптимальної маршрутизації. Існують дві версії RIP: RIPv1 та RIPv2, головна відмінність між якими полягає в тому, що RIPv1 в оновленнях не передає інформацію про маску.

Протокол IGRP (Interior Gateway Routing Protocol) також належить до групи дистанційно-векторних протоколів і був розроблений в середині 80-х років минулого століття фірмою Cisco. Головна відмінність від RIP - використання композитної метрики, яка розраховується на основі кількох параметрів маршруту. На сьогоднішній день вважається застарілим.

Протокол OSPF (Open Shortest Path First) аналогічно RIP відноситься до відкритих протоколів, однак базується на принципово інших підходах щодо визначення маршрутів. Кожний маршрутизатор на основі інформації про топологію всієї мережі розраховує оптимальні маршрути, використовуючи

алгоритм Дейкстри. Метрики розраховуються на основі пропускної спроможності каналів. Широко застосовується в гетерогенних мережах.

Протокол IS-IS (Intermediate System To Intermediate System) також є стандартним протоколом класу IGP і за принципом роботи дуже схожий на OSPF. Відмінності полягають в реалізації багатозонної маршрутизації. Крім того IS-IS підтримує більшу кількість маршрутів, що робить його привабливішим з точки зору масштабованості.

Протокол EIGRP (Enhanced Interior Gateway Routing Protocol) розроблений фірмою Cisco як альтернатива IGRP. Він об'єднує позитивні моменти як дистанційно-векторних протоколів, так і протоколів з урахуванням стану каналу.

4.10. Дистанційно-векторні протоколи маршрутизації

При використанні дистанційно-векторних протоколів маршрутизатори періодично обмінюються копіями таблиць маршрутизації (оновлення) один одному. Таким чином вони регулярно повідомляють своїх сусідів про зміни топології мережі. Дистанційно-векторні протоколи базуються на алгоритмі Белмана-Форда.

Проілюструємо роботу дистанційно-векторного протоколу на прикладі протоколу RIP. На рис. 4.7 наведено схему мережі, в якій для забезпечення маршрутизації використовується протокол RIP.

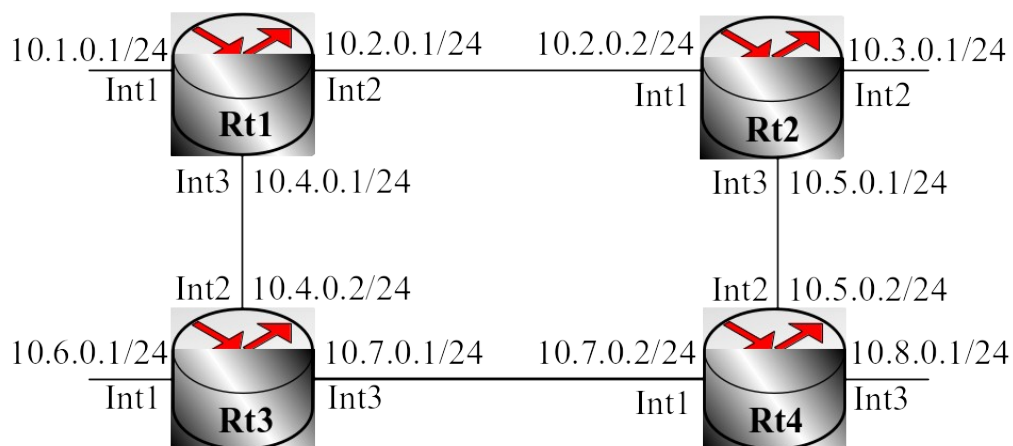


Рисунок 4.7 - Мережа для налаштування протоколу RIP

Формування таблиць маршрутизації при використанні протоколу RIP здійснюється в кілька етапів. Розглянемо цей процес на прикладі маршрутизатора Rt1.

На початковому етапі маршрутизатор заносить в таблицю маршрутизації інформацію про безпосередньо приєднані мережі. Це відбувається автоматично після налаштування інтерфейсів без залучення засобів статичної або динамічної маршрутизації. Таким чином по завершенні цього етапу таблиця маршрутизації Rt1 матиме вигляд, як показано в таблиці 4.2.

Таблиця 4.2 - Таблиця маршрутизації Rt1 на початковому етапі

Network	Mask	Interface	Next Router	Metric
10.1.0.0	255.255.255.0	Int1	-	0
10.2.0.0	255.255.255.0	Int2	-	0
10.4.0.0	255.255.255.0	Int3	-	0

Зауважимо, що поле Next Router (наступний маршрутизатор) порожнє, а поле метрики дорівнює 0, оскільки мережі безпосередньо приєднані. Нагадаємо, що протокол RIP за метрику використовує кількість проміжних маршрутизаторів.

На наступному етапі кожний маршрутизатор відправляє свої таблиці маршрутизації сусідам і отримує аналогічні таблиці від них. В оновленнях передається інформація про адреси мереж, маску та метрики, причому значення метрик збільшується на одиницю. Отримавши оновлення, маршрутизатор додає нові записи в таблицю маршрутизації. Звертаємо увагу, що в поле Next Router заноситься адреса відправника оновлення, а в поле Interface — ім'я інтерфейсу, через який оновлення отримано. Так для схеми на рис. 4.7 після отримання оновлень від Rt3 та Rt2 таблиця маршрутизації Rt1 набуде вигляду, наведеному в таблиці 4.3.

В тому випадку, коли на маршрутизатор надходить інформація про мережу, для якої вже є запис в таблиці маршрутизації, перевага надається

запису з меншою метрикою. Так, в наведеному прикладі оновлення про мережі 10.4.0.0 та 10.2.0.0, що надійшли відповідно від Rt3 та Rt2, будуть проігноровані.

Таблиця 4.3 - Таблиця маршрутизації Rt1 з першими оновленнями

Network	Mask	Interface	Next Router	Metric
10.1.0.0	255.255.255.0	Int1	-	0
10.2.0.0	255.255.255.0	Int2	-	0
10.4.0.0	255.255.255.0	Int3	-	0
10.4.0.0	255.255.255.0	Int3	10.4.0.2	1
10.6.0.0	255.255.255.0	Int3	10.4.0.2	1
10.7.0.0	255.255.255.0	Int3	10.4.0.2	1
10.2.0.0	255.255.255.0	Int2	10.2.0.2	1
10.3.0.0	255.255.255.0	Int2	10.2.0.2	1
10.5.0.0	255.255.255.0	Int2	10.2.0.2	1

Аналогічно відбувається формування маршрутних таблиць іншими маршрутизаторами. Так, після отримання перших оновлень в таблицях маршрутизації Rt3 та Rt2 з'явиться інформація про мережу 10.8.0.0 з метрикою 1, яка на наступному етапі розсилки оновлень потрапить на Rt1 і таблиця маршрутизації набуде вигляду, наведеному в таблиці 4.4.

