Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)
Кафедра комп'ютерних систем та мереж
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

Магістр
(назва освітнього ступеня)

на тему: **Методи і засоби підвищення продуктивності передачі даних в комп'ютерних мережах шляхом модифікації протоколів маршрутизації**

Виконав: студент(ка) 6 курсу, групи СІм-62
спеціальності 123 «Комп'ютерна інженерія»
(шифр і назва спеціальності)

(підпис) Озарків Т. А. (прізвище та ініціали)

Керівник (підпис) Жаровський Р.О. (прізвище та ініціали)

Нормоконтроль (підпис) Тиш Є.В. (прізвище та ініціали)

Завідувач кафедри (підпис) Осухівська Г.М. (прізвище та ініціали)

Рецензент (підпис) Мудрик І.Я. (прізвище та ініціали)

Тернопіль
2023

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних систем та мереж
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ *Осухівська Г.М.*
(підпис) (прізвище та ініціали)
« » грудня 2023 р.

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня *магістр*
(назва освітнього ступеня)

за спеціальністю *123 «Комп'ютерна інженерія»*
(шифр і назва спеціальності)

студенту *Озарківу Тарасу Андрійовичу*
(прізвище, ім'я, по батькові)

1. Тема роботи *Методи і засоби підвищення продуктивності передачі даних в комп'ютерних мережах шляхом модифікації протоколів маршрутизації*

Керівник роботи *Жаровський Руслан Олегович, кандидат технічних наук*
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « *01* » *грудня 2023* року № *4/7-1132.*

2. Термін подання студентом завершеної роботи *28.12.2023 р.*

3. Вихідні дані до роботи *Протоколи маршрутизації, структура мережі*

4. Зміст роботи (перелік питань, які потрібно розробити)
Вступ 1.Теоретичний огляд методів маршрутизації даних в комп'ютерних мережах,
2 Аналіз методів оптимізації роботи протоколу EIGRP в мережах під змінними навантаженнями
3 Апробація запропонованих методів підвищення продуктивності передачі даних
4 Охорона праці та безпека в надзвичайних ситуаціях
Висновки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)
1. Актуальність і мета дослідження.
2. Задачі дослідження, об'єкт і предмет, наукова новизна і практична цінність дослідження.
3. Метрики досліджуваних протоколів маршрутизації
4. Балансування навантаження в EIGRP
5. Блок схема модифікації протоколу EIGRP
6. Модель мережі в OMNeT++ IDE
7. Результати експериментального дослідження.
8. Висновки

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	*Осухівська Г. М., зав. кафедри КС*		
Безпека в надзвичайних ситуаціях	*Стадник І. Я., професор кафедри ОХ*		

7. Дата видачі завдання *20.11.2023*

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	*Аналіз сучасних протоколів маршрутизації*	*01.12.2023*	*Виконано*
2.	*Аналіз метрик сучасних протоколів маршрутизації на основі яких здійснюється визначення параметрів і підбір маршруту передачі даних*	*03.12.2023*	*Виконано*
3.	*Аналіз можливих рішень по модернізації протоколу EIGRP для оптимізації його роботи в мережах із динамічним навантаженням*	*10.12.2023*	*Виконано*
4.	*Моделювання і визначення ефективності запропонованого методу*	*16.12.2023*	*Виконано*
5.	*Охорона праці та безпека в надзвичайних ситуаціях*	*18.12.2023*	*Виконано*
6.	*Оформлення пояснювальної записки і графічного матеріалу*	*19.12.2023*	*Виконано*
7.	*Попередній захист кваліфікаційної роботи магістра*	*20.12.2023*	*Виконано*
8.	*Захист кваліфікаційної роботи магістра*	*28.12.2023*	*Виконано*

Студент _____ *Озарків Тарас Андрійович*
(підпис) (прізвище та ініціали)

Керівник роботи _____ *Жаровський Руслан Олегович*
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Методи і засоби підвищення продуктивності передачі даних в комп'ютерних мережах шляхом модифікації протоколів маршрутизації // Кваліфікаційна робота магістра // Озарків Тарас Андрійович // ТНТУ, комп'ютерна інженерія, група СІм-62 // Тернопіль, 2023 // с. – 89, рис. – 28, табл. – 4, бібліогр. – 35.

Ключові слова: маршрутизація, OSPF, EIGRP, пропускна здатність, метрика, завантаження, таблиця маршрутизації, оптимізація.

У кваліфікаційній роботі магістра досліджено методи і засоби оптимізації сучасних протоколів маршрутизації. Дана робота спрямована на дослідження та порівняння двох ключових протоколів маршрутизації мереж – OSPF та EIGRP, а також на розробці та впровадженні модифікацій до протоколу EIGRP для оптимізації роботи в умовах великого обсягу мережевого трафіку. В роботі проведено аналіз переваг та недоліків обраних протоколів, виявлені обмеження, що обмежують їх ефективність в умовах великого навантаження на мережу.

Запропонований метод підвищення продуктивності ґрунтується на вимірі актуального навантаження на інтерфейсах маршрутизаторів та перерахунку маршрутів в разі зниження їх ефективності. Розроблений метод використовує протокол EIGRP та дозволяє оптимізувати роботу мережі в умовах змінного трафіку.

Запропоновані модифікації були імплементовані у бібліотеці ANSAINET для фреймворку OMNeT++. В результаті моделювання продемонстрована працездатність та ефективність розробленого методу, що сприяє підвищенню продуктивності мережі в умовах збільшеного трафіку.

ABSTRACT

Methods and tools for enhancing data transmission performance in computer networks through the modification of routing protocol // Master’s graduation thesis // Ozarkiv Taras Andriiovych // TNTU, computer engineering, group CIm-62 // Ternopil, 2023 // p. – 89, fig. – 28, tab. – 4, bibliography. - 35.

Keywords: routing, OSPF, EIGRP, throughput, metric, load, routing table, optimization.

The master's thesis examines methods and means of optimizing modern routing protocols. This work is aimed at researching and comparing two key network routing protocols - OSPF and EIGRP, as well as at developing and implementing modifications to the EIGRP protocol to optimize work in conditions of a large amount of network traffic. The paper analyzes the advantages and disadvantages of the selected protocols, reveals the limitations that limit their effectiveness in conditions of high network load.

The proposed method of increasing productivity is based on measuring the current load on router interfaces and recalculating routes in case of a decrease in their efficiency. The developed method uses the EIGRP protocol and allows to optimize the network operation in conditions of variable traffic.

The proposed modifications were implemented in the ANSAINET library for the OMNeT++ framework. As a result of the simulation, the workability and efficiency of the developed method were demonstrated, which contributes to the increase of network productivity in conditions of increased traffic.

.

ЗМІСТ

ПЕРЕЛІК ОСНОВНИХ УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ І СКОРОЧЕНЬ

EIGRP (Enhanced Interior Gateway Routing Protocol) - дистанційно-векторний протокол маршрутизації.

IP SLA – Internet Protocol Service Level Agreements.

IS-IS (Intermediate System to Intermediate System) — протокол маршрутизації проміжних систем.

SDN (Software Defined Network) – програмно - конфігурована мережа

OSPF (Open Shortest Path First) – протокол динамічної маршрутизації.

RIP (Routing Information Protocol) — протокол маршрутизації в невеликих комп'ютерних мережах/

АС - автономна система (англ. Autonomous system), набір маршрутизаторів з підтримкою EIGRP, які повинні стати сусідами EIGRP.

ІТКМ - інформаційно-телекомунікаційна мережа.

ВСТУП

Актуальність теми. Методи і засоби підвищення продуктивності передачі даних в комп’ютерних мережах є актуальною задачею у мережах із складною топологією та великою кількістю альтернативних маршрутів. Використання протоколів маршрутизації автоматизує побудову таблиць маршрутизації, а також дозволяє відшукати нові маршрути при змінах мережі: відмови або появу нових ліній зв'язку та маршрутизаторів.

Протоколи маршрутизації забезпечують пошук та фіксацію маршрутів передачі даних через мережу TCP/IP. Хоча більшість протоколів маршрутизації в своїй роботі використовують таблиці маршрутизації, однак існують способи передачі пакетів даних в комп’ютерних мережах, які не вимагають формування таблиць маршрутизації.

Поширеним дистанційно-векторним протоколом є удосконалений протокол динамічної маршрутизації EIGRP. Для визначення маршруту в EIGRP беруться значення мінімальної пропускної здатності і значення затримок на маршруті. Це можливо змінити, додавши в формулу метрики, наприклад, надійність і завантаження каналу, але вони фіксуються тільки в момент зміни топології мережі, тобто цей протокол маршрутизації не перебудовується під змінні навантаження, що виникають в комп'ютерній мережі, коли швидкість передачі даних сильно падає.

Це реалізовано з метою зниження рівня службового трафіку в мережі та збереження сумісності векторних метрик IGRP та EIGRP, щоб була можливість одночасного використання цих протоколів і (або) легкої міграції з IGRP до EIGRP. У IGRP така сама складова метрика і дуже схожа формула її обчислення, цей протокол періодично анонсує маршрутну інформацію, тобто навантаження і надійність інтерфейсу регулярно поширюються по всій мережі, тому ці параметри включені до розрахунку загальної метрики, що може призвести до нестабільності роботи мережі через велику кількість службового трафіку, що викликається постійними змінами даних показників [3].

Методи для отримання інформації про стан зв'язків дозволяють маршрутизаторам будувати граф мережі. Усі маршрутизатори оперують на основі цього спільного графа. Це зроблено щоб процедура маршрутизації була стійкою до конфігураційних змін. Крім того інформацію про граф мережі маршрутизатор використовує при побудові оптимального маршруту проходження пакетів даних у відповідності до певного критерію.

В магістральних IP-мережах на сьогоднішній день широко використовується протокол OSPF для маршрутизації. У ньому використовується алгоритм Дейкстри для розподілу вхідних потоків. Проте цей алгоритм не завжди ефективний у мережах з високим рівнем динамічно змінюваного трафіку, і перевантаження може призводити до відмови важливих вузлів мережі з точки зору передачі даних.

Протоколи OSPF та EIGRP є важливими протоколами, і їх базові характеристики описані в документах RFC 2328 [5; 6] та RFC 7868 [4] відповідно. У протоколі EIGRP вбудовано механізм обліку навантаження на лінію, але цей інструмент не завжди гнучко адаптується. Хоча EIGRP можна налаштувати для використання метрики із різними складовими, що можуть змінюватися, перерахунок маршрутів відбувається лише при зміні топології мережі, і динамічно змінюване навантаження на канал не враховується.

Тому необхідним є розробити метод, з використанням сучасних протоколів маршрутизації, який буде враховувати поточне навантаження на інтерфейсах маршрутизаторів і перераховувати маршрут при зниженні їх продуктивності.

Мета кваліфікаційної роботи. Метою даної роботи є підвищення продуктивності передачі даних в комп’ютерних мережах шляхом модифікації протоколів маршрутизації і моделювання отриманих результатів для визначення ефективності роботи запропонованих методів.

Для досягнення зазначеної мети були поставлені наступні завдання:

- розглянути найбільш використовувані протоколи маршрутизації мереж, OSPF та EIGRP;

- виявити переваги і явні недоліки OSPF і EIGRP, що не дозволяють ефективно використовувати ці протоколи в мережах з високим рівнем трафіку;

- запропонувати модифікацію протоколу EIGRP для перебудови таблиць маршрутизації з урахуванням динамічно змінюваного навантаження на канали;

- реалізувати запропоновані зміни, показати їх ефективність та оцінити їх параметри шляхом моделювання мережі в фреймворку ANSAINET для OMNet++.

Об'єкт дослідження: протоколи маршрутизації в комп'ютерних мережах.

Предмет дослідження: методи і алгоритми роботи протоколів маршрутизації комп'ютерних мереж.

Методи дослідження: використовувати поняття та методи теорії графів, теорії алгоритмів, теорії масового обслуговування, теорії мов програмування, теорії систем і мереж. На етапі аналізу повинен бути визначено критерій ефективності роботи мережі. При реалізації деталей модифікації об'єктно-орієнтований підхід є основним.

Наукова новизна дослідження полягає в розробці методу підвищення продуктивності мережі при змінних навантаженнях на канали зв'язку, що заснований на вимірюванні поточного рівня завантаження на інтерфейсах маршрутизаторів і перерахунку маршрутів при падінні продуктивності на них, де в якості протоколу маршрутизації використовується протокол EIGRP.

Практичне значення результатів кваліфікаційної роботи у розробці нового методу реакції протоколів маршрутизації на зміну завантаження каналів зв'язку. Отримані результати можуть бути використані для подальшого дослідження факторів, що впливають на якість роботи комп'ютерних мереж.

Публікації. Результати дослідження апробовано на XI науково-технічній конференції Тернопільського національного технічного університету імені Івана Пулюя «Інформаційні моделі, системи та технології», XII міжнародній науково-технічній конференція молодих учених та студентів «Актуальні задачі сучасних технологій», у вигляді тез конференцій.

Озарків Т., Жаровський Р. Метод оптимізації EIGRP протоколу для підвищення продуктивності передачі даних в комп'ютерних мережах. Матеріали XI науково-технічної конференції Тернопільського національного технічного

університету імені Івана Пулюя «Інформаційні моделі, системи та технології» (13-14 грудня 2023 року). Тернопіль: ТНТУ. 2023. С.167

Озарків Т., Жаровський Р. Оптимізація роботи протоколу EIGRP в умовах великих мереж зі складною топологією. Матеріали XII Міжнародна науково-технічна конференція молодих учених та студентів «Актуальні задачі сучасних технологій» (6-7 грудня 2023 року). Тернопіль: ТНТУ. 2023. С. 442.

Структура роботи. До складу кваліфікаційної роботи магістра входить розрахунково-пояснювальна записка та графічний матеріал. Розрахунково-пояснювальна записка містить вступ, 4 розділи, загальні висновки, список використаної літератури і додатки. Обсяг роботи: розрахунково-пояснювальної записки – 89 арк. формату А4, графічна частина – 8 аркушів формату А1.

РОЗДІЛ 1
ТЕОРЕТИЧНИЙ ОГЛЯД МЕТОДІВ МАРШРУТИЗАЦІЇ ДАНИХ В КОМП’ЮТЕРНИХ МЕРЕЖАХ

Балансування трафіку є важливою функцією в програмному забезпеченні маршрутизації та є стандартною на всіх платформах маршрутизаторів. Ця функція автоматично активується, якщо в таблиці маршрутизації є кілька шляхів до адресата. Вона розподіляє трафік за допомогою протоколів, таких як EIGRP, OSPF або за допомогою статично налаштованих маршрутів.

В даному розділі проведемо огляд використовуваних протоколів маршрутизації, виявимо їх переваги і недоліки. А також проведемо аналіз літератури пов’язаною з темою кваліфікаційної роботи.

1.1. Огляд літератури по тематиці кваліфікаційної роботи

Протоколи маршрутизації EIGRP і OSPF не адаптовані до різких збільшень навантаження на канали на тривалий час, що також буде продемонстровано моделюванням.

Вирішення (хоча б часткове) цієї проблеми має теоретичне та практичне значення для більш ефективного використання ресурсів мережі, це дозволить зменшити навантаження на окремі вузли мережі та збільшити їх відмовостійкість, наприклад за рахунок зміни маршруту, яким передається інформація, на альтернативний, менш завантажений, і в той же час ж час дасть додаткові можливості для подальшого аналізу і покращення роботи протоколів.

Багато дослідників пропонують свої способи для пом'якшення наслідків виникнення перевантажень в мережах. У ході аналізу наявних розробок у цій галузі було виявлено, що для EIGRP дослідження ведуться не так активно, як, наприклад, для OSPF. Це може бути пов'язано з тим, що протокол маршрутизації EIGRP є пропрієтарною розробкою фірми Cisco і був відкритий на початку 2013 року як інформаційний RFC, а не як стандарт [4], це дозволяє Cisco Systems, Inc. зберігати

контроль над протоколом EIGRP. При цьому компанія Cisco залишила закритими деталі функціонування і (або) реалізації деяких функцій і особливостей EIGRP для збереження і захисту досвіду клієнтів і їх інвестицій в розгортання мереж [11; 12]. Цей факт може уповільнювати (обмежувати) впровадження та використання EIGRP в маршрутизатори інших виробників мережевого обладнання. Аналіз деяких з існуючих рішень як для EIGRP, так і для OSPF наведено нижче.

У [13] представлено Рішення завдання пошуку найкоротших шляхів алгоритмом Беллмана-Форда, який покладено основою протоколів маршрутизації RIP, BGP, EBGP, IBGP, IGRP, EIGRP. Модифікований алгоритм Беллмана-Форда для побудови резервних шляхів використовує дані про ребра, що входять до вузлів. Його використання в складі протоколів маршрутизації дозволяє підвищити стійкість телекомунікаційної системи на величини, пропорційні топологічній складності та зв'язності її мережі. Наприклад, приріст стійкості мережі за показником середньомережева ймовірність стійкості інформаційного спрямування зв'язку становив 11 %. Це досягається завчасним формуванням мережі резервних шляхів і швидким переходом на резервні канали без витрат часу на пошук нових шляхів.

Є роботи, спрямовані на підвищення інформаційної безпеки під час передачі даних у EIGRP-мережах. Використання параметра ризику інформаційної безпеки у формулі розрахунку метрики протоколу EIGRP для маршрутизації трафіку по самим безпечним маршрутам в мережі пропонується в [14]. Тут ризик розраховується на основі двох складових: ризик на основі стандарту NIST CVSS і ризик, обчислюваний на основі формули рівня вразливості вузла з теорії живучості інформаційних систем. Це дозволяє розглядати як інформаційну безпека маршрутизуються пакетів, так та структурну цілісність мережі. Також пропонується модифікований алгоритм-ритм розподілу навантаження між маршрутами, що дозволяє розвантажити найбільш ефективний вузол маршрутизації, коли на мережа Виготовляється DoS-атака.

Результати дослідження [14] показують, що пропонований підхід може бути використаний для збільшення ймовірності запобігання порушення інформаційної безпеки пакетів, що маршрутизуються, і для забезпечення безпеки найбільш

ефективних вузлів маршрутизації в мережі, які дають змогу ефективно маршрутизувати довірений трафік, коли мережа знаходиться під DoS-атакою, або в ній відсутні критичні системні ресурси.

Авторами статті [15] було запропоновано підвищити продуктивність традиційного EIGRP шляхом додавання деяких SDN-функцій (англ. Software Defined Networking, програмно-визначувана або програмно-конфігурована мережа) в вигляді модифікованого підходу для покращення деяких показників продуктивності мережі (перевантаження каналів та рівень втрат пакетів), що наводить до збільшення (оптимізації) загальної пропускної здатності мережі. Цей підхід призводить до створення мережі, де є інтелектуальний динамічний контролер-коректор (англ. dynamic supervisory corrector controller), здатний виявляти точку перевантаження до або на початку її появи. Контролер обробляє вибрані потоки даних на вибраних маршрутизаторах по всій мережі, щоб запобігти перевантаженості за допомогою інтелектуального евристичного алгоритму запобігання перевантаженню та алгоритму маршрутизації, тим самим максимально зменшуючи генерацію керуючих повідомлень.

У [16] розглянуто питання ефективного розподілу вхідних інформаційних потоків у магістральних IP-мережах при використанні протоколу OSPF з метою покращення роботи алгоритму Дейкстри при перевантаженнях в інформаційних мережах.

Численні випробування нового алгоритму Дейкстри спільно з алгоритмом робастної корекції здійснювалися методом Монте-Карло. Моделювання проводилося із гранично високим мережним навантаженням.

В результаті було запропоновано алгоритм, в основу якого закладено алгоритм Дейкстри, суттєво понижуючий ймовірність перевантажень. Це досягається за рахунок використання в критерії розподілу замість пропускної здатності каналу його залишкової пропускної здатності. Суть запропонованого узагальненого алгоритму Дейкстри ось у чому. При розподілі чергової частки вхідного потоку $Q\ r(t)$ (матриця $Q(t)$ визначає багатоадресний потік даних всіх маршрутизаторів мережі) в критерій якості включається інформація про завантаженість каналу зв'язку

раніше розподіленими частинами вхідного потоку. Тому поточний розподіл частини потоку враховуватиме реальну пропускну здатність IP-мережі. Моделювання нового алгоритму показало суттєве покращення мінімаксного критерію якості (метрики) зі зростанням розбиття вхідного потоку інформації.

Експерименти по порівнянні якості функціонування запропонованого в роботі [16] алгоритму спільно з алгоритмом робастної корекції і розподілу навантаження в IP-мережах за допомогою методу лінійного програмування дозволяють зробити висновок про високий рівень ефективності модифікованого алгоритму. До недоліку слід віднести відсутність достатньої гарантії від мережевих навантажень. Для розріджених IP-мереж із числом вузлів 27 і більше, високу ефективність нового алгоритму порівняно з алгоритмом лінійного програмування показати не вдалося.

У [17] авторами статті було поставлено питання можливості підвищення стійкості зв'язку при маршрутизації (стійкість відноситься до одної з основних властивостей мереж зв'язку) шляхом модифікації алгоритму Дейкстри, що дозволяє одночасно з рішенням завдання пошуку найкоротших шляхів сформувати резервні шляхи до вузлів мережі. Підхід цієї роботи було взято за основу при модифікації алгоритму Беллмана-Форда [13].

Методи: на користь використання топологічної надмірності мережі зв'язку модифікується алгоритм Дейкстри в напрямку розширення його функціональності за рахунок формування як найкоротших так і резервних шляхів. Це розширення забезпечується введенням додаткових множин в розрахунок, а також нових блоків у тіло алгоритму.

У результаті була проведено модифікація алгоритму Дейкстри, заснована на використанні вхідних в вузли ребер для побудови резервних шляхів до вузлів. Оцінку приросту стійкості мережі зв'язку здійснено за показником ймовірності стійкості інформаційного спрямування. Розглянуто роботу алгоритму на прикладі мережі, і показано, що його застосування дає підвищення стійкості від 5 до 35 % за обґрунтованим показником. Підвищення стійкості інформаційно-телекомунікаційних мереж (ІТКМ) по показнику ймовірність стійкості інформаційного спрямування відбувається за рахунок одночасного підвищення

показників коефіцієнту готовності інформаційного спрямування (завдяки зниженню часових затримок відновлення інформаційного спрямування зв'язку за рахунок перемикання інформаційних потоків) і ймовірність виживання інформаційного потоку в результаті деструктивно-руйнівних впливів (досягається введенням резервних шляхів). Приріст ефективності, отримуваний при використанні модифікованого алгоритму Дейкстри, зростає зі збільшенням топологічної складності мережі і при зниженні тривалості часових параметрів конкретних протоколів маршрутизації.

Запропонована в [17] модифікація алгоритму Дейкстри може бути використана для вдосконалення (поліпшення ефективності) протоколів маршрутизації PNNI і OSPF в цілях забезпечення заданого рівня стійкості мереж ATM і TCP/IP в умовах деструктивно-руйнівних впливів на мережеві елементи.

У [18] розглядається питання гарантованого розподілу навантаження на мережі під керуванням протоколу OSPF за допомогою застосування планованої адаптивної стратегії маршрутизації, яка спиралася б на оцінку пропускної здатності та стану каналів. Наведено результати досліджень 3-х алгоритмів адаптивного балансування трафіку в мережі, що функціонує за протоколом маршрутизації OSPF, та результати виявлення їхніх недоліків та переваг при моделюванні.

Кожен із описаних алгоритмів має свої особливості в описі та розробці та був протестований певним чином. Алгоритм адаптивного балансування через оцінку ефективної пропускної здатності (алгоритм № 1) моделювався серед OPNET Modeler. Алгоритм адаптивного та розподіленого балансування в мережі OSPF (алгоритм № 2) був випробуваний на випадкових мережах, що генеруються механізмом, описаним в одному з використаних у роботі джерел, за різних сценаріїв трафіку.

Модифікований алгоритм адаптивного балансування в мережі OSPF на основі нечіткої логіки (алгоритм № 3) тестувався в симуляторі ns-2 на різних рівнях навантаження в мережі. Для алгоритму № 1 якість мережі оцінювалась експериментально шляхом моделювання з точки зору середньої пропускної здатності мережі, затримки передачі пакетів і швидкості втрати пакетів. З метою

порівняння традиційних методів маршрутизації OSPF було прийнятий за основу. Було підтверджено, що адаптивна стратегія маршрутизації приймає кращі рішення, ніж при еталонному OSPF.

Цей результат був інтуїтивно очікуваним в зв'язку з тим, що адаптивний алгоритм маршрутизації може краще розподілити навантаження трафіку серед мережевих з'єднань. Безпосереднім наслідком цього факту є висока пропускна здатність мережі, збережена навіть при високій інтенсивності трафіку.

Головним недоліком цього алгоритму є велика затримка за рахунок додатковою навантаження повідомленнями про оновлення вартості каналу. Але додаткова затримка, сформована для повідомлень, має незначний вплив по порівнянні з введеними альтернативними маршрутами. За темпами втрат пакетів використання адаптивної стратегії маршрутизації стає ще кориснішим, коли в мережі переважають потоки з високою швидкістю.

Якщо порівнювати алгоритм №1 без модифікацій з алгоритмами № 2 і № 3, то попри на свою більш просту математичну модель та неефективність при малих навантаженнях на мережу, алгоритм охоплює більше параметрів якості обслуговування та гарантує більше використання мережевих ресурсів (за порівнянні з алгоритмом № 2). У алгоритмі № 2 визначальною особливістю є використання функцій задачі оптимізації (неоптимальної або субоптимальної) та функцій статичної задачі балансування навантаження (оптимальної). Використання цих завдань дозволяє сформувати критерії та більш гнучко розрахувати параметри для зміни вагових коефіцієнтів каналу. У алгоритмі № 3 використовуються функції приналежності, що дозволяє отримати найоптимальніший результат, заснований на попередньому складанні та результатах. У роботі пропонується модифікація, метою якої є зменшення кількості і частоти змін вартості каналів, що дозволяє уникнути різких коливань трафіку, але модифікація не призвела до суттєвих покращень показників якості, проте суттєво зменшила кількість змін коефіцієнтів вартості каналів. Також вдалося зменшити кількість втрат пакетів на 20%.

Завдання OSPF-маршрутизації відноситься до NP-складних завдань. Запропоновані в [18] алгоритми показують можливість створення таких алгоритмів,

які дозволяли б адаптивно та динамічно керувати трафіком у мережі та досягти стабільності та конвергентності в найкоротший проміжок часу, високого рівня використання мережевих ресурсів зі зниженням навантаження на окремі канали та зменшенням рівня втрати пакетів.

У [19] порушується питання підвищення ефективності використання протоколу OSPF в корпоративних мережах в умовах динамічних навантажень на лініях зв'язку за рахунок зменшення трудомісткості розрахунку таблиць маршрутизації.

Для підтвердження правильності пропонованого алгоритму на базі протоколу OSPF була розроблена програма імітаційного моделювання процесів маршрутизації в корпоративних мережах. При розробці основна увага приділялася коректності запропонованого алгоритму і розмірності завдання, що вирішується (виражається через кількість вершин, для яких необхідний пошук найкоротшого шляху; корпоративна мережа тут представляється в вигляді неорієнтованого зваженого зв’язаного графа).

Для підвищення ефективності використання протоколу OSPF на його базі було запропоновано алгоритм парних перестановок маршрутів у корпоративних мережах, що дозволяє за рахунок збору додаткової інформації врахувати можливі зміни конфігурації корпоративної мережі та не проводити повний перерахунок маршрутних таблиць, а також зменшити трудомісткість побудови (розрахунку) таблиць маршрутизації з величини $O(N\ 2\ )$ (при виборі оптимального маршруту по алгоритму Дейкстри), де N - число маршрутизаторів в корпоративної мережі, до величини $O(N)$.

Під час імітаційного моделювання на різних розмірностях завдання (графи, що складаються з 10, 100 і 500 вершин) було показано, що математичне сподівання числа змін не перевищує величини $N/2$, а його максимальне значення не перевищує N. На основі цього був зроблено висновок, що запропонований алгоритм адаптивної прискореної маршрутизації на базі протоколу OSPF є ефективним при пошуку оптимальних маршрутів в умовах часткових динамічних змін на лініях зв'язку

корпоративної мережі за рахунок використання додаткової інформації про можливі заміни в маршрутах.

Таким чином, розроблений алгоритм парних перестановок маршрутів на базі протоколу OSPF дозволяє зменшити трудомісткість побудови таблиць маршрутизації до величини порядку $O(N)$, через що підвищується ефективність функціонування корпоративних мереж в умовах динамічних навантажень на лініях зв'язку в цих мережах.

У [20] автор поставив питання підвищення енергоефективності мережевих пристроїв при маршрутизації. Зокрема, розглядається нова ідея енергоефективного розширення протоколу OSPF.

Для експериментів використовується середовище моделювання комп'ютерних мереж OMNeT++.

У результаті запропоновано нову концепцію протоколу OSPF, засновану на алгоритмі Дейкстри, який перераховує найкоротший шлях щоразу, коли оновлюється метрика. У OSPF вводиться нове значення метрики протоколу, що залежить від інтенсивності трафіку на інтерфейсах маршрутизаторів. Це розширення протоколу реалізовано OMNeT++. Були проведені тести для покращення OSPF-симулятора в середовищі OMNeT++, надалі він буде забезпечувати: інтерфейси маршрутизаторів, що не використовуються, можуть бути відключені або переключені на різні енергетичні стану (у залежно від навантаження на них), завдяки цьому енергія може бути збережена, а комп'ютерна мережа зможе працювати більш ефективно.

Робота по розширенню модуля OSPF в середовищі OMNeT++ ще продовжується. Ціль запропонованого рішення складається в контролі енергетичних станів інтерфейсів маршрутизатора. Енергоефективність (для даної роботи) розуміється як підтримка процесу балансування навантаження для інтерфейсів маршрутизатора. З допомогою нової функціональності маршрутизатори зможуть обмінюватись інформацією про споживання енергії на інтерфейсах, що дозволить змінювати їх стан і в кінцевому результаті зберегти енергію і скоротити витрати в великих комп'ютерних мереж зв'язку.

У той час як працездатність та ефективність розглянутих алгоритмів підтверджена, деталі їх реалізації недоступні, методи [13; 14; 17] ніде не реалізовані. Подані в [16; 17; 18; 19] алгоритми впроваджено для використання в відмінних від OMNeT++ середовищах моделювання. Робота [20], реалізована для OMNeT++, усе ще не закінчена, акцент тут зроблено на підвищенні енергоефективності мережі, а не на покращенні роботи OSPF при виникненні пікових навантажень. У [14] для оптимізації використовуються параметри, що враховують ризики інформаційної безпеки. Автори роботи [15] вказують, що при дуже високих рівнях трафіку, що застосовується всередині SDN-контролера алгоритм не дозволяє повністю захистити мережі від втрат пакетів, тут також вводиться додатковий керуючий вузол, який не завжди реально знайти. У [16] відсутня достатня гарантія від мережевих перевантажень, для мереж з великою кількістю вузлів не вдалося показати високу ефективність нового алгоритму.

Одна з запропонованих у [18] модифікація не призвела до значних поліпшень показників якості, а розроблений у [19] алгоритм призначений для функціонування в корпоративних мережах, які зазвичай є відносно невеликими за розміром. Рішення, запропоновані в [16; 17] актуальні для протоколів маршрутизації на основі алгоритму Дейкстри, а в [18; 19; 20] - саме для OSPF.

1.2. Протокол динамічної маршрутизації OSPF

Протокол OSPF використовується для маршрутизації трафіку в TCP/IP мережах. Інформація про маршрути в OSPF поширюється між маршрутизаторами однієї автономної системи. Протокол OSPF працює на основі технології SPF на відміну від алгоритмів Беллмана-Форда, які використовуються традиційними протоколами маршрутизації TCP/IP [6].

Протокол OSPF підготовлений однойменною робочою групою IETF і призначений для використання в середовищах TCP/IP. Протокол включає явну підтримку безкласової адресації і установки міток (англ. tagging) при використанні зовнішньої маршрутної інформації. OSPF використовує автентифікацію та групову

адресацію (англ. IP multicast) під час обміну маршрутними сполученнями. Крім того, при розробці протоколу були додані значні зусилля щодо прискорення обробки топологічних змін у мережі та зниження рівня службового трафіку.

OSPF забезпечує маршрутизацію пакетів IP виключно на основі IP-адрес одержувачів, визначених з заголовка пакетів IP. Пакети IP маршрутизуються без їх зміни, тобто не використовується інкапсуляція в якісь інші пакети. OSPF є динамічним протоколом маршрутизації, що забезпечує швидке виявлення топологічних змін в AS (наприклад, збої маршрутизаторів або каналів) і розрахунок нових безпетльових маршрутів. Час розрахунку нового маршруту невеликий і при цьому передається невеликий об’єм службового трафіку [6]. Кожен маршрутизатор підтримує базу даних із описом топології AS, ці бази містять інформацію про стан каналів зв’язку. Бази даних усіх маршрутизаторів однієї області ідентичні. Кожен елемент бази даних містить інформацію про певний маршрутизатор (наприклад інтерфейси які підтримуються, доступні сусіди). Маршрутизатори розповсюджують інформацію про свій локальний стан шляхом лавинної маршрутизації.

Усі маршрутизатори в мережі працюють паралельно, використовуючи ідентичний алгоритм. Кожен маршрутизатор, базуючись на інформації про канали, будує дерево найкоротших шляхів, де корінь відповідає самому маршрутизатору. Це дерево включає маршрути до всіх адресатів всередині внутрішньої системи (AS), а маршрутна інформація зовнішнього походження представлена як листя дерева.

Якщо існують кілька шляхів із однаковою вартістю до одного адресата, трафік рівномірно розподіляється між усіма цими маршрутами (пакети для кожного з маршрутів відправляються по черзі). Вартість маршруту визначається безрозмірною метрикою, яка представлена у вигляді одного числа. За замовчуванням ця метрика враховує пропускну здатність каналів зв'язку.

Наприклад, для Ethernet пропускна здатність дорівнює 10, для Fast Ethernet - 1, для каналу T-11 - 65, і для каналу із пропускною здатністю 56 Кбіт/с – 1785. При використанні високошвидкісних каналів, таких як Gigabit Ethernet або STM-16/64, адміністратор повинен адаптувати шкалу швидкостей, встановивши одиничну відстань для найшвидшого каналу [1]. Метрики, які враховують затримки і

надійність передачі пакетів даних каналами зв'язку, також можуть бути використані. Для кожної метрики протокол OSPF будує окрему таблицю маршрутизації. Вибір конкретної таблиці залежить від значень бітів TOS в заголовку IP-пакету, які надходять (Type of Service, призначений для пріорітизації IP-трафіку на мережевих маршрутизаторах з метою забезпечення високої якості передачі даних).

Ідея поділу мережі OSPF на зони полягає в спрощенні адміністрування та оптимізації доступних ресурсів. Оптимізація ресурсів особливо важлива для великих корпоративних мереж із великою кількістю мереж і каналів. Якщо багато маршрутизаторів обмінюються базою даних про стан зв’язку, це може переповнити мережу та знизити її ефективність – це була необхідність, яка призвела до створення концепції зон.

Області (зони) — це логічний набір маршрутизаторів (рис. 1.1.), які мають однаковий ідентифікатор зони або номер всередині мережі OSPF. Сама мережа OSPF може містити кілька областей, перша й головна область називається магістральною областю «Область 0», усі інші області мають підключатися до зони 0.