Таблиця 4.4 - Таблиця маршрутизації Rt1 з другими оновленнями

Network	Mask	Interface	Next Router	Metric
10.1.0.0	255.255.255.0	Int1	-	0
10.2.0.0	255.255.255.0	Int2	-	0
10.4.0.0	255.255.255.0	Int3	-	0
10.6.0.0	255.255.255.0	Int3	10.4.0.2	1
10.7.0.0	255.255.255.0	Int3	10.4.0.2	1
10.3.0.0	255.255.255.0	Int2	10.2.0.2	1
10.5.0.0	255.255.255.0	Int2	10.2.0.2	1
10.8.0.0	255.255.255.0	Int3	10.4.0.2	2
10.8.0.0	255.255.255.0	Int2	10.2.0.2	2

Таким чином, для мережі, наведеної на рис. 4.7 процес формування таблиці маршрутизації завершиться після отримання двох оновлень. Звертаємо увагу, що мережа 10.8.0.0 представлена в таблиці двома записами, тобто до неї ведуть два маршрути з однаковими метриками. У такому випадку маршрутизатор буде одну частину трафіку пересилати одним маршрутом, а іншу другим. Така технологія називається балансуванням навантаження (load balancing).

Одна з основних функцій протоколу маршрутизації - оперативне реагування на зміни в топології мережі, які можуть бути додатними (поява нових мереж) та від'ємними (втрата доступу або відключення існуючих мереж). Реакція дистанційно-векторного протоколу на додатні зміни є прогнозованою - інформація про нові мережі потрапляє в чергові оновлення і розповсюджується на інші маршрутизатори. Що стосується від'ємних змін, то вони можуть суттєво ускладнити роботу протоколу і навіть привести мережу в непрацездатний стан.

У випадку, коли мережа 10.3.0.0 стала недоступною, очевидно, що Rt2 вилучить запис про неї зі своєї таблиці маршрутизації. Після цього надходить оновлення від Rt1, в якому є маршрут до мережі 10.3.0.0 (рис. 4.8).

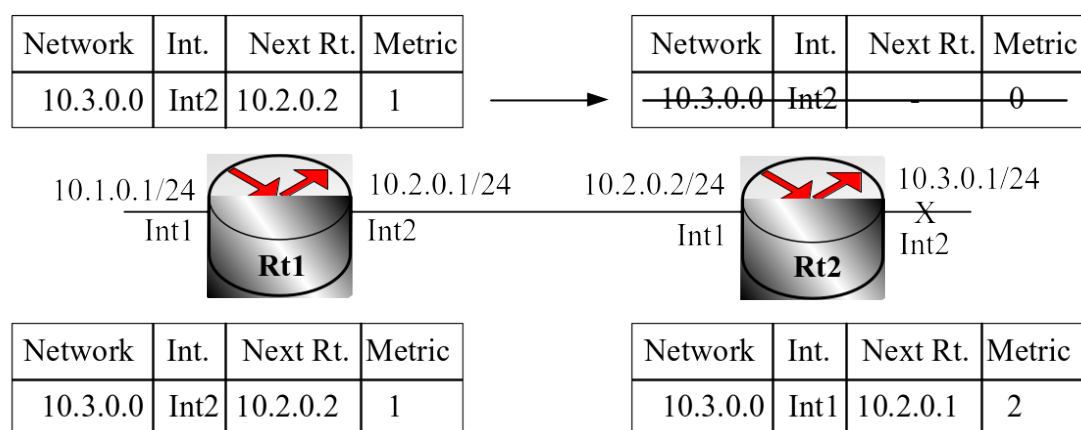


Рисунок 4.8 - Утворення маршрутних петель протоколом RIP

У даному випадку протокол RIP створив так звану маршрутну петлю — пакети, що прямують в мережу 10.3.0.0 будуть передаватись між Rt1 та Rt2 доки поле TTL не дорівнюватиме 0. Оскільки записи в таблиці маршрутизації

також обмежені в часі, настане момент, коли інформацію про мережу 10.3.0.0 буде вилучено з Rt1. Однак після надходження чергового оновлення з Rt2 цей запис знову з'явиться, тільки вже з метрикою 3 і т.д. Якщо не вжити якихось заходів, петля буди існувати нескінченно.

Для обмеження часу існування петель та запобігання їх утворенню розроблено низку методів, зокрема: обмеження максимального значення метрики, розщеплення горизонту, миттєве оновлення тощо. Максимальне значення метрики, а, відповідно і максимальна кількість послідовно розташованих маршрутизаторів в протоколі RIP обмежено числом 15. Метрика 16 використовується для інформування сусідів про недосяжність мережі.

Головна причина утворення маршрутних петель полягає в поширенні застарілої інформації. Так, в наведеному прикладі, Rt2 сприйняв як актуальну інформацію від Rt1 про мережу 10.3.0.0, яку, до речі, він раніше сам і надіслав до Rt1. Тобто Rt1 не був вчасно проінформований, що мережа 10.3.0.0 стала недосяжною, і в своїх оновленнях розповсюджував інформацію, що втратила свою актуальність.

Суть методу **розщеплення горизонту** (split horizon) полягає в тому, що маршрутна інформація про певну мережу ніколи не передається на той маршрутизатор, звідки її було отримано. Так, маршрутизатор Rt1 (рис. 4.7) в оновленнях, що будуть відсилатись через Int2, передаватиме тільки маршрути до мереж 10.1.0.0, 10.4.0.0, 10.6.0.0 та 10.7.0.0, а через Int3 — 10.1.0.0, 10.2.0.0, 10.3.0.0 та 10.5.0.0. Метод розщеплення горизонту вирішує проблему петель між двома сусідніми маршрутизаторами (рис. 4.8), однак петля може утворюватись трьома і більше маршрутизаторами. Так, в наведеному прикладі, застаріла інформація про мережу 10.3.0.0 може надійти через Rt3 та Rt4 і тоді петлю утворять всі чотири маршрутизатори.

Метод **миттєвих оновлень** (triggered update) полягає в тому, що отримавши інформацію про змінення метрики до певної мережі, маршрутизатор не чекає моменту розсилання періодичних оновлень, а відправляє оновлення терміново. Насправді передача здійснюється не миттєво,

а з затримкою у кілька секунд, тому певна ймовірність утворення петлі залишається.