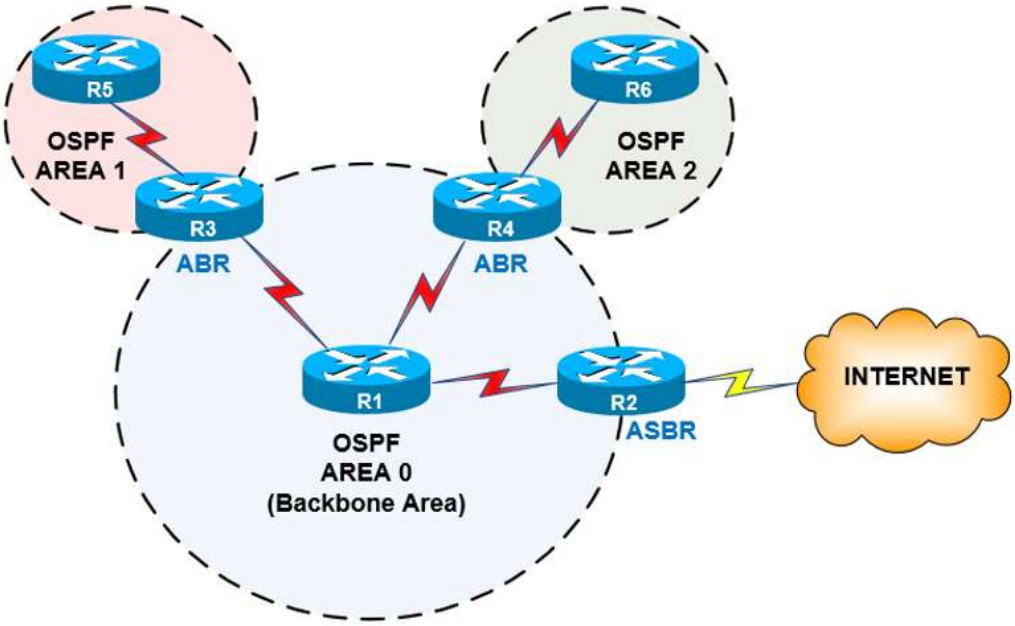

Рис.1.1. Області OSPF

OSPF дозволяє гнучке налаштування IP підмереж. Кожен маршрут у OSPF поширюється із вказанням адреси та маски підмережі. Дві різні підмережі однієї IP

мережі можуть мати різні розміри (тобто різні маски) – для позначення цього зазвичай використовують термін variable length subnetting (змінний розмір підмереж). Пакети маршрутизуються шляхом з найкращою відповідністю. Маршрути до хостів розглядаються як шляхи в підмережі з маскою з одних одиниць (0xffffffff або 255.255.255.255).

Весь обмін інформацією в рамках OSPF здійснюється з використанням автентифікації, що забезпечує участь лише уповноважених маршрутизаторів в маршрутизації всередині AS. Різні схеми автентифікації можуть використовуватись для різних IP-мереж.

Зовнішні дані маршрутизації, такі як маршрути від зовнішніх шлюзів EGP (Exterior Gateway Protocol), наприклад BGP, анонсуються через AS. Ці дані зберігаються окремо від інформації OSPF про стан каналів. Кожен зовнішній маршрут може бути позначений маршрутизатором, який його анонсує і надає можливість обміну додатковою інформацією між маршрутизаторами на кордоні AS [6].

Протокол OSPF використовує ідентифікатор 89 для інкапсуляції в IP і не вимагає додаткової фрагментації або складання пакетів; при необхідності використовується звичайна фрагментація та складання IP. Формат пакетів OSPF дозволяє легко розділяти великі блоки протокольної інформації на менші пакунки. Рекомендується уникати фрагментації IP.

Усі пакети OSPF передаються з нульовим значенням поля IP TOS. Пакети маршрутизації повинні мати перевагу над звичайним IP-трафіком як при прийомі, так і при передачі. Всі пакети OSPF використовують однотипні заголовки.

Типи пакетів OSPF включають Hello для організації сусідських відносин, Database Description та Link State Request для підтримки відносин суміжності, а також Link State Update та Link State Acknowledgment для гарантованого обміну Оновленнями OSPF [6].

Таблиця 1.1

Типи пакетів OSPF

Тип	Назва	Призначення
1	Hello	Виявлення і підтримка сусідства
2	Database Description	Опис вмісту БД
3	Link State Request	Завантаження БД
4	Link State Update	Оновлення БД
5	Link State Acknowledgment	Підтвердження лавинної розсилки

Кожен пакет Link State Update містить набір нових анонсів стану каналів LSA один інтервал (англ. hop) віддалених від пункту генерації анонсу (рис 1.2.).

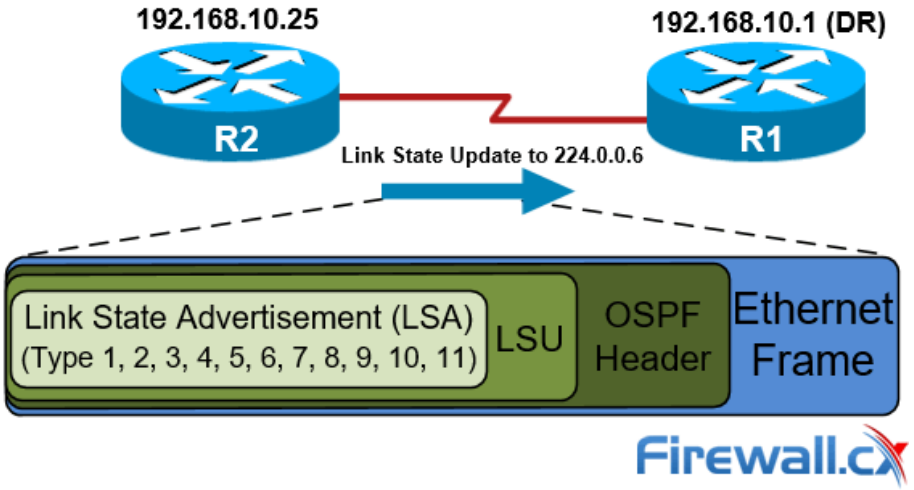

Рис. 1.2. Пакет OSPF LSU, що містить оголошення про стан зв’язку (LSA)

Один пакет Link State Update може отримувати новини LSA від кількох маршрутизаторів. Кожен запис LSA позначається ідентифікатором маршрутизатора, що створив анонс, і супроводжується контрольною сумою вмісту. У кожному записі LSA є поле типу; можливі варіанти цього поля описані в табл. 1.2.

Пакети протоколу OSPF, за винятком Hello, передаються лише між суміжніми маршрутизаторами. Таким чином, всі пакети OSPF пролягають через один інтервал між маршрутизаторами, за винятком випадків використання віртуальних з'єднань. IP-адреса відправника пакету OSPF визначається як адреса одного суміжного маршрутизатора, а IP-адреса одержувача — як адреса іншого суміжного маршрутизатора або групова IP-адреса [6].

Таблиця 1.2

Типи анонсів LSA в OSPF

Тип	Ім'я LSA	Опис LSA
1	Router - LSA	Генеруються усіма маршрутизаторами. Цей тип LSA описує стан інтерфейсів маршрутизатора в області. Анонс розсилається у лавинному режимі всередині області.
2	Network-LSA	Генерується виділеним маршрутизатором DR для широкомовних та NBMA-мереж. Цей тип LSA включає список маршрутизаторів, підключених до мережі. Розсилається в лавинному режимі всередині області.
3, 4	Summary - LSA	Генерується граничними маршрутизаторами областей і розсилається у лавинному режимі в межах пов'язаної з LSA області. Кожен анонс summary-LSA описує маршрут до адресата поза даною областю, але всередині даної AS (міждоменний маршрут). Тип 3 summary-LSA описує маршрути у мережі, а тип 4 – дограничним маршрутизаторам AS.
5	AS-external-LSA	Генерується граничними маршрутизаторами AS та розсилається по всій автономній системі. Кожен анонс AS-external-LSA описує маршрут до адресатам в інший AS. Прийняті по замовчуванням маршрутизатори AS також можуть описуватися в AS-external- LSA.

Переваги OSPF:

- швидка збірка маршрутів: OSPF може швидко адаптуватися до змін у мережі, перебудовуючи маршрути при виникненні змін у топології;

- підтримка багатозадачності: OSPF дозволяє передавати дані з різних джерел і використовується для розподілу трафіку в мережі.

- широка різноманітність метрик: OSPF дозволяє використовувати різні метрики для визначення оптимальних маршрутів, такі як пропускна здатність, затримка, надійність тощо.

- підтримка VLSM (Variable Length Subnet Masking): OSPF підтримує VLSM, що дозволяє ефективно використовувати IP-адреса, зменшуючи втрату адресного простору.

- розширена безпека: OSPF має механізми аутентифікації, які дозволяють забезпечити безпеку обміну маршрутною інформацією.

Недоліки OSPF:

- споживання ресурсів: OSPF може вимагати значних обчислювальних ресурсів, особливо великих мережах, що може вплинути на продуктивність маршрутизаторів.

- складність конфігурації: Налаштування OSPF може бути складним завданням, особливо для великих мереж, і вимагає досвіду в адмініструванні мережі.

- витрати на широкомасштабні мережі: Великі OSPF-мережі можуть викликати значні навантаження на мережевий трафік та пам'ять маршрутизаторів.

- обмеженість в виборі маршрутів: Хоча OSPF підтримує різні метрики, але вибір маршрутів може бути обмеженим в порівнянні з іншими протоколами маршрутизації.

- обмежена підтримка для інших протоколів: OSPF спроектовано для використання в середовищі TCP/IP, і йому може бракувати підтримки для інших мережевих протоколів.

1.3. Протокол динамічної маршрутизації EIGRP

EIGRP представляє собою удосконалену версію IGRP, використовуючи ту ж технологію DVA, і взаємодіє з маршрутизаторами IGRP, забезпечуючи повну сумісність. За замовчуванням маршрути IGRP мають вищий пріоритет, аніж маршрути EIGRP, хоча це можна змінити за необхідності. Властивості збіжності та ефективності роботи протоколу значно покращено, що дозволяє модернізувати мережеву архітектуру, економлячи витрати, вкладені у розробку мережі на основі IGRP [7; 8].

Технологія конвергенції в EIGRP ґрунтується на дослідженнях, проведених компанією SRI International. Для отримання безпетельових маршрутів у будь-який момент часу застосовується розподілений оновлюваний алгоритм DUAL, який забезпечує синхронізацію всіх маршрутизаторів, що беруть участь у зміні топології. Маршрутизатори, які не зазнали змін топології, не взаємодіють у цьому процесі. Час

конвергенції за алгоритмом DUAL порівняно з іншими існуючими протоколами маршрутизації [7; 8].

EIGRP, як незалежний від мережевого рівня протокол, забезпечує алгоритму DUAL можливість підтримувати інші групи протоколів. В протокол EIGRP входять наступні основні компоненти [7; 8; 9]:

1)Процес виявлення/відновлення сусіда: Маршрутизатори використовують цей процес для динамічного вивчення інших маршрутизаторів, які є безпосередньо підключені до їхньої мережі. HELLO-пакети регулярно відправляються для виявлення сусідів. Сусідство діє, доки маршрутизатор отримує ці пакети, і призначено для виявлення недоступності сусіда.

2)Протокол RTP гарантує надійність та порядок доставки пакетів EIGRP всім сусідам. Він підтримує групову та одноадресну адресацію. Надійність передачі застосовується тільки в разі необхідності для певних видів пакетів, щоб підвищити ефективність.

3)Блок DUAL реалізує процеси прийняття рішень для розрахунків усіх маршрутів. DUAL використовує дистанційну інформацію (метрику) для вибору ефективних шляхів без петель. Алгоритм визначає наступників для додавання до таблиці маршрутизації, керуючись принципом ймовірних наступників.

4) PDM модулі обробляють вимоги конкретних протоколів мережного рівня. Наприклад, IP-EIGRP відповідає за взаємодію з IP-пакетами, їх розбір, відправлення та отримання пакетів EIGRP. Він взаємодіє з алгоритмом DUAL для прийняття рішень про маршрутизацію та зберігання результатів у таблиці маршрутизації IP.

Протокол EIGRP використовує різні типи пакетів [4; 7; 8; 10]:

1)HELLO/ACK (підтвердження):

–multicast-пакети для виявлення/відновлення сусіда, що не потребують підтвердження отримання;

–пакет HELLO без даних також використовується як підтвердження;

–пакет підтвердження ACK завжди відправляється в режимі одиночної відправки та включає ненульовий номер підтвердження.

2)UPDATE:

–пакети оновлень передають параметри вузлів призначення;

–при виявленні нового сусіда, вони використовуються для створення таблиці топології та відправляються в режимі одиночної відправки;

–у випадках, таких як зміна вартості зв'язку, пакети надсилаються в режимі мультивідправки;

–оновлення завжди передаються із підтвердженням.

3)QUERY і REPLY:

–пакети запитів (QUERY) і відгуків (REPLY) відправляються, коли маршрути призначення переходять у активний стан;

–пакети QUERY відправляються в multicast-режимі, тільки якщо вони не вирушають у відповідь на отриманий запит. У цьому випадку запит надсилається в unicast-режимі назад наступному елементу, що створив початковий запит;

–REPLY завжди відправляються в відповідь на QUERY, щоб повідомити запитувача, що переходити в активний стан не потрібно, оскільки відгукується має ймовірні наступні елементи. Пакети REPLY відправляються в режимі одиночної відправки запитувачу;

–запити та відповіді передаються з підтвердженням.

4)REQUEST-пакети:

–використовуються для отримання конкретної інформації від одного або кількох сусідів;

–можуть передаватися в multicast- або unicast-режимах;

–запити передаються з негарантованою доставкою.

Метрика протоколу EIGRP базується на п'яти компонентах [2; 4; 10]:

–пропускна здатність (BW). Визначається як найменша ширина пропускання між вузлами джерелом та призначенням;

–затримка (DELAY). Представляє собою загальний час затримки на всьому шляху, обчислений на основі інтерфейсів;

–надійність (REL). Визначається як найгірший показник надійності на всьому шляху, базуючись на повідомленнях про стан системи (keepalive);

–завантаження (LOAD). Представляє собою найгірший показник завантаження каналу на всьому шляху;

–MTU (Максимальний Розмір Передачі). Показник максимального розміру передачі на всьому шляху. Включається до оновлень EIGRP, але не використовується для обчислення метрики.

За замовчуванням для розрахунку метрики використовуються BW та DELAY, оскільки інші критерії рекомендується уникати через часті перерахунки маршрутів.

Щоб детальніше розібратися в балансуванні навантаження, спочатку слід розглянути загальну формулу розрахунку метрики протоколу EIGRP (1.1) 25]:

$$M_p = \left[\left(K_1 \cdot B^p_{\min} + \frac{K_2 \cdot B^p_{\min}}{256 - L^p_{\max}} + K_3 \cdot D^p_{sum} \right) \cdot \frac{K_5}{K_4 + R^p_{\min}} \right] \cdot 256, \qquad (1.1)$$

де $B^p_{\min}$ - найменше значення виваженого показника пропускної здатності в маршруті p; $L^p_{\max}$ – найбільше завантаження одного з каналів зв'язку у маршруті p; D^p_{sum} - Сумарна затримка пакетів у маршруті $[мкс]$; $R^p_{\min}$ - Найменша надійність одного з каналів зв'язку в маршруті p; $p \in P_{i,j}$, $P_{i,j}$ – всі можливі маршрути в заданій мережі під час передачі між вузлами i, j, при $i \neq j$. Коефіцієнти K_1, K_2, K_3, K_4, K_5 дозволяють враховувати в метриці вказані вище параметри. За замовчуванням у стандартному алгоритмі, описаному Cisco, дані коефіцієнти мають такі значення: $K_1 = K_3 = 1$ та $K_2 = K_4 = K_5 = 0$.

При $K_2 = K_4 = K_5 = 1$ виникають випадки, коли динамічна зміна параметрів, таких як надійність і завантаженість каналів зв'язку, буде приводити до постійного перерахунку метрики (оскільки ці величини динамічно змінюються в процесі передачі трафіку), що згубно впливатиме на центральний процесор маршрутизатора. Тому Cisco не рекомендує їх використовувати для розрахунку метрики.

Розрахунок зваженої пропускної здатності (1.1) провадиться наступним чином:

$$B_{\min}^{p} = \left\lfloor \frac{10^{7}}{\min\left(B_{i,j}^{l,p}\right)} \right\rfloor \left[\frac{Кбіт}{c} \right], \tag{1.2}$$

де $\min\left(B_{i,j}^{l,p}\right)$– найменша пропускна здатність одного з каналів зв'язку l у маршруті p при передачі інформації між вузлами i – відправника та j – одержувача.

Для розрахунку сумарної затримки маршруту використовується така формула:

$$D_{sum}^{p} = \sum_{i \neq j} D_{i,j}^{p}, \tag{1.3}$$

де $\sum_{i \neq j} D_{i,j}^{p}$ – сума затримок пакетів кожного з каналів зв'язку, що входять у маршрут p під час передачі інформації між i – вузлом відправником та j – вузлом одержувачем.

Таким чином, складова метрика EIGRP дозволяє ретельно (оптимально) налаштовувати роботу мереж передачі даних.

1.4. Обґрунтування критерію ефективності роботи мережі

Для оптимізації та ефективного функціонування мережі необхідно визначити критерії її ефективності, які можна поділити на дві основні групи:

– продуктивність. Показники витрат часу оцінюють затримку, яку вносить мережа при обміні даними. Показники пропускної здатності каналу зв’язку, або інтерфейсу відображають об’єм інформації, переданої мережею за одиницю часу;

– надійність. Здатність до правильної роботи протягом тривалого часу: Визначається, наприклад, стеженням за кількістю відмов.

Ці дві групи критеріїв дозволяють оцінити якість мережі та її здатність виконувати завдання. Показники продуктивності та надійності є взаємно зворотні, тобто знаючи один із них, можна обчислити інший.

Одним із ключових часових показників є час реакції, який визначає інтервал часу між запитом та отриманням відповіді на цей запит. Наприклад, показник RTT, який вимірюється за допомогою утиліти ping, є одним з таких показників.

Аналізуючи RTT та частоту втрати пакетів, можна визначити завантаженість каналів передачі даних та виявити можливі проблеми на проміжних пристроях. RTT також пов'язаний з довжиною черги буфера на інтерфейсах пристрою.

Пропускна здатність мережі - це міра її здатності передавати дані між вузлами протягом одиниці часу. Повний обсяг пропускної здатності мережі визначається матрицею трафіку вузлів мережі, яку можна виміряти за допомогою спеціальних засобів вимірювання.

Для методу підвищення продуктивності мережі під навантаженням вибрано поточну пропускну здатність як критерій ефективності роботи мережі, оскільки цей показник добре характеризує швидкість передачі даних між вузлами. Фіксація значень параметра є найбільш простим і поширеним процесом, і вимірювання проводяться на кожному EIGRP-маршрутизаторі в бітах в секунду без розподілу трафіку.

Висновки до розділу 1.

Як видно, для протоколу маршрутизації OSPF є багато різних способів його оптимізації з погляду розподілу навантаження на мережу, а EIGRP ще недостатньо досліджено. Таким чином, є необхідність у створенні, описі та реалізації методу оптимізації протоколу маршрутизації EIGRP при роботі в мережах з великим або нерівномірним рівнем трафіку. Вносити модифікації було вирішено саме в EIGRP.

РОЗДІЛ 2
АНАЛІЗ МЕТОДІВ ОПТИМІЗАЦІЇ РОБОТИ ПРОТОКОЛУ EIGRP В МЕРЕЖАХ ПІД ЗМІННИМИ НАВАНТАЖЕННЯМИ

2.1. Методи оптимізації роботи протоколу маршрутизації

Виберемо для аналізу топологію мережі, подану на рис. 2.1. На цьому малюнку зазначено значення затримки пакетів між маршрутизаторами та пропускні здатності каналів зв'язку.