Метод **тимчасового утримання від змін** (holddown) змушує маршрутизатори, яким стало відомо про недосяжність тієї чи іншої мережі, утриматись на певний час від прийняття оновлень про цю мережу від сусідніх маршрутизаторів. Припускається, що протягом відповідного періоду маршрутизатори вилучать зі своїх таблиць інформацію про недосяжну мережу, оскільки про неї не будуть надходити оновлення.

Не зважаючи на значний арсенал методів боротьби з маршрутними петлями стовідсоткової гарантії уникнення цього явища дистанційно-векторні протоколи не дають. Окрім маршрутних петель існують і інші недоліки дистанційно-векторних протоколів, зокрема обмежений діаметр мережі (15 послідовних маршрутизаторів для RIP), періодичне відсилання повної таблиці маршрутизації на широкомовну або групову адресу, що завантажує службовим трафіком канали зв'язку та ресурси отримувачів, використання як метрики кількості проміжних маршрутизаторів (протокол RIP). Саме тому протокол RIP здебільше використовується в невеликих локальних мережах, де зазначені недоліки не є критичними.

4.11. Протоколи маршрутизації з урахуванням стану каналу

Протоколи маршрутизації з урахуванням стану каналу принципово відрізняються від дистанційно-векторних. Головна відмінність — наявність в кожному маршрутизаторі інформації про **топологію всієї мережі**, на основі якої здійснюється розрахунок оптимальних маршрутів. Для визначення маршруту як правило використовується алгоритм Дейкстри, який ще називають алгоритмом вибору найкоротшого шляху (Shortest Path First — SPF). Для реалізації даного типу маршрутизації використовуються наступні компоненти.

Анонси стану каналу LSA (Link-State Advertisement) — невеликі пакети, що містять інформацію про параметри та стан безпосередньо приєднаних до маршрутизатора каналів.

Топологічна база даних, яка створюється безпосередньо кожним маршрутизатором на основі отриманих LSA.

Алгоритм вибору найкоротшого шляху використовується для знаходження оптимальних маршрутів до всіх мереж.

Таблиця маршрутизації містить інформацію про всі відомі мережі, інтерфейси, через які проходять найкращі маршрути та метрики цих маршрутів.

Процес створення таблиці маршрутизації засобами LSR протоколу містить наступні етапи.

1. Кожний маршрутизатор вивчає інформацію про всі безпосередньо приєднані мережі.
2. Кожний маршрутизатор відправляє в усі безпосередньо приєднані мережі пакет Hello для встановлення так званих сусідських відносин, в результаті чого формується база даних «сусідів».
3. Кожний маршрутизатор формує LSA і відправляє їх всім своїм сусідам.
4. Отримавши LSA, кожен маршрутизатор заносить цю інформацію до власної топологічної бази і відправляє анонс іншим сусідам.
5. Отримавши LSA від усіх маршрутизаторів, кожен маршрутизатор будує граф мережі, вершинами якого є маршрутизатори та локальні мережі, а ребрами — канали зв'язку між ними.
6. Застосовуючи алгоритм Дейкстри, кожен маршрутизатор будує логічну топологію у вигляді дерева, коренем якого є він сам, а гілками — мінімальні маршрути до інших вершин графу.
7. Маршрутизатор заносить до таблиці маршрутизації найкращі маршрути та порти, через які вони доступні.

Одним із найпоширеніших LSR протоколів є OSPF. Розглянемо процес формування таблиці маршрутизації на маршрутизаторі Rt1 (рис. 4.9).

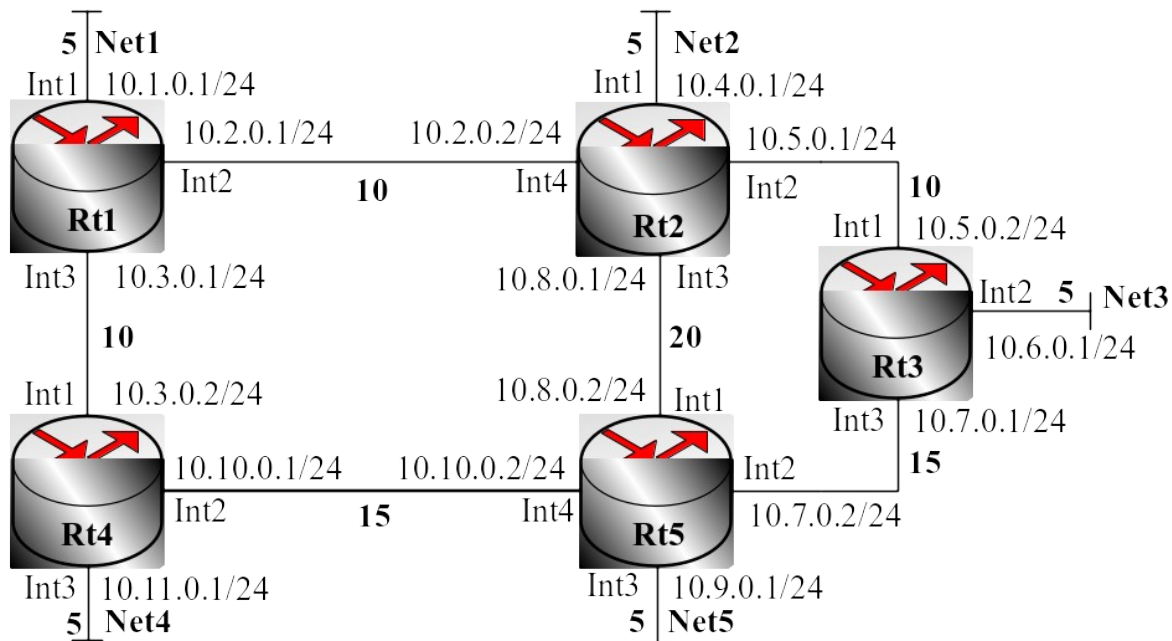


Рисунок 4.9 - Мережа для налаштування OSPF

По завершенні першого і другого етапів на Rt1 буде сформовано таблицю сусідів (табл. 4.5).

Таблиця 4.5 - Таблиця сусідів Rt1

Router	Interface	Network	Next Router	Metric
Rt2	Int2	10.2.0.0/24	10.2.0.2	10
Rt4	Int3	10.3.0.0/24	10.3.0.2	10
-	Int1	10.1.0.0/24	-	5

Аналогічні таблиці створюють і інші маршрутизатори. На основі отриманих таблиць формуються LSA і відправляються сусіднім маршрутизаторам (етап 3).

У процесі обміну LSA повідомленнями маршрутизатори формують топологічні бази (етап 4), які на завершальному етапі для всіх маршрутизаторів будуть однаковими і матимуть вигляд, наведений в таблиці 4.6.

Таблиця 4.6 - Топологічна база протоколу OSPF

	Rt1	Rt2	Rt3	Rt4	Rt5	Net1	Net2	Net3	Net4	Net5
Rt1		10		10		5				
Rt2	10		10		20		5			
Rt3		10			15			5		
Rt4	10				15				5	
Rt5		20	15	15						5

Етап 5 завершується формуванням графу мережі, який наведено на рис. 4.9.

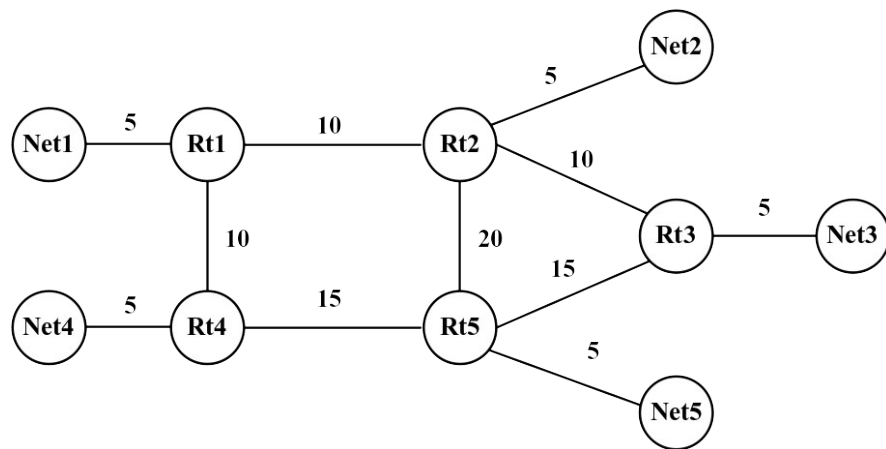


Рисунок. 4.10 - Граф, що відповідає мережі, наведений на рис. 4.9

Застосовуючи алгоритм Дейкстри, кожний маршрутизатор визначає оптимальні маршрути до кожної із мереж (етап 6). Результати виконання цього етапу для маршрутизатора Rt1 наведені на рис. 4.11.

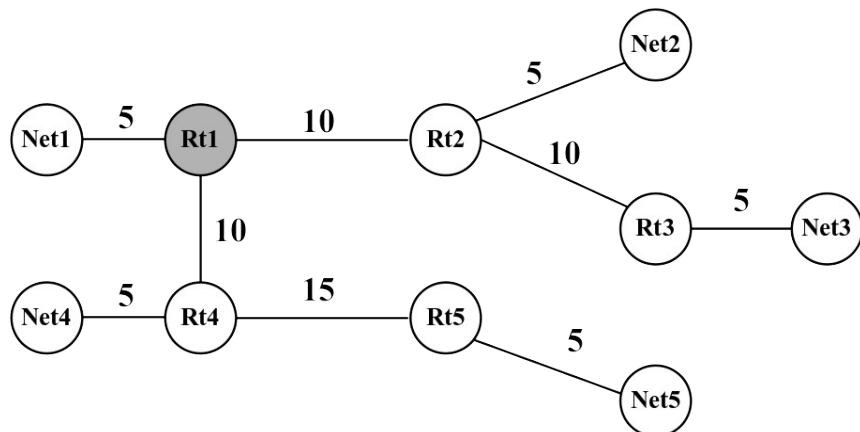


Рисунок. 4.11 - Логічна топологія мережі, побудована Rt1

З логічної топології генеруються записи для таблиці маршрутизації Rt1 (табл. 4.7).

Після завершення формування таблиці маршрутизації кожний маршрутизатор періодично (для протоколу OSPF кожні 10 с) відправляє своїм сусідам пакети Hello, які фактично підтверджують працездатність маршрутизатора та відповідних каналів зв'язку. Це єдиний службовий трафік, що генерується за відсутності змін в топології.

Якщо маршрутизатору стає відомо про зміни топології, як додатні (поява нових мереж) так і від'ємні (втрата зв'язку з існуючими мережами), генеруються відповідні LSA та відсилаються сусіднім маршрутизаторам, які, в свою чергу, розповсюджують цю інформацію далі. Отримавши інформацію про зміни в топології, кожний маршрутизатор відповідно модифікує свою топологічну базу та здійснює перерахунок маршрутів.

Таблиця 4.7 - Таблиця маршрутизації Rt1

Network	Mask	Interface	Next Router	Metric
10.1.0.0	255.255.255.0	Int1	-	5
10.2.0.0	255.255.255.0	Int2	-	10
10.3.0.0	255.255.255.0	Int3	-	10
10.4.0.0	255.255.255.0	Int2	10.2.0.2	15
10.5.0.0	255.255.255.0	Int2	10.2.0.2	20
10.6.0.0	255.255.255.0	Int2	10.2.0.2	25
10.7.0.0	255.255.255.0	Int2	10.2.0.2	35
10.8.0.0	255.255.255.0	Int2	10.2.0.2	30
10.9.0.0	255.255.255.0	Int3	10.3.0.2	30
10.10.0.0	255.255.255.0	Int3	10.3.0.2	25
10.11.0.0	255.255.255.0	Int3	10.3.0.2	15

Метрика в протоколі OSPF називається **вартістю** (cost). З кожним інтерфейсом маршрутизатора асоціюється певне значення вартості, яке може бути примусово встановлено адміністратором, або розраховано за замовчуванням на основі смуги пропускання. Більшому значенню смуги

пропускання відповідає менше значення метрики. Сумарна метрика маршруту визначається шляхом звичайного додавання метрик його окремих ланок.

Особливості роботи LSR протоколів висувають додаткові вимоги до обладнання, зокрема маршрутизатори повинні мати:

- достатню кількість оперативної пам'яті для зберігання топологічної бази, таблиці сусідів, таблиці маршрутизації тощо;
- потужний процесор, який має оперативно розраховувати оптимальні маршрути.

При роботі з LSR протоколами слід враховувати, що на етапах 3 та 4 відбувається лавиноподібне зростання кількості LSA в мережі, що може тимчасово знизити пропускну спроможність каналів зв'язку. Однак по завершенні етапу формування топологічних баз обсяг службового трафіку зводиться до мінімуму.

Оскільки будь-які зміни в топології мережі призводять до перерахунку оптимальних маршрутів усіма маршрутизаторами, що при використанні великої кількості маршрутизаторів та каналів зв'язку може ввести мережу в стан постійної конвергенції. Крім того, збільшення кількості маршрутизаторів збільшує розміри топологічної бази, і, таким чином, ускладнює задачу пошуку оптимальних маршрутів. Виходом з такої ситуації є застосування багатозонної маршрутизації, коли одна велика мережа поділяється на кілька окремих доменів маршрутизації.