Інформація передається з пристрою, що знаходиться в $Cloud\,1$, на пристрій, розташований в $Cloud\,2$. На маршрутизаторі $R1$ увімкнено функцію балансування навантаження. Для маршрутизатора $R1$ існує два шляхи до мережі $Cloud\,2$: перший маршрут $p_1 \in [R1; R2; R4; Cloud\,2]$, другий маршрут $p_2 \in [R1; R3; R4; Cloud\,2]$.

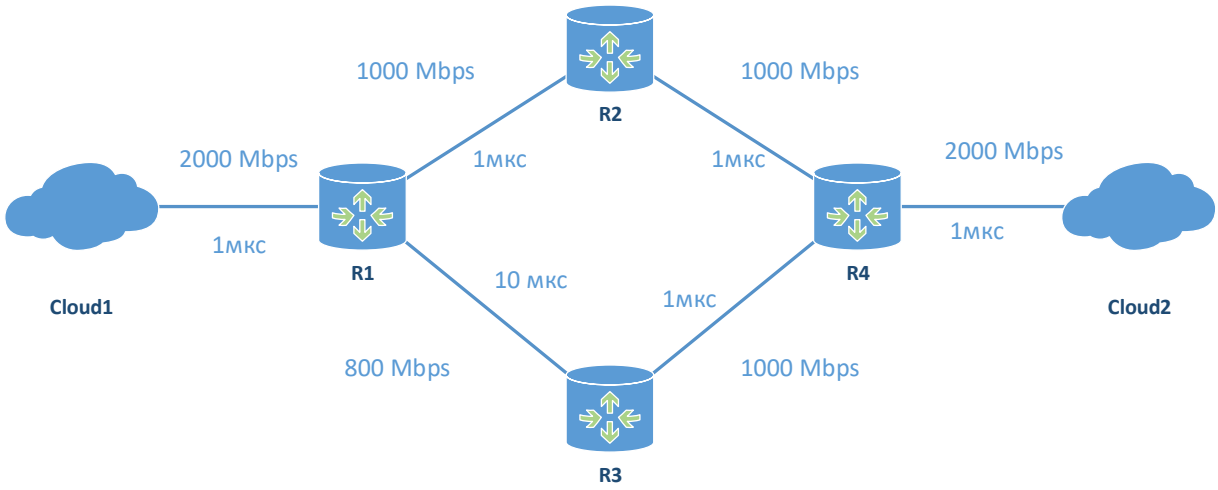

Рис.2.1. Приклад топології мережі з ілюстрацією проблеми при нерівнозначному балансуванні навантаження у динамічному протоколі маршрутизації EIGRP

У стандартній ситуації без використання балансування навантаження маршрутизатор $R1$ вибере маршрут p_1 з меншою метрикою M_{p_1}. Для того, щоб EIGRP зміг здійснювати балансування навантаження необхідно, щоб маршрути, що

залишилися (з метрикою меншою, ніж у кращого маршруту), стали запасними. При цьому використовується M_p - поточна відстань - це метрика певного маршруту від заданого маршрутизатора, до мережі призначення і M_p^{RD} – заявлена відстань - це метрика певного маршруту отримана від сусіднього заданого маршрутизатора, через якого проходить цей маршрут до мережі призначення.

Маршрут p_2 з метрикою M_{p_2} може бути обраний як запасний при дотриманні певних умов (рис. 2.2.). Щодо топології (рис. 2.1), такими умовами є те, що заявлена відстань $M_{p_3}^{RD}$ маршруту p_3 має бути менше, ніж поточна відстань маршруту p_1.

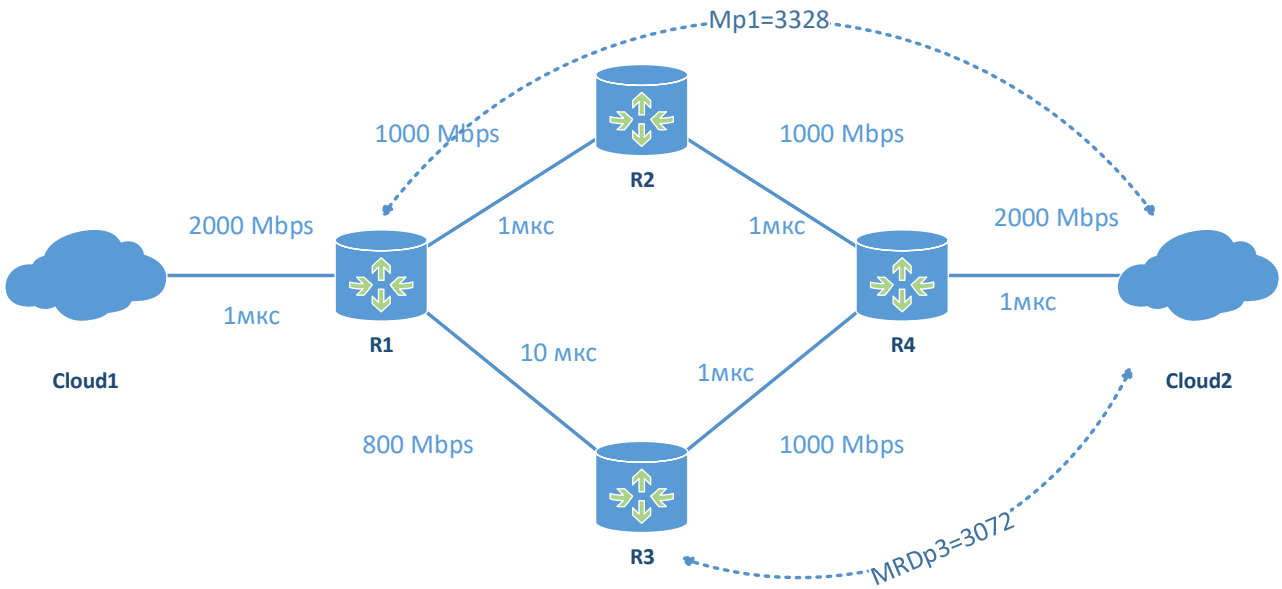

Рис. 2.2. Метрики, необхідні для ухвалення рішення щодо дотримання умови правдоподібності на маршрутизаторі $R1$

Оскільки маршрут стає запасним p_2 і $M_{p_3}^{RD} < M_{p_1}$ може бути використаний для подальшого балансування навантаження.

Для активації балансування навантаження шляхом нееквівалентної вартості необхідно налаштувати ще один параметр під назвою розбіжність (англ. « variance »)

[27]. За допомогою цього параметра EIGRP вибирає маршрути, придатні для балансування трафіку. Параметр розбіжності може бути виражений таким чином:

$$\begin{cases} N_p = \left\lfloor \dfrac{M_{\max}}{M_p} \right\rfloor, якщо\, M_{\min} \cdot V \le M_p \\ N_p = 0, якщо\, M_{\min} \cdot V \ge M_p \end{cases}, \quad (2.1)$$

де N_p – це пропорція кількості пакетів, яка буде передана цим маршрутом pпри балансуванні навантаження, $N \in [1;128]$; V - параметр розбіжності, $V \in [1;128]$; $M_{\max}$ – це максимальна метрика одного з маршрутів до заданої мережі, які беруть участь у балансуванні трафіку (у топології на рис. 2.1). $M_{\max} = M_{p_2}$). При цьому, максимальна кількість маршрутів, якими EIGRP можна балансувати трафік дорівнює 16.

За замовчуванням $V = 1$. Даний параметр вказує, наскільки метрика запасних маршрутів може бути більшою порівняно з мінімальною еталонною метрикою. Якщо метрика маршруту M_p потрапляє під виконання умови $M_{\min} \cdot V \le M_p$ – маршрут p буде вибрано для балансування трафіку.

За замовчуванням маршрутизатори Cisco використовують технологію CEF [28]. Ця технологія дозволяє знизити навантаження на процесор шляхом створення спеціальних таблиць для пакетної комутації. При використанні технології CEF балансування навантаження здійснюється за потоками, а не пакетами. Це призводить до нерівномірного балансування трафіку і навантаження на канали зв'язку є випадковим, оскільки один потік може генерувати більше трафіку, ніж інший. У такому разі, один із каналів зв'язку між $R1 \leftrightarrow R3$ або $R1 \leftrightarrow R2$ може переповнитися набагато швидше. Для усунення подібного явища компанія Cisco запропонувала кілька рекомендацій [30], проте, навіть використовуючи ці підходи, комутація пакетів все ж таки проводиться по потоках.

В умовах нестачі пропускної здатності динамічний протокол маршрутизації не має можливості передавати пакети, які підтримують EIGRP сусідство. Це призведе

до розриву сусідства між маршрутизаторами $R1$ та $R3$ відповідно, до відсутності балансування навантаження. Подібні дослідження наведено у статті [29]. Така ситуація призведе до того, що трафік передаватиметься за маршрутом $R1 \leftrightarrow R2$, що призведе і до його навантаження. Крім проблем із завантаженням каналів зв'язку виникає і додаткове навантаження на центральний процесор маршрутизатора $R1$, оскільки задіється DUAL алгоритм протоколу EIGRP при відновленні сусідства $R3$. Можливими шляхами вирішення цієї проблеми є:

1)Збалансувати пропускні здатності каналів зв'язку між $R1 \leftrightarrow R2$ та $R1 \leftrightarrow R3$ щоб вони стали однаковими.

2)Задавати сусідства EIGRP вручну. При цьому протокол EIGRP не передаватиме пакети для підтримки сусідства і, у разі навантаження, сусідство буде активним і відкидатиметься лише транзитний трафік.

3)Використання трафік шейпінг та трафік полісинг для обмеження вхідного потоку від маршрутизатора.

Всі вище запропоновані способи зосереджені на тому, щоб урівноважити пропускні здатності каналів зв'язку, які використовуються для балансування навантаження або урізання швидкості на вхідному каналі, з метою запобігання перевантаженням. У такому разі втрачається частина пропускної здатності каналів зв'язку.

Одним із можливих шляхів вирішення цієї проблеми також може бути облік коефіцієнта завантаженості каналів зв'язку маршруту у формулі розрахунку метрики EIGRP. З метою перевірки даного припущення було проведено експериментальне моделювання, опис та результати якого представлені в наступному розділі.

2.2. Модифікація для оптимізації протоколу маршрутизації

Пропонується наступна модифікація в роботу протоколу EIGRP. Після повної ініціалізації роботи EIGRP у мережі, коли вона зійшлася:

- на кожному інтерфейсі маршрутизатора, який використовує EIGRP як протокол маршрутизації, проводити розрахунок поточної пропускної здатності

(навантаження) за певний інтервал або за певну кількість пакетів у бітах за секунду;

- якщо завантаження одному з інтерфейсів, наприклад, протягом 1 секунди, менше певної межі, необхідно змінити завантаження EIGRP- інтерфейсу на величину, відповідну падінню пропускної здатності;

- при зміні стану каналу викликати перерахунок маршрутів, які він входить, після чого сповістити про це маршрутизатори-сусіди. У цьому випадку (тобто при зміні топології) на кожному з маршрутизаторів, що залишилися. також повинен статися перерахунок необхідних маршрутів (при необхідності).

Блок-схема алгоритму для одного інтерфейсу представлена на рис. 2.3.

Рис. 2.3. Блок-схема алгоритму стеження за поточною здатністю каналу зв'язку

Максимальний поріг поточної пропускної здатності, після перебільшення якого має бути змінено завантаження інтерфейсу, інші важливі для роботи алгоритму значення та робота елементів, що відповідають за зміну завантаження та перебудову маршрутів, будуть розглянуті та визначені пізніше.

2.3. Загальні відомості про OMNeT++

OMNeT++ (англ. Objective Modular Network Testbed in C++) [22] – модульна C++ бібліотека моделювання, що розширюється, на основі компонентів, призначена для створення мережевих симуляторів. Цей продукт з відкритим вихідним кодом є безкоштовним лише для академічного та некомерційного використання. Тут мережа розуміється у ширшому сенсі, включаючи провідні та бездротові мережі зв'язку, чіпи, які побудовані по архітектурі мережа на чипі (мається на увазі, що кожне обчислювальне ядро пов'язане безпосередньо тільки з найближчими ядрами), мережі масового обслуговування і так далі.

Предметно-орієнтована функціональність (підтримка сенсорних мереж, бездротових ad-hoc-мереж або бездротових динамічних (самоорганізованих) мереж, інтернет-протоколів, моделювання продуктивності, фотонних мереж тощо) забезпечується модельними фреймворками, розробленими як самостійні проекти. OM-Net++ надає IDE на основі Eclipse, графічне середовище виконання, а також безліч інших інструментів. Є розширення для моделювання у реальному часу, емуляції мережі, інтеграції з базами даних, з SystemC і низку інших функцій.

OMNeT++ надає компонентну архітектуру для моделей. Компоненти (модулі) запрограмовані в C++, потім зібрані у більші компоненти і моделі з використанням мови високої рівня (NED). Цей інструмент має розширену підтримку графічного інтерфейсу користувача, та завдяки своїй модульної архітектурі ядро моделювання (і моделі) може бути легко вбудовано в користувацькі програми.

OMNeT++ працює на Windows, Linux, Mac OS X та інших UNIX-подібних системах і складається з наступних компонентів:

- бібліотека ядра моделювання;

- мова описи топології NED (англ. NEtwork Description);
- IDE на базі платформи Eclipse;
- GUI для виконання моделювання Tkenv, лінкується в виконуваний файл симуляції;
- інтерфейс командної рядки для виконання моделювання (Cmdenv);
- утиліти (інструмент створення makefile і так далі);
- документація, приклади моделей та інше.

OMNeT++ надає ефективні інструменти для користувача, щоб описати структуру реальною системи. До головних особливостей можна віднести ієрархічно вкладені модулі, які є екземплярами типів модулів, та мову опису топології NED. Модулі зв'язуються повідомленнями каналами, мають гнучкі параметри.

Модель OMNeT++ складається з наступних частин:

- описи топологій на мовою NED (файли з розширенням. Ned), які описують структуру модуля з параметрами, шлюзами тощо;
- визначення повідомлень (.msg файли). Можна визначити різні типи повідомлень і додати поля даних в них. OMNeT++ автоматично переведе визначення повідомлень у повноцінні класи C++;
- вихідні коди модуля представляють собою C++ файли з розширенням.h або.cc.

Система моделювання надає такі компоненти, як ядро моделювання, яке керує моделюванням і класовими бібліотеками моделювання, і інтерфейси користувача.

Програми моделювання побудовані з вказаних вище компонентів. Спочатку.msg файли перетворюються в код на C++ з використанням компілятора повідомлень opp_msgc. Після все вихідники на C++ компілюються і лінкуються з ядром моделювання та бібліотекою інтерфейсу користувача для формування виконуваного файлу моделювання або загальної бібліотеки. NED-файли завантажуються динамічно у вихідних текстових формах під час запуску програми моделювання.

Після запуску програма зчитує всі файли NED, потім конфігураційний файл (зазвичай називається omnetpp.ini). Цей файл містить параметри, які визначають, як виконується моделювання, значення параметрів моделі тощо.

Процес побудови та запуску програм моделювання представлений на рис. 2.4.

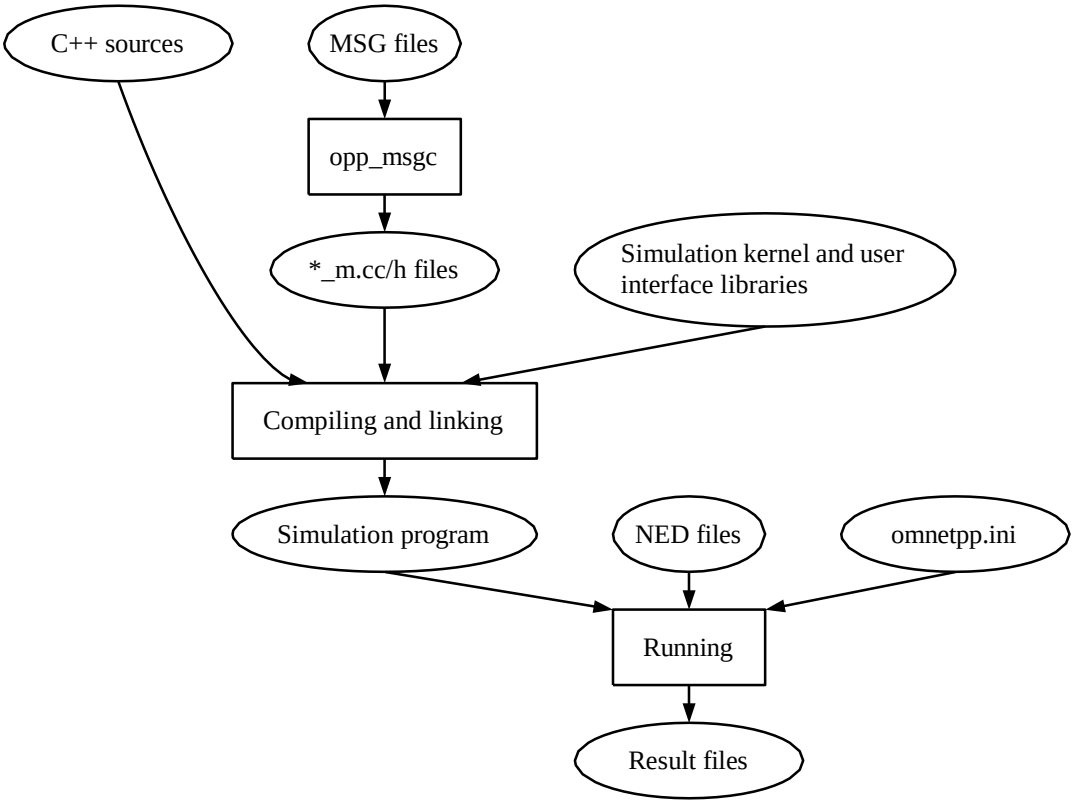

Рис. 2.4. Огляд процесу побудови і запуску програм моделювання, створених у OMNeT++

Вихідні дані моделювання записуються в файли результатів, які можуть бути проаналізовані за допомогою OMNeT++: вихідні файли векторів (запис даних під час виконання моделювання), вихідні файли скалярів (підсумкові результати, обчислені під час моделювання та записані після завершення моделювання), вихідні файли, визначені користувачем.