5. ЕТАПИ ВИКОНАННЯ І ПОРЯДОК ЗАХИСТУ КУРСОВОГО ПРОЄКТУ

5.1. Графік виконання курсового проєкту

Для ефективного планування часу виконання курсового проєкту студенту надається календарний план-графік, який містить основні етапи КП та терміни їх виконання (табл. 5.1).

Таблиця 5.1 – Календарний план-графік виконання КП

Тиждень семестру	Назва етапу роботи	Навчальний час	
		Ауд.	СРС
4	<i>Отримання теми та завдання</i>	<i>Zoom</i>	2
5 - 6	<i>Підбор та вивчення літератури</i>	<i>Zoom</i>	4
7 - 8	<i>Виконання завдання 1</i>	<i>Zoom</i>	6
9 – 10	<i>Виконання завдання 2</i>	<i>Zoom</i>	6
11 – 13	<i>Виконання завдання 3</i>	<i>Zoom</i>	10
14 – 15	<i>Виконання завдання 4</i>	<i>Zoom</i>	7
	<i>Виконання завдання 5</i>	<i>Zoom</i>	6
16	<i>Подання курсового проєкту на перевірку</i>	<i>На поштову адресу</i>	2
17	<i>Захист курсового проєкту</i>	<i>Zoom</i>	2

Керівник здійснює контроль за ходом роботи, надає студенту необхідну консультативну допомогу при вивченні теоретичного матеріалу та розробленні програмного забезпечення, дає висновок про допуск роботи до захисту.

5.2. Порядок розгляду та захисту курсового проєкту

До захисту курсового проєкту допускаються студенти, які виконали його в повному обсязі, у встановлені терміни і представили відповідні матеріали керівнику.

У разі недопуску курсового проєкту до захисту студент повинен врахувати зауваження керівника КП та доопрацювати його з повторною подачею керівникові у встановлений термін.

Порядок розгляду, захисту курсового проєкту передбачає виконання наступних кроків.

1. Готовий курсовий проєкт (одним файлом) відправляється на перевірку викладачу на поштову адресу не пізніше ніж за тиждень до захисту.
2. Викладач перевіряє надісланий курсовий проєкт, робить зауваження та маркує ті місця роботи, на які необхідно звернути особливу увагу. При цьому:
 - **жовтим** кольором виділяються неточності, граматичні та синтаксичні помилки;
 - **червоним** кольором виділяються недоречності та принципові помилки, які свідчать про неправильне розуміння процесів передачі та моделювання або некоректні вхідні дані (або їх відсутність в пояснювальній записці).
3. Всі помилки обов'язково **необхідно виправити!**
4. Для захисту курсового проєкту обов'язково необхідно представити два варіанти КП: надіслану першу версію та виправлену. Курсовий проєкт захищається з представленням розробленої програми, **результатів проведених досліджень, обґрунтуванням отриманих результатів та зроблених висновків.**
5. Студент захищає курсовий проєкт, відповідаючи на запитання викладача. Виставляється остаточна оцінка.

УВАГА!

Курсові проєкти, які не захищені до **30 грудня**, можуть захищатись тільки в додаткову сесію в січні 2022 року!

5.3. Рейтингова система оцінювання курсового проєкту

При визначенні оцінки за курсовий проєкт враховується поточна робота студента над завданням на КП, результати роботи та її захисту, а саме:

- якість виконання окремих етапів КП, дотримання студентом затвердженого календарного плану виконання курсового проєкту;
- якість та своєчасність виконання курсового проєкту в цілому;
- відповідність оформлення курсового проєкту та пояснювальної записки до нього нормативним вимогам та технічному завданню;
- самостійність розробки;
- коректність, повнота та обґрунтованість висновків на основі отриманих результатів;
- правильність і аргументованість відповідей на запитання.

За дострокове та якісне виконання окремих етапів виконання курсового проєкту та роботи в цілому студенту можуть бути нараховані **заохочувальні бали**.

Рейтингова оцінка за курсовий проєкт (КП) включає:

- 1) бали за розроблення програмного продукту (ПП) – критерії оцінювання наведені у таблиці 5.1.

Таблиця 5.1

№	Критерій оцінювання	Бали
1.	Повнота та якість теоретичних положень	0 - 6
2.	Опис алгоритмі маршрутизації	0 - 6
3.	Наявність та якість самостійно створених програмних компонентів	0 - 6
4.	Опис структури розробленої програми	0 – 6
5.	Опис функцій та процедур, які реалізовані в розробленій програмі	0 - 6
	Всього (максимальна кількість балів):	30 балів

- 2) бали порівняльний аналіз отриманих результатів – критерії оцінювання наведені у таблиці 5.2.

Таблиця 5.2

№	Критерій оцінювання	Бали
1.	Наявність достатньої кількості тестів	0 - 8
2.	Наявність достатньої кількості таблиць результатів дослідження	0 - 8
3.	Порівняльний аналіз отриманих результатів	0 - 8
4.	Повнота, коректність та конкретність порівняльного аналізу результатів у висновках	0 - 8
	Всього (максимальна кількість балів):	32 бали

- 3) бали за розроблення технічної документації КП – критерії оцінювання наведені у таблиці 5.3.

Таблиця 5.3

№	Критерій оцінювання	Бали
1.	Повнота технічної документації, загальний вигляд та правильність структури пояснювальної записки	0 - 6
2.	Якість оформлення технічної документації, в тому числі технічного завдання	0 - 8
	Всього (максимальна кількість балів):	14 балів

- 4) бали за попередній захист КР – критерії оцінювання наведені у таблиці 5.4.

Таблиця 5.4

№	Критерій оцінювання	Бали
1.	Наявність та якість ПП, представленого до попереднього захисту	0 - 4
	Всього (максимальна кількість балів):	4 бали

- 5) бали за основний захист КР – критерії оцінювання наведені у таблиці 5.5.

Таблиця 5.5

№	Критерій оцінювання	Бали
1.	Повнота та якість доповіді, презентації та демонстрації ПП, ступінь володіння матеріалом	0 - 10
2.	Повнота та аргументованість відповідей на запитання	0 - 10
	Всього (максимальна кількість балів):	20 балів

6) **заохочувальні бали** – критерії оцінювання наведені у таблиці 5.6.

Таблиця 5.6

№	Критерій оцінювання	Бали
1.	Здача курсового проєкту за три тижні до закінчення семестру, якщо якість пояснювальної записки заслуговує не менше 70 балів	0 - 5
2.	Оригінальність та особлива докладність і обґрунтованість висновків	0 - 10
3.	Виконання курсового проєкту за індивідуальним завданням підвищеної складності	10 - 20
	Всього (максимальна кількість балів):	25 балів

P.S.

- Якщо студент з урахуванням заохочувальних балів отримує більше 100 балів, йому проставляється 100 балів.
- Заохочувальні бали **не нараховуються**, якщо виявлена несамотійність виконання курсового проєкту.

7) **штрафні** (від'ємні) бали.

Штрафні бали нараховуються, якщо:

- *КП не представлено вчасно до захисту (відповідно до графіку захистів): - 30 балів,*
- *якщо пояснювальна записка була здана несвоєчасно (таблиця 5.7).*

Таблиця 5.7

№	Термін здачі курсового проекту	Максимальна оцінка
1.	Курсовий проект здано до 06.12.2021	Студент може отримати оцінку відмінно (залежно від якості виконаної роботи)
2.	Курсовий проект здано в період 07.12.2021 – 13.12.2021	Студент може отримати оцінку не вище дуже добре , незалежно від набраних балів (залежно від якості виконаної роботи)
3.	Курсовий проект здано в період 14.12.2021 – 20.12.2021	Студент може отримати оцінку не вище добре , незалежно від набраних балів (залежно від якості виконаної роботи)
4.	Курсовий проект здано в період 21.12.2021 – 27.12.2021	Студент може отримати оцінку не вище задовільно , незалежно від набраних балів (залежно від якості виконаної роботи)
5.	Курсовий проект здано після 27.12.2021	Студент може отримати оцінку не вище достатньо , незалежно від набраних балів

P.S.

- у випадку виявлення несамотійності виконання курсового проекту бали за захист курсового проекту студенту не нараховуються, а також нараховується до **20 штрафних балів** залежно від рівня несамотійності або одразу проставляється оцінка Fx – незадовільно.

Максимальна кількість балів за курсовий проект (рейтингова шкала)
складає:

30 балів + 32 бали + 14 балів + 4 бали + 20 балів = **100 балів**.

Оцінка (ECTS та традиційна) за курсовий проект виставляється відповідно до набраних балів R_{KP} згідно з таблицею 5.8.

Таблиця 5.8

Сумарний рейтинг $R_{\text{кр}}$	Оцінка ECTS	Традиційна оцінка
95 - 100	A	Відмінно
85 - 94	B	Дуже добре
75 - 84	C	Добре
65 - 74	D	Задовільно
60 - 64	E	Достатньо
$R_{\text{кр}} \leq 59$	Fx	Незадовільно
Курсовий проєкт не представлено до захисту або не виконані умови допуску ¹	F	Не допущений

¹ Умови допуску до захисту:

- студент допускається до захисту, якщо його рейтинг за пояснювальну записку до курсового проєкту з урахуванням штрафних та заохочувальних балів складає не нижче 40 балів;
- у випадку, якщо поточний рейтинг студента за перший семестр на початок тижня захистів курсового проєкту складає не менше 75 балів (тобто Відмінно, Дуже добре або Добре), то він має право, не захищаючи курсового проєкту, отримати оцінку за КП «автоматом» відповідно до кількості сумарних балів за якість пояснювальної записки до курсового проєкту за якість звіту плюс 20 балів за захист.

СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. Комп'ютерні мережі: навчальний посібник [Текст] / Азаров О.Д., Захарченко С.М., Орлова М.М., Тарасенко В.П.. – Підручник. - Вінниця: ВНТУ. – 2020. – 385 с.
2. Комп'ютерні мережі: навчальний посібник [Текст] / Азаров О.Д., Захарченко С.М., Кадук О.В., Орлова М.М., Тарасенко В.П.. – Вінниця: ВНТУ. – 2013. – 371 с.
3. Таненбаум Э., Уэзерпол Д. Компьютерные сети. – СПб.: Питер. – 2012. – 960 с.
4. В.Г.Олифер, Н.А. Олифер. Компьютерные сети. Принципы, технологии, протоколы. Учебник. ПИТЕР. – 2020. – 467 с.
5. Є.Буров. Комп'ютерні мережі. Львів. – 2003. – 584 с.
6. Столлингс В. Компьютерные сети, протоколы и технологии Интернета. – СПб.: Питер. – 2005. – 832 с.
7. Стивенс У.Р. Протоколы TCP/IP. Практическое руководство / СПб.: "Невский Диалект" – "БХВ-Петербург", 2003. – 672 с.
8. Таненбаум Э. Компьютерные сети. – СПб.: Питер. – 2002. – 848 с.
9. Золотов С. Протоколы Internet. - Спб. : BHV. – 1998.
10. Семенов Ю.А. Протоколы и ресурсы Internet.: М. – 1996.
11. Семенов Ю.А. Сети Internet. Архитектура и протоколы.: М. – 1998.
12. Семенов Ю.А. Протоколы Интернет. Энциклопедия.: М.: Горячая линия-Телеком. – 2001.
13. Максим Кульгин. Технологии корпоративных сетей. ПИТЕР. – 1999.
14. Максим Кульгин. Коммутация и маршрутизация IP/IPX трафика. Монография. КомпьютерПресс. – 1998.
15. Максим Кульгин. Практика построения компьютерных сетей. Для профессионалов. – 2001.
16. Мэтью Ногл. TCP/IP. Иллюстрированный учебник. М.: ДМКПресс. - 2001. – 465 с.

17. Фейт С. TCP/IP. Архитектура, протоколы, реализация. М.: Лори. – 2000.
18. Тодд Леммл, Моника Леммл, Джеймс Челлис. Учебное руководство для специалистов MCSE: TCP/IP. /Пер. с англ. – М.: Изд-во «Лори».- 1997.
19. Кулаков Ю.А., Луцкий Г.М. Компьютерные сети. Киев. – 1998.
20. К.Джамса, К.Коуп. Программирование для INTERNET в среде Windows. Санкт-Петербург: ПИТЕР. - 1996.
21. Шон Уолтон. Создание сетевых приложений в среде Linux. Руководство разработчика. М.-Спб-Киев. Вильямс. – 2001.