OMNeT++ може бути розширений за допомогою спеціальних бібліотек моделювання. Найбільш відомої і поширеною є INET Framework [23], вихідний код якої відкрито. INET надає протоколи, агенти та інші моделі для дослідників та студентів, що працюють із мережами передачі даних. INET особливо корисна при

розробці та затвердженні нових протоколів, при вивченні нових екзотичних сценаріїв.

INET містить моделі для стека протоколів Інтернету (TCP, UDP, IPv4, IPv6, OSPF, BGP тощо), для дротових та бездротових протоколів каналу рівня (Ethernet, PPP, IEEE 802.11 і так далі), для підтримки мобільності, протоколів MANET (Mobile Ad hoc Network), DiffServ, MPLS з LDP та RSVP-TE-сигналізацією, включає кілька моделей додатків, багато інших протоколів та компонентів.

Існує безліч заснованих на INET фреймворків моделювання, які підтримуються незалежними дослідницькими групами. До числу таких бібліотек відноситься ANSAINET (англ. Automated Network Simulation and Analysis (ANSA) [24] – проект дослідників та студентів Технологічного університету Брно факультету інформаційних технологій в Чехії. Він присвячений розробці різних імітаційних моделей, сумісних зі специфікаціями RFC або еталонними (базовими) реалізаціями (англ. referential implementations), розширюючи цим дротовий мережевий IP-фреймворк INET (тому вихідний код називається ANSAINET). Згодом ці модулі та пов'язані з ними інструменти могли б дозволити проводити формальний аналіз. реальних мереж і їх конфігурацій. ANSAINET може вільно використовуватись в якості основи для маршрутизації/комутації для подальших дослідницьких ініціатив, то є в симуляції, доказують (або спростовують певні аспекти мережевих технологій (наприклад, пошук вузьких місць і єдиних точок відмови, помилок конфігурації, несправних станів мережі тощо).

Основні цілі проекту ANSAINET:

- розробити точні імітаційні модулі (сумісні з базовими реалізаціями) для протоколів, використовуваних в традиційних провідних TCP/IP- мережах;

- популяризувати використання симуляторів як засобів верифікації та валідації під час розгортання мереж та розробки протоколів;

- зробити внесок і надати підтримку у спільноті OMNeT++, інтегруючи популярні функції ANSAINET до оригінального INET.

У ANSAINET підтримуються наступні протоколи:

- HSRP, VRRPv2, GLBP;

- IS- IS;
- RIPv2 і RIPng, EIGRP, Babel;
- CDP, LLDP;
- STP, TRILL;
- LISP;
- PIM-DM, PIM-SM, IGMPv2 і IGMPv3.

Для моделювання роботи комп'ютерної мережі з протоколом EIGRP використовуватиметься бібліотека ANSAINET версії 3.4.0 (рис. 2.5.).

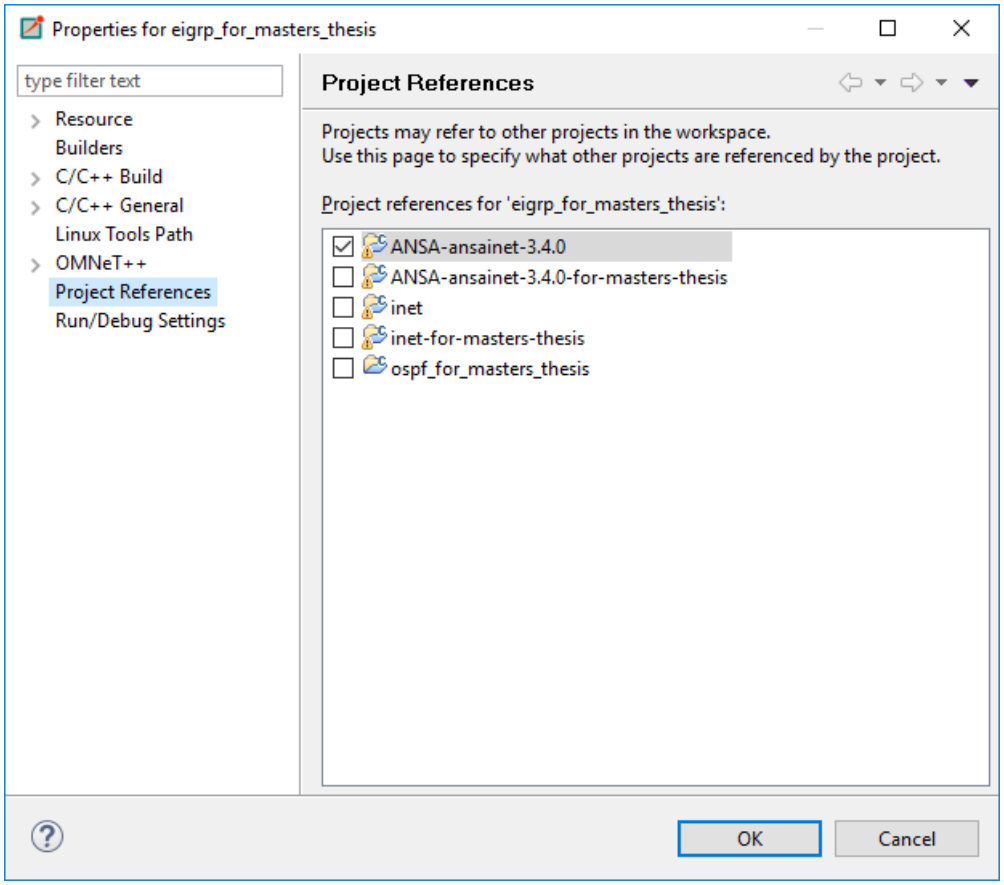

Рис. 2.5. Вибір ANSA-ansainet-3.4.0 для використання в проекті OMNeT++

2.4. Реалізація методу в бібліотеці ANSAINET для OMNeT++

У ході реалізації запропонованого методу були використані наступні вихідні класи та файли бібліотеки ANSAINET (табл. 2.1).

Таблиця 2.1

Класи та файли бібліотеки ANSAINET

Назва	Призначення
EtherEncap (файли EtherEncap.h і EtherEncap.cc)	реалізує інкапсуляцію та декапсуляцію Ethernet-фреймів
IPv4 (файл IPv4.cc)	втілює протокол IPv4
ANSA_MultiNetworkLayerLowerMultiplexer (файл ANSA_MultiNetworkLayer Lower Multiplexer.cc)	клас, який, ймовірно, за мультиплексування мережевих шарів нижче даного
EigrpIpv4Pdm (файли EigrpIpv4Pdm.h і EigrpIpv4Pdm.cc)	модуль EIGRP для IPv4, що залежить від протоколу
EigrpIpv6Pdm (файл EigrpIpv6Pdm.cc)	модуль EIGRP для IPv6, що залежить від протоколу
EigrpMetricHelper (файл EigrpMetricHelper.cc)	клас для обчислення метрики EIGRP
EigrpDeviceConfigurator (файл EigrpDeviceConfigurator.cc)	відповідає за початкову конфігурацію мережі EIGRP
EigrpMessage.msg	файл, в якому описані пакети EIGRP

Для того, щоб EIGRP враховував завантаження інтерфейсу при обчисленні метрики, потрібно встановити значення коефіцієнта K2 в 1 в наступні файли протоколу EIGRP (рис. 2.6-2.7)

```
// Struct for K-values
struct EigrpKValues
{
      ...
      uint16_t K2 = 1;
      ...
}

- EigrpIpv4Pdm.cc:

EigrpIpv4Pdm::EigrpIpv4Pdm() : EIGRP_IPV4_MULT(
                               IPv4Address(224, 0, 0, 10))
{
    ...
    kValues.K1 = kValues.K2 = kValues.K3 = 1;
    kValues.K4 = kValues.K5 = kValues.K6 = 0;
    ...
}
```

Рис. 2.6. Лістинг файлу EigrpMessage.msg

```
void EigrpDeviceConfigurator::loadEigrpProcessesConfig(
     cXMLElement *device, IEigrpModule<IPv4Address>
     *eigrpModule)
{
    ...
    kval.K2 = loadEigrpKValue((*procElem), "k2", "1");
    ...
}
```

Рис. 2.7. Лістинг файлу EigrpDeviceConfigurator.cc

Файл EigrpIpv6Pdm.cc аналогічно файлу EigrpIpv4Pdm.cc. І аналогічно в функції:

```
void EigrpDeviceConfigurator:: loadEigrpProcesses6Config(cXMLElement
*device, IEigrpModule<IPv6Address>
*eigrpModule.
```

Тоді формулу метрики (1.1) можна записати у вигляді:

$$metric_1 = BW + \frac{BW}{256 - LOAD} + DELAY \quad (2.2)$$

У файл ANSA_EIGRP_Router.ned що містить опис структури EIGRP-маршрутизатора (рис. 2.8), були додані змінні:

– bool isThresh-oldPassed для попередження про те, що пропускна здатність сильно впала і потрібно зробити необхідні дії;

– int ifIndex для збереження номера інтерфейсу, на якому спостерігається зниження даного параметра, для подальшої зміни рівня навантаження на ньому в підмодулі eigrpIpv4Pdm модуля eigrp типу EigrpProcessDS (рис. 2.9).

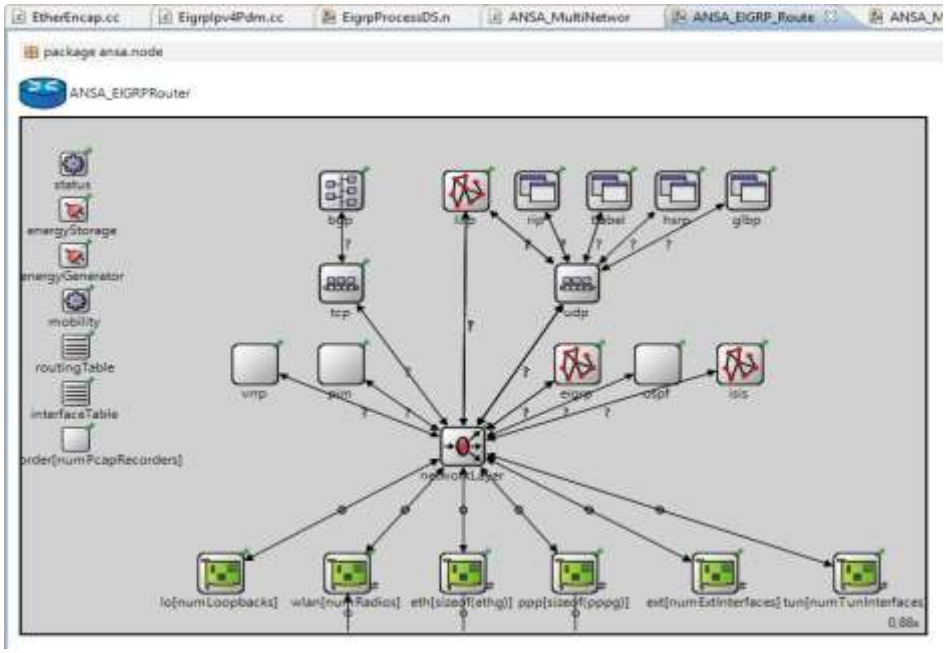

Рис. 2.8. Структура модуля ANSA_EIGRPRouter, що містить опис структури маршрутизатора EIGRP у OMNeT++

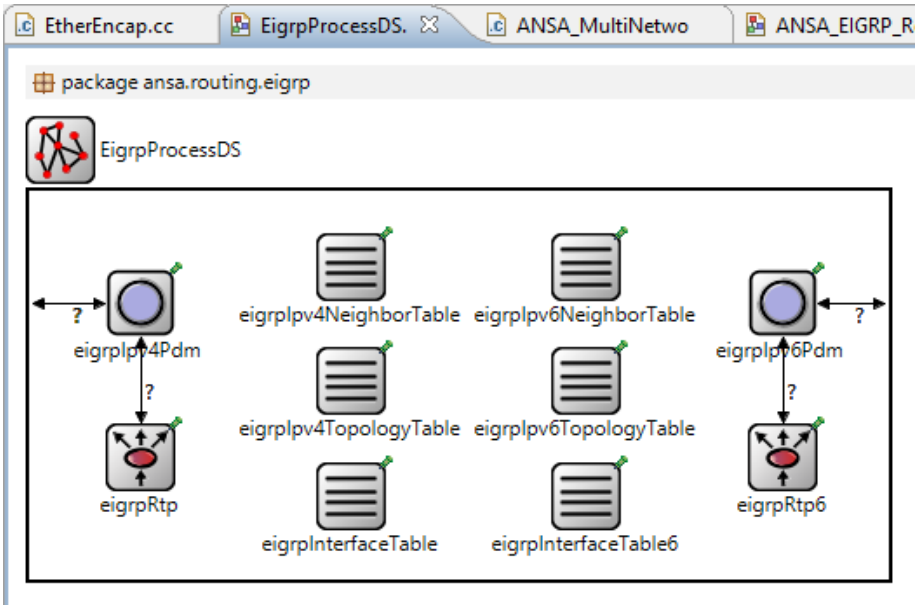

Рис. 2.9. Структура модуля EIGRP типу EigrpProcessDS, що описує процес маршрутизації EIGRP

Опис маршрутизатора у цьому файлі виглядає так (рис. 2.10):

```
package ansa.node;
package ansa.node;
import ansa.node.ANSA_Router;
module ANSA_EIGRPRouter extends ANSA_Router
{
    parameters:
        hasEIGRP = true;
        eigrp.configData =
        configData;
        bool isThresholdPassed =
        default(false); int ifIndex =
        default(-1);
}
```

Рис. 2.10. Вміст файлу ANSA_EIGRP_Router.ned

Пропускна здатність розраховується на рівні Ethernet, значення має кількість інкапсульованих кадрів. В клас class INET_API Ether-Encap : public cSimpleModule були введені такі додаткові дані (властивості), опис змінних наведено в таблиці 2.2:

```
simtime_t startTime; // start time
unsigned int batchSize; // number of packets in a batch
// max length of measurement interval (measurement ends
// if either batchSize or maxInterval is reached, whichever
// is reached first)
    simtime_t maxInterval;
simtime_t timestamp; // time when threshold passed
// current measurement interval
    simtime_t intvlStartTime;
    unsigned long packetsPerIntvl;
    unsigned long bitsPerIntvl;
    double threshold, // threshold value
          oldThreshold,
          newEigrpIfaceLoad; // new EIGRP interface
    load bool shouldChangeLoad;
cModule *ethIface; // EthernetInterface module (network card)
```

У методі цього класу void EtherEncap::initialize()ініціалізуються нові дані:

```
startTime = 0;
batchSize = 25;
maxInterval = 1;
timestamp = -1;
intvlStartTime = 0;
packetsPerIntvl = bitsPerIntvl = 0;
    threshold = oldThreshold = 0;
    newEigrpIfaceLoad = 1;
    shouldChangeLoad = false;
ethIface = getParentModule();
    WATCH(intvlStartTime);
```

```
WATCH(packetsPerIntvl);
WATCH(bitsPerIntvl);
WATCH(timestamp);
WATCH(shouldChangeLoad);
```

Таблиця 2.2

Опис введених змінних для вимірювання пропускної здатності

Змінна	Опис змінної
simtime_t startTime	Час початку моделювання
unsigned int batchSize	Кількість пакетів в партії, по пришестю яких починається новий вимірювальний інтервал
simtime_t maxInterval	Максимальна довжина вимірювального інтервалу
simtime_t timestamp	Час, коли максимальний поріг пропускної здатності було пройдено
simtime_t intvlStartTime	Час початку нового інтервалу
unsigned long пакетівPerIntvl	Кількість пакетів за інтервал
unsigned long bitsPerIntvl	Кількість бітів за інтервал
double threshold	Порогове значення
bool shouldChangeLoad	Потрібно чи ні збільшити завантаження на інтерфейсі
cModule *ethIface	Покажчик на поточний Ethernet-інтерфейс, де відбувається розрахунок пропускної здатності

Розмір набору пакетів batchSize в кількості 25 підібрано шляхом експериментів при моделюванні. Після досягнення даного значення знову розраховується значення пропускної здатності.

Визначення пропускної здатності double bitpersec відбувається у функції void EtherEncap::processPacketFromHigherLayer(cPacket *msg), що обробляє пакети, від вищих рівнів маршрутизатора (Додаток Б).

Якщо пропускна здатність впала на 30% і більше, то відбувається фіксація цього моменту часу в timestamp, параметр isThresholdPassed маршрутизатора встановлюється значення true, а в ifIndex записується номер інтерфейсу, у якому сталося падіння продуктивності. Значення порога threshold = 0.70 * bitpersec; було обрано таким під час експериментів, воно відповідає падінню продуктивності, що

створюється при передачі даних під час моделювання і однаковий для всіх моделей мереж.

Далі, якщо відбувається зменшення пропускної здатності і потрібно поміняти завантаження інтерфейсу, формується повідомлення TCN типу TplgyChngNtfctn, що сигналізує про зміну топології мережі, яка відправляється вищележачому рівню (модулю) пристрою, в даному випадку - модулю (AN networkLayer, що відповідає за роботу протоколу IPv4 та деяких інших протоколів мережевого рівня моделі OSI. У цьому повідомленні дублюється номер інтерфейсу ifIndex і вказується нове навантаження для EIGRP-каналу, що відповідає падінню пропускної здатності.

Наприклад, якщо максимальна швидкість передачі даних каналу дорівнює 100 Мбіт/с, і поточна швидкість передачі даних – 50 Мбіт/с, то нове навантаження на EIGRP-інтерфейс дорівнюватиме 127. Для створення та використання такого TCN-повідомлення потрібно було реалізувати його в.msg файлі.

Був створено файл TCN.msg :

```
namespace inet;
// Topology Change Notification message TplgyChngNtfctn
{
    string source; string destination;
    bool shouldChangeTopology = false; short ifIndex = -1;
    double newEigrpIfaceLoad = 1;
}
```

При наступній компіляції автоматично створюються файли TCN_m.cc та TCN_m.h з C++ класом class TplgyChngNtfctn: public::omnetpp::cMessage, який можна використовувати в модулях. Ці файли містять реалізацію створеного класу (у тому числі сетери та гетери). Для використання класу потрібно підключити згенерований заголовний файл (це було зроблено для всіх використаних у роботі класів):

```
#include "inet/linklayer/ethernet/TCN_m.h"
```

Модуль мережевого рівня networkLayer EIGRP-маршрутизатора, у свою чергу, складається з множини інших підмодулів (рис. 2.11), де теж потрібно обробляти будь-яке повідомлення.