Допоміжна література

1. Столлингс Вильям. Компьютерные системы передачи данных. 6-е издание.: М.: Издательский дом "Вильямс". – 2002. – 928 с.
2. Блэк Ю. Сети ЭВМ. Протоколы, стандарты, интерфейсы: Пер. с англ. – М.: Мир, - 1990.
3. Шварц М. Сети связи: протоколы, моделирование и анализ: В 2-х ч.: Пер. с англ. – М.: Наука. – 1992.
4. Шварц М. Сети ЭВМ. – М.: Наука. – 1990.
5. Пайк М. Internet в подлиннике: Пер. с англ. – СПб.: BHV – Санкт-Петербург, 1996.
6. Лаура Ф. Чаппел, Дэн Е. Хейкс. Анализатор локальных сетей NetWare (Руководство Novell). Москва: ЛОРИ. - 1995.
7. Куроуз Дж., Росс К. Компьютерные сети. – СПб.: Питер. – 2004. – 765 с.
8. Положення про випускню атестацію студентів КПІ ім. Ігоря Сікорського [Електронний ресурс] / Уклад.: В. П. Головенкін, В. Ю. Угольніков. – Київ : КПІ ім. Ігоря Сікорського, 2018. – 98 с.
9. ДСТУ 8302:2015. Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання. // [Електронний ресурс]. Режим доступу: http://www.knmu.kharkov.ua/attachments/3659_3008-2015.PDF

10. Звіти у сфері науки і техніки. Структура та правила оформлювання. ДСТУ 3008:2015 // [Електронний ресурс]. Режим доступу: http://www.knmu.kharkov.ua/attachments/3659_3008-2015.PDF
11. ГОСТ 19.701-90 «Единая система программной документации. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения» // [Електронний ресурс]. Режим доступу: <http://www.gostrf.com/normadata/1/4294848/4294848992.pdf>

ДОДАТКИ

ДОДАТОК А. Зразок оформлення титульного аркуша курсового проєкту

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

імені ІГОРЯ СІКОРСЬКОГО»

Кафедра системного програмування і
спеціалізованих комп'ютерних систем

КУРСОВИЙ ПРОЄКТ

з дисципліни «Комп'ютерні мережі»

на тему: Маршрутизація в мережі передачі даних

Студента (ки) IV курсу, групи КВ-XX

за спеціальністю

123 «Комп'ютерна інженерія»

Іванов І.І.

Керівник: доцент, к.т.н., доц. Орлова М.М.

Національна оцінка _____

Кількість балів: _____ Оцінка: ECTS _____

Члени комісії _____ доцент, к.т.н. Орлова М.М.

_____ доцент, к.т.н. Щербина О.А.

Київ- 202X рік

ДОДАТОК Б. Завдання на курсовий проєкт**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»****ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ**

Кафедра системного програмування і спеціалізованих комп'ютерних систем

Рівень вищої освіти – перший (бакалаврський)

Спеціальність 123 «Комп'ютерна інженерія»

**ЗАВДАННЯ
на курсовий проєкт студента**

(прізвище, ім'я, по батькові)

1. Тема проєкту _____

керівник проєкту _____,

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

2. Термін подання студентом проєкту _____

3. Вихідні дані до проєкту _____

4. Зміст пояснювальної записки _____

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка

Студент

(підпис)

(ім'я, ПРІЗВИЩЕ)

Керівник проєкту

(підпис)

(ім'я, ПРІЗВИЩЕ)

ДОДАТОК В. Приклад оформлення анотації

АНОТАЦІЯ

Кваліфікаційна робота включає пояснювальну записку (58 с., 21 рис., 3 табл., список використаної літератури з 18 найменувань, 3 додатки).

Метою курсового проєкту є розробка системи моделювання та дослідження та структуризація теоретичних відомостей, необхідних для розробки та реалізації онлайн-гри, і на основі отриманих даних розробити власну онлайн-гру клієнт-серверної архітектури.

Для досягнення поставленої мети розроблено та проведено

В результаті роботи було програмно реалізовано, в якій було використано оптимальні та спеціалізовані під певну задачу алгоритми. Перевагою даного проєкту є

Результати курсового роботи можуть бути використанні при

При розробці було використано наступні ресурси:

Ключові слова: (5 – 15 ключових слів).

ДОДАТОК Г. Правила оформлення списку літератури

Приклади оформлення списку літератури
згідно ДСТУ 8302:2015 «Інформація та документація. Бібліографічні
посилання. Загальні положення та правила складання»

Характеристика джерела	Приклад оформлення
Книги	
Один автор	Таненбаум Э. Современные операционные системы. СПб.: Питер, 2019. 865 с.
Два автори	Таненбаум Э., Уэзеролл Д. Компьютерные сети : Изд. 5-е. СПб.: Питер, 2019. 891 с.
Три автори	Назаров А. Н., Разживин И. А., Симонов М. В. АТМ: Технические решения создания сетей / Под ред. А. Н. Назарова. М.: Горячая линия-Телеком, 2001. 376 с.
Чотири автори	ISDN просто и доступно / Титтель Эд, Джеймс Стив, Пискителло Дэвид, Пфайфер Лайза. Изд-во «ЛЮРИ», 1999. 282 с.
П'ять і більше авторів	Комп'ютерні мережі : навчальний посібник / О. Д. Азаров та ін. Вінниця : ВНТУ, 2013. 371 с.
Багатотомне видання	Сети TCP/IP : у 3 т. / Камер Дуглас Э. М. : Издательский дом «Вильямс», 2003. Т. 1. 880 с. Сети TCP/IP : у 3 т. / Камер Дуглас Э., Стивенс Дэвид Л. М. : Издательский дом «Вильямс», 2002. Т. 3. 592 с.
Частина видання	
Розділ книги	Филимонов А.Ю. Построение мультисервисных сетей Ethernet. СПб.: БХВ-Петербург, 2007. С. 147–174.
Тези доповідей, матеріали конференцій	Орлова М. М., Багінський Є. С. Оптимізація балансування навантаження в стільникових мережах lte/te-a. Збірник тез доповідей Десятої наукової конференції «Прикладна математика та комп'ютинг ПМК 2018», 21–23 березня 2018 р.