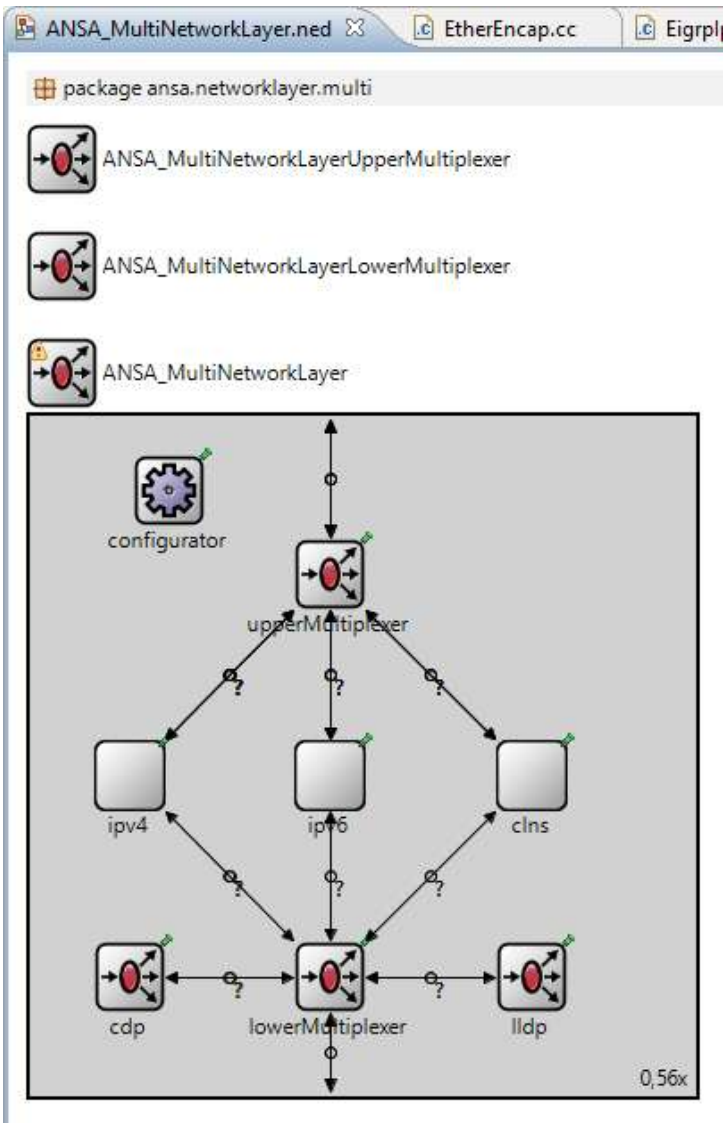

Рис. 2.11. Структура модуля ANSA_MultiNetworkLayer, що відповідає за роботу різних протоколів мережного рівня

Для можливості коректного перенаправлення TCN-повідомлення в цьому модулі змінювалися тільки підмодулі lowerMultiplexer та ipv4. У підмодулі upperMultiplexer TCN правильно приймається і надсилається без додаткових маніпуляцій з вихідним кодом класу цього підмодуля. Для lowerMultiplexer зміни піддався метод void ANSA_MultiNetworkLayerLower Multiplexer::handleMessage (cMessage *message):

```
// received message is a TCN message,
// send it to IPv4NetworkLayer module
if (dynamic_cast<TplgyChngNtfctn *>(message))
{
    cMessage *msgCopy = message->dup(); delete message;
```

```
    send(msgCopy, "ifUpperOut",
         getProtocolCount() * arrivalGate->getIndex() + 0);
}
else
{
    int res = getProtocolIndex(message);
    ...
```

Коли надіслане повідомлення надійшло до модуля ipv4, відбувається його відправлення до upperMultiplexer і далі до EIGRP. Це здійснюється в void IPv4::handleMessage(cMessage *msg):

```
// received message is a TCN message, send it to
// ANSA_MultiNetworkLayerUpperMultiplexer module else if
    (dynamic_cast<TplgyChngNtfctn *>(msg))
{
    cMessage *msgCopy = msg->dup(); delete msg;
    cGate *outGate = gate("transportOut", 0); send(msgCopy,
        outGate);
}
```

При прийнятті модулем, залежним від протоколу, (EigrpIpv4Pdm) eigrpIpv4Pdm модуля (EigrpProcess) eigrp TCN-повідомлення в функції обробки повідомлень void EigrpIpv4Pdm::handleMessage(cMessage *msg) починається перерахунок маршрутів, до яких входить даний інтерфейс:

```
// received TCN, take necessary actions if
    (dynamic_cast<TplgyChngNtfctn *>(msg))
{
        TplgyChngNtfctn *tcnMsg = check_and_cast<TplgyChngNtfctn *>
                                  (msg);
    EV << "Received TCN: (" << tcnMsg->getClassName() << ")"
       << tcnMsg->getName() << "\n";
    if(tcnMsg->getShouldChangeTopology())
    {
        // try to find the necessary EIGRP interface
        // to change the load on it
        int ifaceId = tcnMsg->getIfIndex(); EigrpInterface
            *eigrpIface =
              getInterfaceById(INTERFACEIDS_START + ifaceId); if
            (eigrpIface == NULL)
        {
            EV << "EIGRP interface eth" << ifaceId << "("
                   << INTERFACEIDS_START + ifaceId << ") is not
                found\n"; delete msg;
            delete tcnMsg; return;
        }
```

```
            // get and set a new load
            int newEigrpIfaceLoad = tcnMsg->getNewEigrpIfaceLoad();
                if(newEigrpIfaceLoad != eigrpIface->getLoad())
            {
                eigrpIface->setLoad(newEigrpIfaceLoad); if (eigrpIface-
                    >isEnabled())
                   processIfaceConfigChange(eigrpIface);
                // else do nothing (interface is not part of EIGRP)
            }
        }
        delete msg; delete tcnMsg;
    }
```

Щоб функція processIfaceConfigChange(eigrpIface) працювала правильно, потрібно змінити функцію EigrpWideMetricPar класу EigrpMetricHelper :

```
newMetricPar.load = getMax(ifParam.load, neighParam.delay);
```

замінити на

```
newMetricPar.load = getMax(ifParam.load, neighParam.load);
```

З новими маршрутами, отриманими на основі реального завантаження, дані будуть передаватися іншим шляхом, де рівень трафіку може бути низьким.

Висновки до розділу 2.

Отже в даному розділі розглянуто методи з допомогою яких можна здійснювати оптимізацію передачі даних. Запропоновані зміни до протоколу маршрутизації EIGRP в контексті контролю навантаження на інтерфейси маршрутизатора. І в залежності від цього здійснювати перенаправлення трафіку.

РОЗДІЛ 3
АПРОБАЦІЯ ЗАПРОПОНОВАНИХ МЕТОДІВ ПІДВИЩЕННЯ ПРОДУКТИВНОСТІ ПЕРЕДАЧІ ДАНИХ

3.1. Узагальнена структура мережі

Недолік протоколів EIGRP і OSPF при роботі в мережах під навантаженнями, який описаний раніше в кваліфікаційній роботі, наочно можна, показати на мережах з загальною структурою, наведеною рис. 3.1.

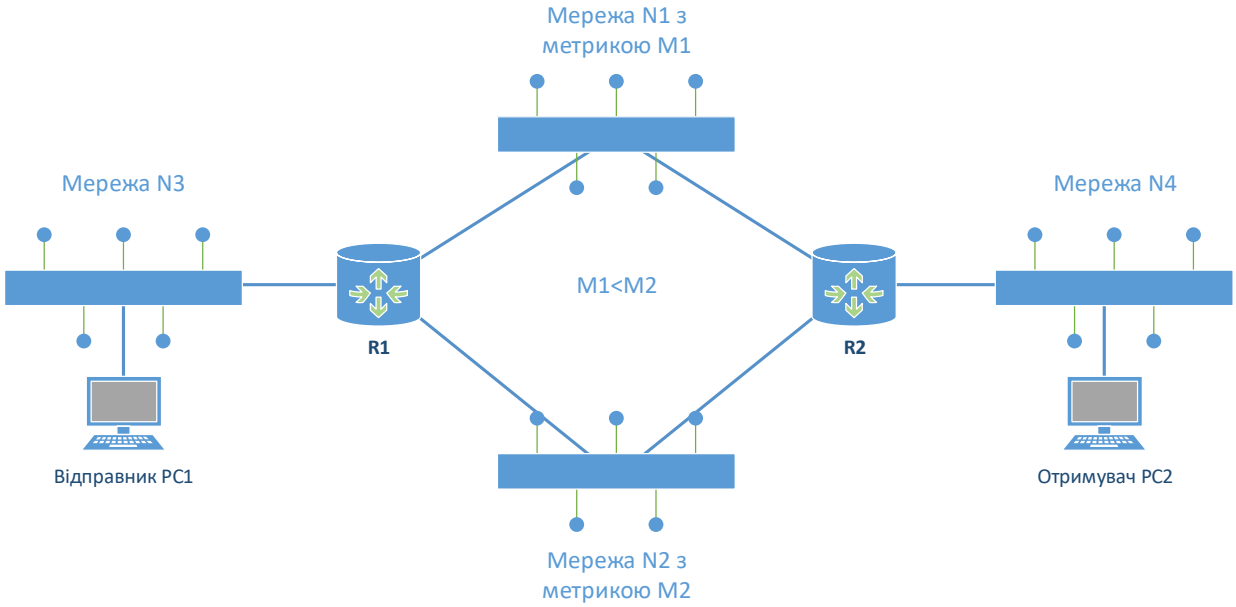

Рис. 3.1. Структурна схема мережі для демонстрації роботи протоколів EIGRP та OSPF

Тут інформація передається від відправника PC1 до одержувача PC2. При цьому інфраструктура мережі N3 і N4 не така важлива, їх може і не бути, в цьому випадку PC1 безпосередньо з'єднаний з маршрутизатором R1, а PC2 – з R2. Топологія мереж N1 і N2 не має значення і може бути довільною.

Основна умова – наявність розгалуження на R1, який є маршрутизатором, який визначає по якому шляху направити трафік. Тоді, якщо на одному з двох можливих маршрутів на шляху трафіку через R1 або R2 знаходиться мережа N1 з

метрикою M1, а на другому – мережа N2 з метрикою M2 і, наприклад, M1 явно краще (менше) M2, згідно з алгоритмами роботи протоколів EIGRP і OSPF дані будуть йти шляхом, що проходить через мережу N1.

При виникненні в мережі N1 такого додаткового трафіку, при якому загальне навантаження сильно збільшиться, швидкість передачі даних впаде, з'являється необхідність перенаправити трафік користувача через мережу N2. Але в EIGRP та OSPF немає механізмів, які забезпечують цю можливість.

Для моделювання підійдуть будь-які варіації описаної мережі зі структурою рис. 3.1. На такому варіанті віртуальної мережі будемо проводити подальші дослідження запропонованих методів.

3.2. Моделювання мережі з протоколами маршрутизації EIGRP і OSPF при змінних навантаженнях

Для проведення моделювання в OMNeT++ IDE було створено мережу, модель якої представлена рис. 3.2.

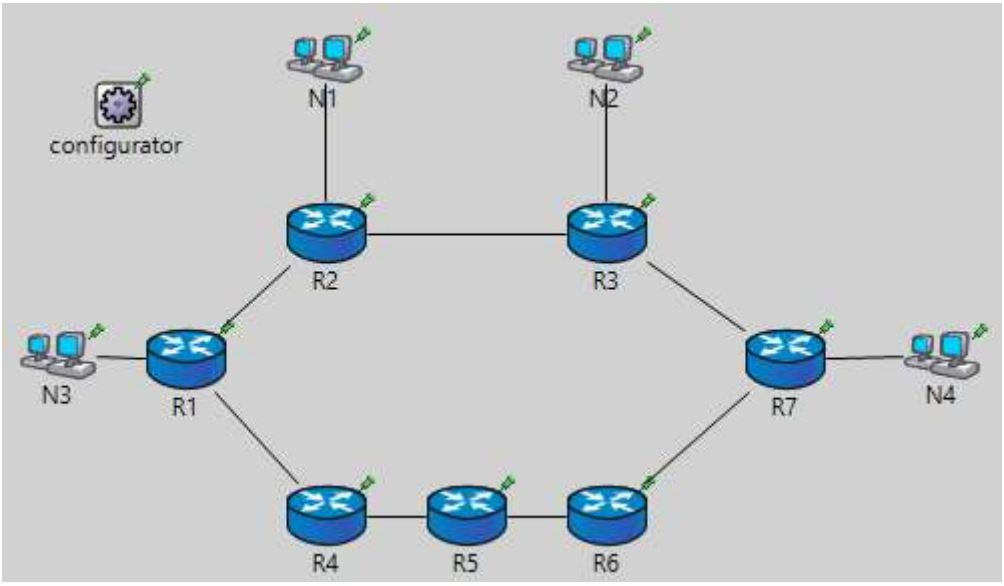

Рис. 3.2. Модель мережі, написана в OMNeT++ IDE

Модель описано в файлі EigrpNet.ned на мові NED, код якого наведено у додатку В.

Конструкція channel C extends ThruputMeteringChannel {} в EigrpNet.ned описує тип каналу з'єднання, це дозволяє використовувати тільки ім'я цього типу в секції connections замість повторення одного і того ж визначення каналу; вибрано такі параметри з'єднання:

- затримка з'єднання дорівнює 0,1 мкс;
- швидкість передачі даних каналу дорівнює 100 Мбіт/с;
- thruputDisplayFormat = "#N" дозволяє виводити загальну кількість прийнятих та надісланих пакетів на інтерфейсі у графічній оболонці Tkenv під час виконання моделювання.

Цей тип каналу використовується для всіх з'єднань (секція connections) у наведеній моделі, які, до тому ж, є двосторонніми.

Модуль module EigrpLan {} визначає локальну мережу з довільною кількістю хостів h в ній. У кожною такий мережі хаб типу EtherHub з'єднаний з усіма хостами ANSA_Host і з одним з маршрутизаторів загальної моделі по Ethernet-інтерфейсам ethg.

Тестова мережа EigrpNet складається з семи маршрутизаторів R1-R7, кожен з яких має тип ANSA_EIGRPRouter і кілька Ethernet-інтерфейсів, чотирьох локальних мереж N1-N4 типу EigrpLan і конфігуратора IPv4NetworkConfigurator, Котрий може автоматично призначати IP-адреси на всіх пристроях і мережах. Але через обмежень, введених розробниками AN-SAINET, основні функції IPv4-конфігуратора при на будівництві мережі не використовуються.

Конфігурація створюваної моделі описана в файлі omnetpp.ini (додаток Д).

Всі EIGRP-маршрутизатори належать до однієї області, ці та інші налаштування тестової мережі знаходяться у файлі config.xml (Додаток Е).

На Ethernet-інтерфейсах всіх пристроїв мережі використовується так званий outputHook типу ThruputMeter, що дозволяє виміряти пропускну здатність, що потім буде використано для візуалізації результатів моделювання.

Щоб створити навантаження на модель мережі через файл omnetpp.ini були створені програми рівня TCP на деяких хостах локальних мереж N1-N4, а саме:

- на одному з хостів мережі N3 створюється додаток, що надсилає

2000000 байт одному з хостів мережі N4, на якому ініціалізується додаток, що надсилає ці дані назад. Час початку відправлення – 50,10 с;

- аналогічна ситуація і в мережах N1 і N2: хост мережі N1 в 50,15 з відправляє хосту мережі N2 1000000 байт, а той відправляє їх назад. Такою маніпуляцією створюється додаткове навантаження на маршрутизатори R2 і R3.

Після запуску проекту створюється симуляція із налаштованою через файл config.xml мережею (рис. 3.3) по закінченню моделювання були отримано графік навантаження інтерфейсів eth[0] що з'єднаний з інтерфейсом eth[0] маршрутизатора R1 та eth[2] що з'єднаний з Ethernet-інтерфейсом хоста з мережі N1 маршрутизатора R2, який представлений на рис. 3.4.

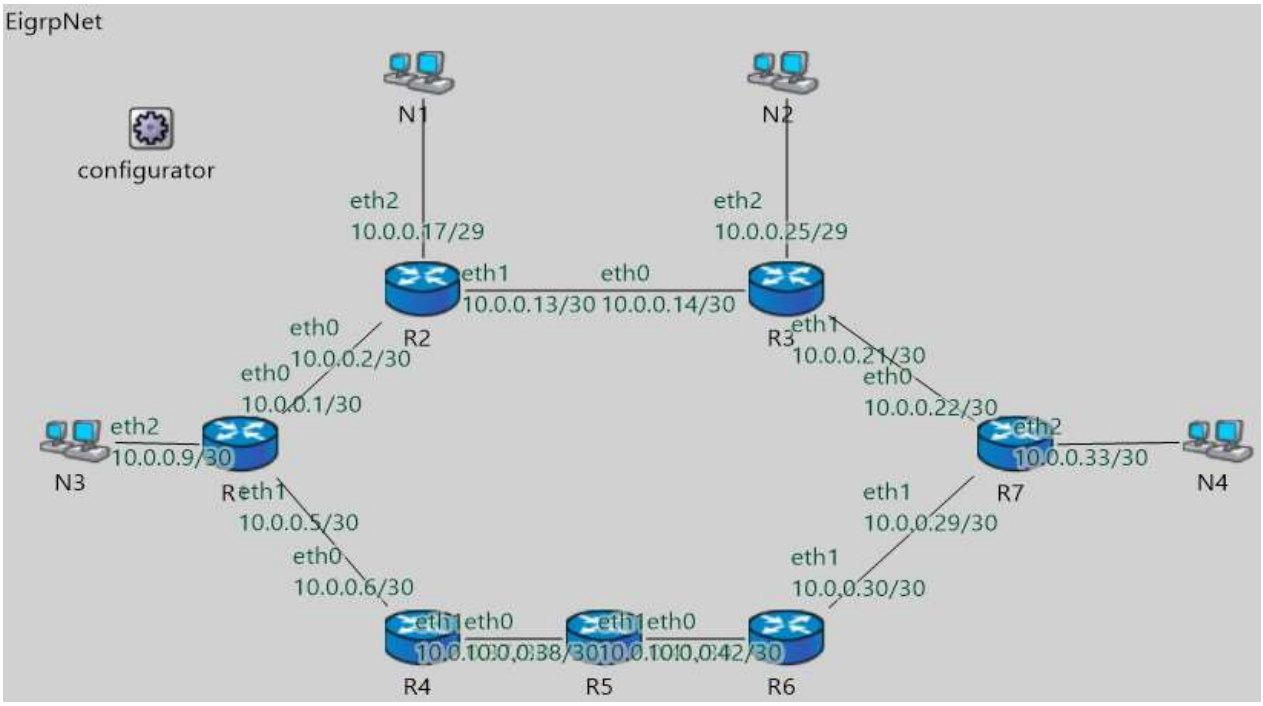

Рис. 3.3. Налаштування мережі при запуску програми моделювання

У момент часу, приблизно рівний 50,15 с дійсно відбувається провал у продуктивності, що обумовлено механізмом роботи протоколу TCP, який динамічно регулює розмір вікна (число, яке визначає в байтах розмір даних, які відправник може надіслати без отримання підтвердження) для зменшення втрат даних.

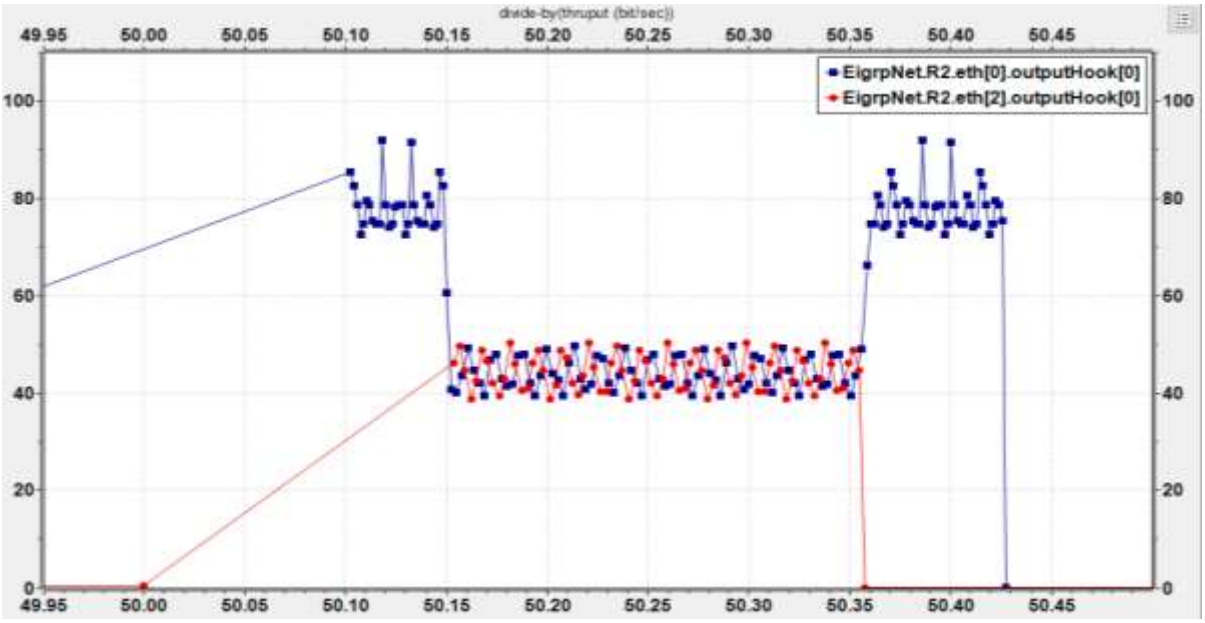

Рис. 3.4. Отриманий графік зміни пропускної здатності каналів маршрутизатора R2 (інтерфейси eth[0] та eth[2]) для EIGRP-мережі

Для порівняння рис. 3.5 наведено графік завантаження інтерфейсів eth[0] та eth[2] маршрутизатора R2 мережі зі структурою (рис 3.3), але на основі протоколу OSPF.

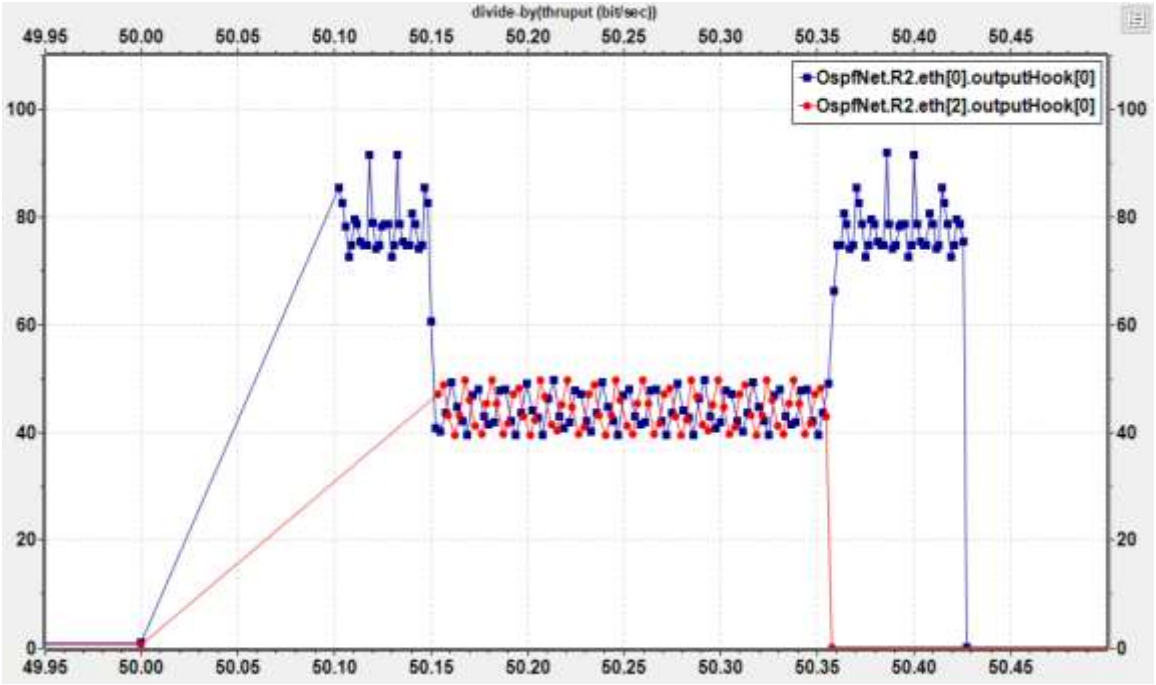

Рис. 3.5. Отриманий графік зміни пропускної здатності каналів маршрутизатора R2 (інтерфейси eth[0] та eth[2]) для OSPF-мережі

Зміни в конфігурації моделі мінімальні і описуватись не будуть. Як видно, результати ідентичні.

Це пов'язано з тим, що ні OSPF, ні EIGRP не враховують поточну пропускну здатність каналів зв'язку для оптимізації навантаження на мережу (наприклад, шляхом відповідної зміни метрики на інтерфейсі та подальшої перебудови таблиці маршрутизації маршрутизатора).

Але в EIGRP можливо внести зміни в процес розрахунку сумарної метрики для шуканої мережі призначення, змінивши значення коефіцієнта K_2 з 0 на 1. Після цього при розрахунку вартості маршруту враховуватиметься параметр $LOAD$ – величина навантаження на лінію, що визначається динамічно як 5-хвилинне експоненційно зважене середнє, яке оновлюється кожні 5 секунд. Однак і в цьому випадку EIGRP розглядає завантаження інтерфейсу тільки при відправці оновлення по якийсь інший причини (наприклад, при відмові каналу зв'язку або зміні топології мережі): оновлення не анонсуються в мережу щоразу при зміні навантаження.

На рис. 3.6 показано кількість втрачених пакетів на інтерфейсах маршрутизатора R2 для EIGRP.

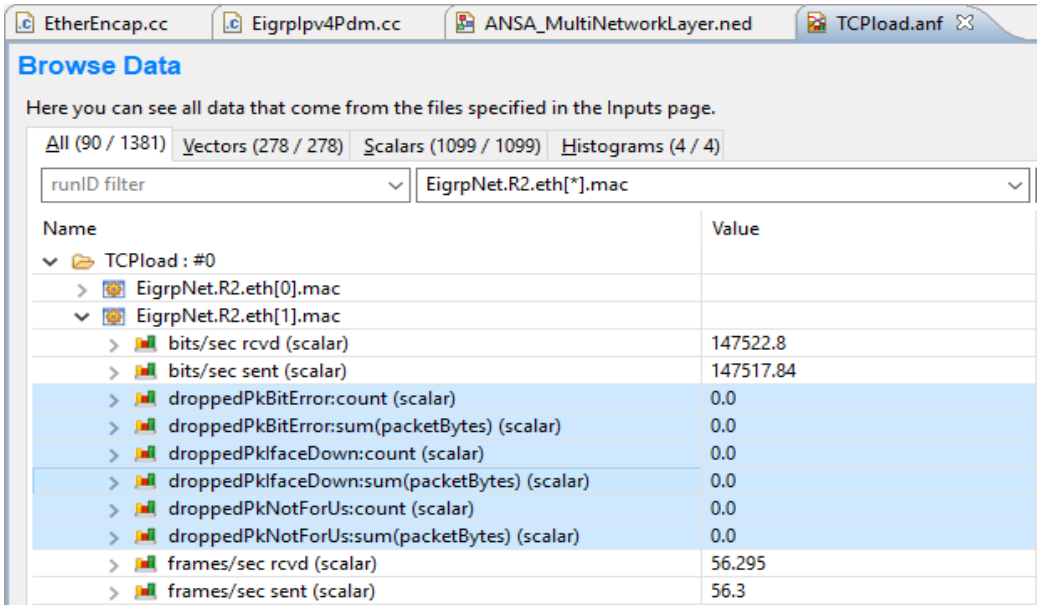

Рис. 3.6. Кількість втрачених пакетів на інтерфейсах маршрутизатора R2 для EIGRP

Як бачимо їх немає, що відповідає нормальній роботі протоколу та мережі.

3.3. Моделювання мережі з оптимізованим протоколом EIGRP при пікових навантаженнях

Моделювання для підтвердження працездатності розробленого методу оптимізації EIGRP здійснюється на моделі мережі, описаної раніше.

Значення основних параметрів для цією моделі:

- batchSize = 25 ;
- maxInterval = 1 ;
- threshold = 0.70 * bitpersec.

При моделюванні на інтерфейсі eth[0] маршрутизатора R1 (рис. 3.7) було зафіксовано перевантаження каналу.

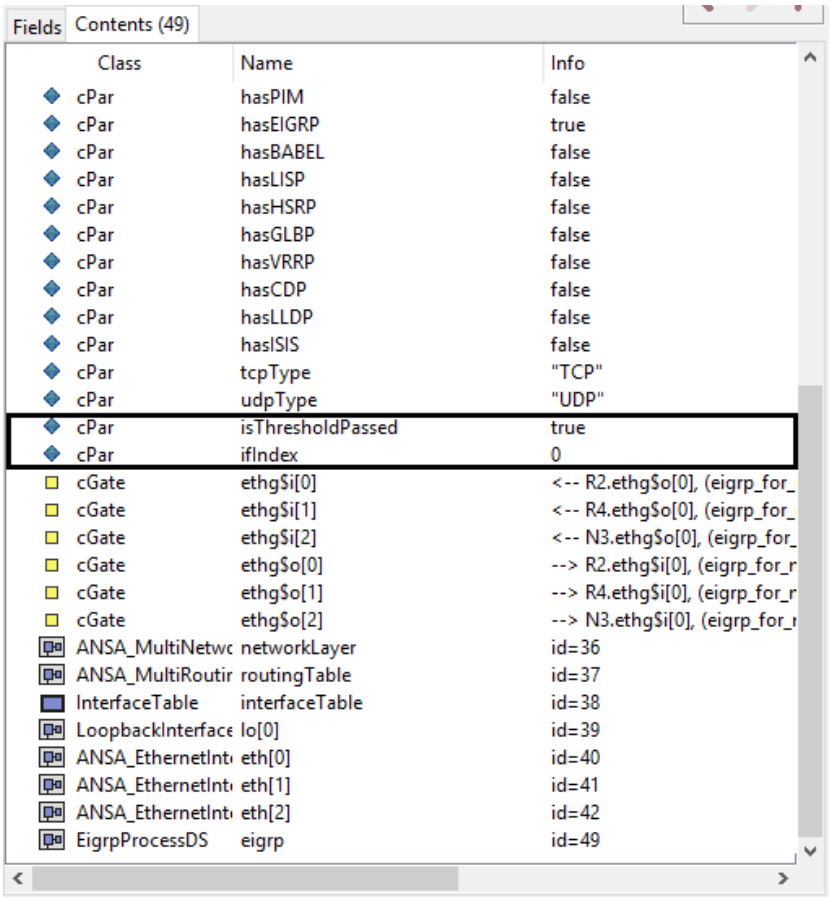

Рис. 3.7. Зміна значень параметрів isThresholdPassed і ifIndex на маршрутизаторі R1 при сильному зменшенні пропускної здатності

На R1 у параметр timestamp було збережено час коли відбулось падіння пропускної здатності (рис. 3.8), після чого модулем EIGRP було отримано TCN-повідомлення (рис. 3.9.), було змінено навантаження інтерфейсу eth[0], відбулося перебудова таблиць топології та маршрутизації, сусідам було розіслано маршрутні оновлення (рис.3.10-3.12).

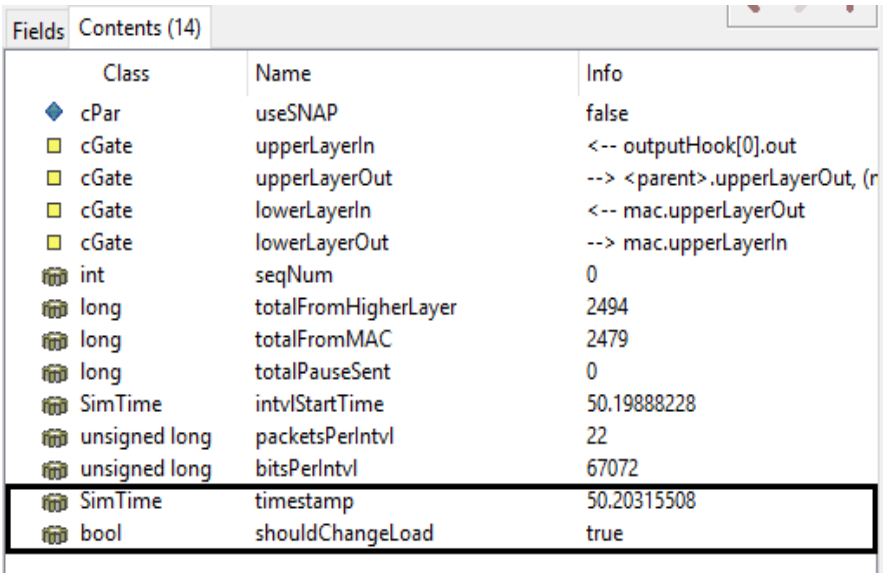

Рис. 3.8. Значення параметрів timestamp і shouldChangeLoad маршрутизатора R1 після визначення падіння пропускної здатності

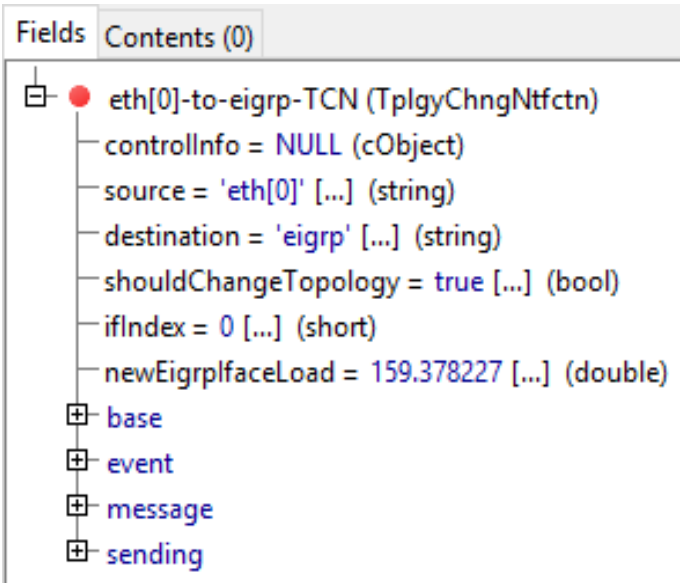

Рис. 3.9. TCN-повідомлення, отримане модулем EIGRP маршрутизатора R1 після визначення падіння пропускної здатності на інтерфейс eth[0]

```
** Event #365760  t=50.20320884  EigrpNet.R1.eigrp.eigrpIpv4Pdm (EigrpIpv4Pdm, id=105)  on eth[0]-to-eigrp-TCN (inet::TplgyChngNtfctn, id=538195)
INFO (EigrpIpv4Pdm)EigrpNet.R1.eigrp.eigrpIpv4Pdm: Received TCN: (inet::TplgyChngNtfctn)eth[0]-to-eigrp-TCN
INFO (EigrpIpv4Pdm)EigrpNet.R1.eigrp.eigrpIpv4Pdm: 1 10.0.0.0
INFO (EigrpIpv4Pdm)EigrpNet.R1.eigrp.eigrpIpv4Pdm: 2 10.0.0.0
INFO (EigrpIpv4Pdm)EigrpNet.R1.eigrp.eigrpIpv4Pdm: DUAL: received Update for route 10.0.0.0 via <unspec> (32426/0)
INFO (EigrpIpv4Pdm)EigrpNet.R1.eigrp.eigrpIpv4Pdm: EIGRP: Search feasible successor for route 10.0.0.0, FD is 28260
INFO (EigrpIpv4Pdm)EigrpNet.R1.eigrp.eigrpIpv4Pdm:     Next hop <unspec> (32426/0) satisfies FC
INFO (EigrpIpv4Pdm)EigrpNet.R1.eigrp.eigrpIpv4Pdm:     FS found, dmin is 32426
INFO (EigrpIpv4Pdm)EigrpNet.R1.eigrp.eigrpIpv4Pdm: DUAL: transit from oij=1 (passive) to oij=1 (passive) by transition 2
INFO (EigrpIpv4Pdm)EigrpNet.R1.eigrp.eigrpIpv4Pdm: EIGRP: Search successor for route 10.0.0.0, FD is 28260
INFO (EigrpIpv4Pdm)EigrpNet.R1.eigrp.eigrpIpv4Pdm:      successor <unspec> (32426/0)
INFO (EigrpIpv4Pdm)EigrpNet.R1.eigrp.eigrpIpv4Pdm: DUAL: send Update message about 10.0.0.0 to all neighbors, metric changed
INFO (EigrpIpv4Pdm)EigrpNet.R1.eigrp.eigrpIpv4Pdm: 1 10.0.0.24
INFO (EigrpIpv4Pdm)EigrpNet.R1.eigrp.eigrpIpv4Pdm: 2 10.0.0.24
INFO (EigrpIpv4Pdm)EigrpNet.R1.eigrp.eigrpIpv4Pdm: DUAL: received Update for route 10.0.0.24 via 10.0.0.2 (37546/30820)
INFO (EigrpIpv4Pdm)EigrpNet.R1.eigrp.eigrpIpv4Pdm: EIGRP: Search feasible successor for route 10.0.0.24, FD is 33380
INFO (EigrpIpv4Pdm)EigrpNet.R1.eigrp.eigrpIpv4Pdm:     Next hop 10.0.0.2 (37546/30820) satisfies FC
INFO (EigrpIpv4Pdm)EigrpNet.R1.eigrp.eigrpIpv4Pdm:     FS found, dmin is 37546
INFO (EigrpIpv4Pdm)EigrpNet.R1.eigrp.eigrpIpv4Pdm: DUAL: transit from oij=1 (passive) to oij=1 (passive) by transition 2
```