Характеристика джерела	Приклад оформлення
	Київ : КПІ, 2018. С. 87–91. Молодецька К. В., Сугоняк І. І. Особливості комп'ютерного моделювання фізичних полів і процесів на основі прямих і зворотних диференціальних спектрів із значною кількістю дискрет. Інформаційно-комп'ютерні технології 2010 : V міжнародна наук. – тех. конф. : 20–22 трав. 2010 р. : тези. – Житомир : ЖДТУ, 2010. С. 12–13.
Статті з журналів та інших періодичних видань	Морозов К. В., Романкевич В. О., Романкевич О. М. О характере влияния модификации рёберных функций GL-модели на её поведение в потоке отказов. Радиоелектронні і комп'ютерні системи. 2016. № 6. С. 108–112. Калиновский Я. А., Бояринова Ю. Э., Сукало А. С. Исследование алгебраических и функциональных свойств обобщенных гиперкомплексных систем четвертой размерности. Реєстрація, зберігання і обробка даних. 2017. Т. 19, № 1. С. 22–33.
Електронні видання	
Книги	Ілляшенко С. М., Шипуліна Ю. С. Товарна інноваційна політика : підручник. Суми : Університетська книга, 2007. 281 с. URL: ftp://lib.sumdu.edu.ua/Books/1539.pdf (дата звернення: 10.11.2017).
Законодавчі документи	Закон про освіту : Закон України від 05 вересня 2017 р. № 2145-VIII. URL: https://zakon.rada.gov.ua/laws/show/2145-19 (дата звернення 25.08.2019). При стандартизацію : Закон України від 11 лют. 2014 р. № 1315. URL: https://zakon1.rada.gov.ua/laws/show/1315-18 (дата звернення 02.11.2017).
Періодичні видання	Марченко О. І., Марченко О. О., Щербина Б. О. Побудова дерева пошуку способом з використання методу Монте-Карло

Характеристика джерела	Приклад оформлення
	і контролем форми дерева. Вісник Вінницького політехнічного інституту. 2017. № 4. С. 65–69. URL: https://visnyk.vntu.edu.ua/index.php/visnyk/article/view/2096/2062 (дата звернення: 12.10.2018). Athey S., Parashkevov I., Sarukkai V., Xia J. Bitcoin Pricing, Adoption, and Usage: Theory and Evidence. Stanford University Graduate School of Business Research Paper, 2016. № 16–42. P. 70.
Сторінки з веб-сайтів	Бездротові мережі передачі даних. Integrity Systems : URL: http://integritysys.com.ua (дата звернення: 26.08.2019).
Інші документи	
Стандарти	ДСТУ 3008:2015. Звіти у сфері науки і техніки. Структура та правила оформлювання. [Чинний від 2015-06-22]. Київ, 2016. 31 с. (Інформація та документація).
Патенти	Мобільний комп'ютерний термінал «к.врт800» : пат. 123194 Україна : МПК H04N 1/00 , G06K 9/00 . № u 2017 10271 ; заявл. 24.10.2017 ; опубл. 12.02.2018, Бюл. № 3.
Дисертації, автореферати дисертацій	Романкевич В.О. Методи та засоби оцінки технічних характеристик гарантоздатності відмовостійких багатопроцесорних систем управління складними об'єктами : дис. на здобуття наук. ступеня д-ра техн. наук : 05.13.05 / Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, 2017. Романкевич В.О. Методи та засоби оцінки технічних характеристик гарантоздатності відмовостійких багатопроцесорних систем управління складними об'єктами : автореф. дис. на здобуття наук. ступеня д-ра техн. наук : 05.13.05. Київ, 2017. 41 с.

ДОДАТОК Д. Технічне завдання

ЗМІСТ

1.	НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ.	2
2.	ПІДСТАВА ДЛЯ РОЗРОБКИ.	2
3.	ЦІЛЬ І ПРИЗНАЧЕННЯ РОБОТИ.	2
4.	ДЖЕРЕЛА РОЗРОБКИ.	2
5.	ТЕХНІЧНІ ВИМОГИ.	2
5.1.	Вимоги до програмного продукту, що розробляється.	2
5.2.	Вимоги до апаратного забезпечення.	3
5.3.	Вимоги до програмного та апаратного забезпечення користувача. .	3
6.	ЕТАПИ РОЗРОБКИ.	4

					ІАЛЦ. 467200.004 ПЗ						
	З	А	№ докум.	Підпис							
Виробив			Микитенко С.С.		ата	<i>Маршрутизація в мережі передачі даних. Пояснювальна записка</i>		Літ.	Ар.	Архівів	
Перевірив			Орлова М.М.						куш	1	xx
Н. контроль											
Затвердив								КПІ ім. Ігоря Сікорського, ФПМ КВ-8Х			

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ

Назва розробки: «Маршрутизація в мережі передачі даних».

Галузь застосування: створення систем «Розумного дому», що керуються через один мобільний додаток.

2. ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на виконання курсового проєкту, затверджене кафедрою системного програмування і спеціалізованих комп'ютерних систем Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського».

3. Мета І ПРИЗНАЧЕННЯ РОБОТИ

Метою даного проєкту є створення мобільного клієнта під операційну систему Android для управління та моніторингу систем «розумного дому».

4. ДЖЕРЕЛА РОЗРОБКИ

Джерелом інформації є технічна та науково-технічна література, технічна документація, публікації в періодичних виданнях та електронні статті у мережі Інтернет.

5. ТЕХНІЧНІ ВИМОГИ

5.1. Вимоги до програмного продукту, що розробляється

- сумісність з операційною системою Android;
- можливість підключення пристроїв розумного дому;
- можливість управління пристроями розумного дому;
- можливість отримання інформації від пристроїв розумного дому;
- можливість відключення пристроїв домашнього розуму;

- наявність зручної системи вибору команди управління;
- прив'язка ір та порта до пристрою розумного дому;
- зберігання встановленого зв'язку з пристроєм розумного дому.

5.2. Вимоги до апаратного забезпечення

- оперативна пам'ять: 1 Гб;
- наявність доступу до мережі Wi-Fi (IEEE 802.11 b/g/n).

5.3. Вимоги до програмного та апаратного забезпечення користувача

- операційна система Android;
- наявність доступу до мережі Wi-Fi (IEEE 802.11 b/g/n).

6. ЕТАПИ РОЗРОБКИ

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів
1.	Вивчення літератури за тематикою проєкту	10.10.2021
2.	Розроблення та узгодження технічного завдання	13.10.2021
3.	Аналіз існуючих рішень	18.10.2021
4.	Підготовка матеріалів першого розділу курсового проєкту	10.11.2021
5.	Підготовка матеріалів другого розділу курсового проєкту	19.11.2021
6.	Підготовка дослідницької частини курсового проєкту	25.11.2021
7.	Оформлення документації курсового проєкту	30.11.2021 – 20.12.2021
8.	Попередній огляд матеріалів курсового проєкту	05.12.2021 – 25.12.2021
9.	Захист курсового проєкту	06.12.2021 – 29.12.2021

ДОДАТОК Е. Перший аркуш пояснювальної записки

					ІАЛЦ. 467200.004 ПЗ			
	З	А	№ докум.	Підпис				Д
Виробив		Микитенко С.С.		ата	<i>Маршрутизація в мережі передачі даних.</i> <i>Пояснювальна записка</i>	Піт	Ар	Архівів
Перевірів		Орлова М.М.					куш 1	xx
Н. контроль								
Затвердив								
						КПІ ім. Ігоря Сікорського, ФПМ КВ-8Х		