Рис. 3.10. Отримання модулем eigrp маршрутизатора R1 TCN-повідомлення про падіння продуктивності на інтерфейсі eth[0], початок процесу оновлення маршрутної інформації та розсилання пакетів UPDATE сусідам

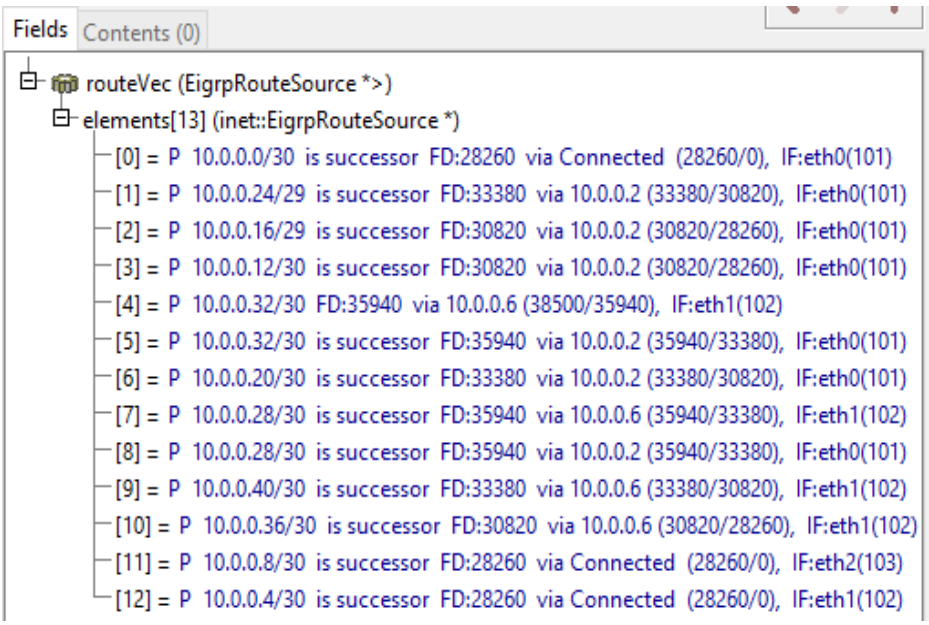

Рис. 3.11. Таблиця топології R1 зі старими метриками

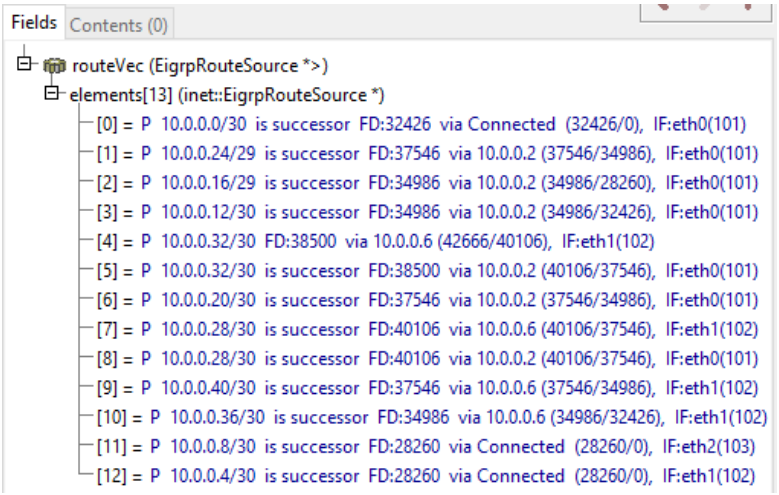

Рис. 3.12. Таблиця топології R1 з новими метриками після визначення навантаження інтерфейсу eth[0]

Після всього цього дані стали передаватися за новим шляхом. Графік зміни пропускної здатності на інтерфейсах eth[0] та eth[1] маршрутизатора R1 та на інтерфейсі eth[2] маршрутизатор R2, отриманий після закінчення моделювання, представлений на рис. 3.13.

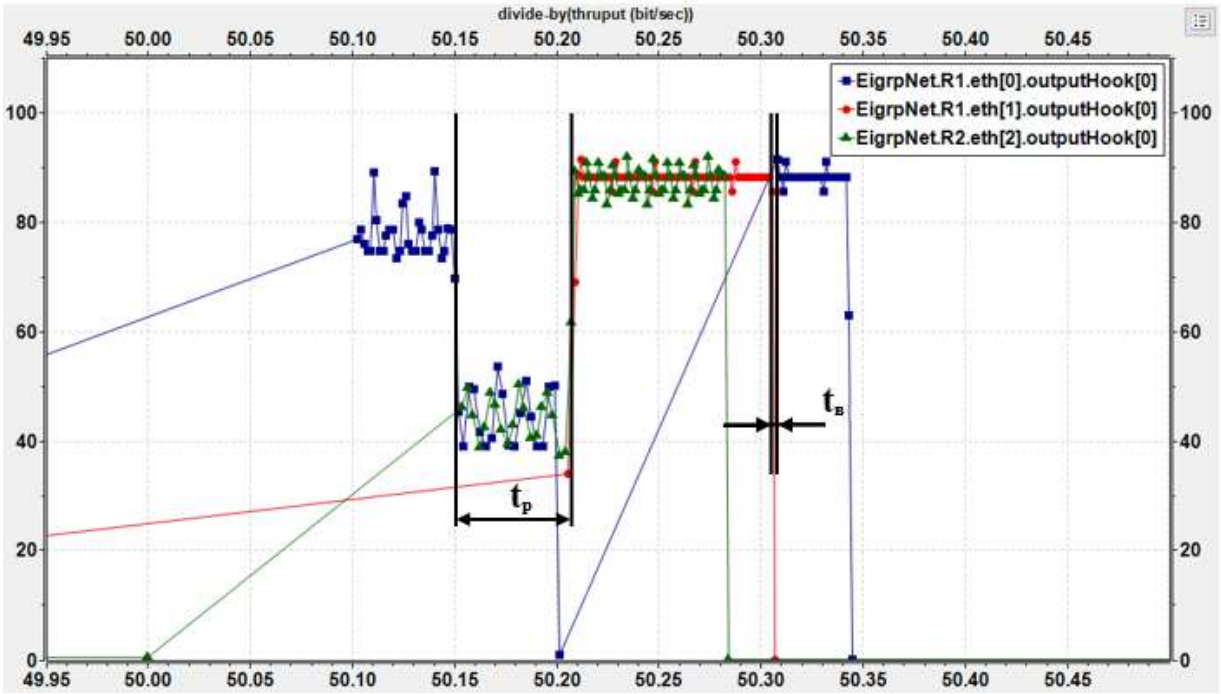

Рис. 3.13. Отриманий графік зміни пропускної здатності інтерфейсів маршрутизатора R1 та R2 під час моделювання мережі з оптимізованим протоколом EIGRP

На цьому графіку t реакції - час, витрачений на визначення навантаження інтерфейсу eth[0] маршрутизатора R1, а t відновлення – час, який потрібен для відновлення звичайної роботи EIGRP у тестовій мережі.

Після 50,15 с пропускна здатність на R1, R2 і R3 сильно впала, на визначення чого пішов час, що дорівнює t реакції. У цей же проміжок часу відбулося оновлення маршрутів на R1 з урахуванням нового навантаження на канал eth [0]. Далі дані від N3 до N4 і назад стали передаватися другим більш довгим, але вигіднішим шляхом, що проходить через інтерфейс eth[1] маршрутизатора R1, через R4, R5 і R6. При цьому дані користувачів від N1 до N2 надсилались та оброблялися весь цей час, тобто мережа продовжувала функціонувати незалежно від виникаючих перевантажень і оновлень маршрутної інформації у ній.

Потім приблизно 50,306 с через t відновлення, коли трафік користувачів від N1 до N2 передався повністю (через маршрутизатори R2 і R3), сталося відновлення мережі, та інформація, що залишилася від N3 до N4 продовжила передаватися по звичайному верхньому шляху через інтерфейс eth[0] вузла R1, через R2 і R3.

Кількість втрачених пакетів для інтерфейсу eth[0] маршрутизатора R1 представлений рис. 3.14, а інтерфейсу eth[1] цього ж маршрутизатора – на рис. 3.15.

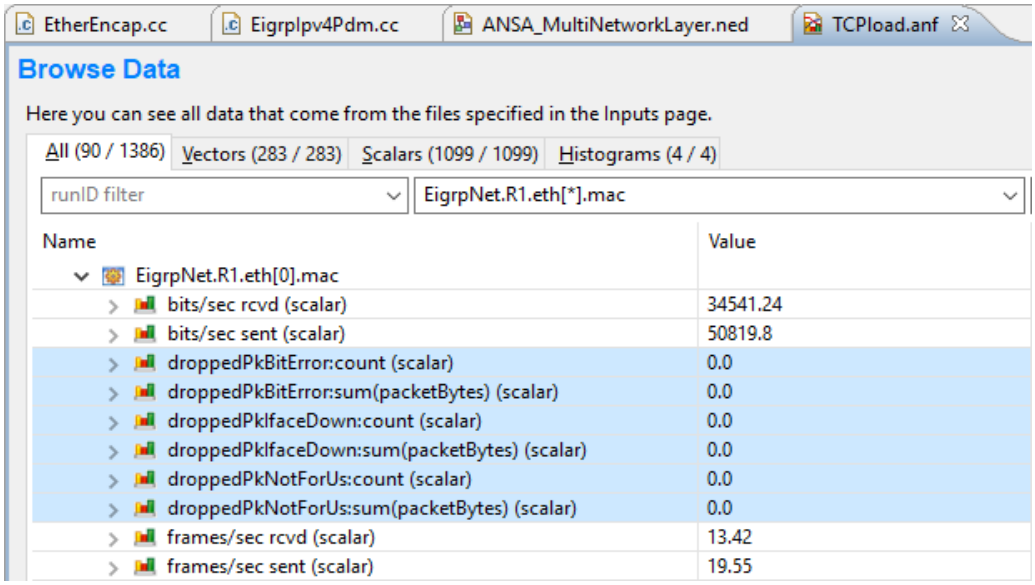

Рис. 3.14. Кількість втрачених пакетів на інтерфейсі eth[0] маршрутизатора R1 для модифікованого EIGRP

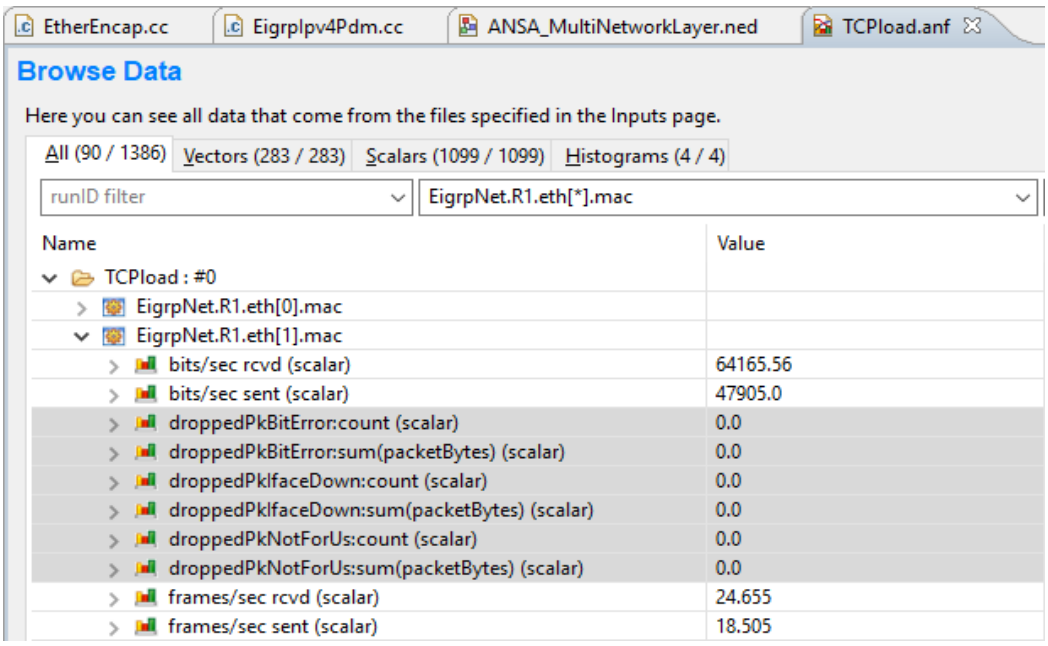

Рис. 3.15. Кількість втрачених пакетів на інтерфейсі eth[1] маршрутизатора R1 для модифікованого EIGRP

Втрат немає, що може говорити про надійність розробленої модифікації протоколу маршрутизації EIGRP. Таким чином, при проведенні моделювання була показано працездатність розробленої модифікації для оптимізації протоколу маршрутизації EIGRP.

Висновки до розділу 3.

Було формалізовано структуру мережі в середовищі Omnet. Проведено серію експериментів з метою визначення ефективності запропонованих змін в роботі протоколів маршрутизації при пікових навантаженнях, де в якості протоколу маршрутизації використовується протокол EIGRP.

У результаті метод був реалізований у бібліотеці ANSAINET для OMNeT ++, ефективність була показана під час моделювання.

РОЗДІЛ 4
ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1. Охорона праці

Усі дослідження методів оптимізації протоколів маршрутизації проводились з дотриманням правил та норм охорони праці і вимог техніки безпеки.

Маршрутизатори і інше активне мережеве обладнання яке забезпечує роботу комп'ютерних мереж використовує різного роду системи охолодження, а робота інших апаратних засобів є потенційним джерелом цілого ряду звуків, що містять як коливання, які можна почути, так і коливання ультразвукового діапазону. Цей шум справляє негативний вплив на функціональний стан користувачів. Вимірювання шуму на робочих місцях і санітарні норми рівня шуму визначені в ДСТУ 2867-94.

Так як не завжди є можливість виділити для розміщення серверів і комутаційного обладнання окреме приміщення їх встановлюють в спеціальних комутаційних шафах. Однак даний конструктив незважаючи на всі свої позитивні особливості не обмежує повністю шумове навантаження від роботи системи охолодження і інших механічних пристроїв.

Висококваліфікована розумова робота, що вимагає зосередженості, може проводитись у приміщеннях, де рівень шуму не перевищує 55 дБ. Сумарний вплив численних джерел шуму у приміщенні у результаті багаторазового відбиття звукових хвиль може значно перевищити енергію прямого звука від тих же джерел. Шум від окремих приладів не повинен перевищувати фоновий більше ніж на 5 дБ.

Для боротьби з шумом застосовують так звані «тихі» системи охолодження зокрема і водяне охолодження, або спеціальні великогабаритні вентилятори. Крім того комутаційні шафи частково поглинають шуми і вібрації. Тому в приміщенні де виконувалась наукова робота рівень шуму був в районі 40-45 дБ, тобто по рівню шумового навантаження приміщення відповідає вимогам охорони праці.

Також при роботі за ЕОМ необхідно особливу увагу звертати на правильне освітлення. Неправильне освітлення (пряма та відбита від екранів близькість,

вуалюючі відбиття, несприятливий розподіл яскравості в полі зору, невірна орієнтація робочого місця відносно світлових отворів) призводить до негативних фізіологічних впливів на користувачів ЕОМ. Погана якість символів, що представлені на екрані, також може викликати зоровий дискомфорт, бути стресовим фактором та ін.

Освітлення повинно відповідати нормальним рівням за ДБН В.2.5-28:2018 Природне і штучне освітлення.

Вимоги до освітлення для візуального сприймання користувачами інформації з двох різних носіїв (з екрана ЕОМ та паперового носія) різні. Надто низький рівень освітленості погіршує сприймання інформації при читанні документів, а надто високий призводить до зменшення контрасту зображення знаків на екрані. Відношення яскравості екрана ЕОМ до яскравості оточуючих його поверхонь не перевищує у робочій зоні 3:1.

Наближено можна вважати, що при 10%-ному зменшенні освітленості працездатність знижується на 1%. Коли за характером роботи вимагається комбінація цих двох носіїв інформації, освітленість можна варіювати від 300 до 700 лк, причому чим рідшою є зміна полів зору в процесі роботи (з екрана на документ та навпаки), тим вищим може бути рівень освітленості. 300-500 лк — оптимальна освітленість робочих приміщень для роботи з ЕОМ. Стрибки яскравості при зміні полів зору мають бути мінімальними, тобто інтенсивність освітлення поверхні, де знаходяться рукописи та документи, не повинна перевищувати яскравості екрана дисплея.

Приміщення в якому виконувалась кваліфікаційна робота забезпечене природнім і штучним освітленням. При роботі за ЕОМ обрано місце, щоб в поле зору не потрапляли вікна або освітлювальні прилади. Крім того шкідливо коли вікна знаходяться за спиною оскільки на моніторі з’являється відбиття світла. Завдяки наявним жалюзі можна регулювати світловий потік і захистити робоче місце від попадання прямих сонячних променів. Адже вікна приміщення орієнтовані на південний схід.

Штучне освітлення у приміщені реалізовано у вигляді комбінованої системи освітлення з використанням люмінесцентних джерел світла у світильниках загального освітлення, які слід розташовані над робочими поверхнями у рівномірно-прямокутному порядку. Для запобігання засвітленню екранів ЕОМ прямими світловими потоками лінії світильників розташовані з достатнім бічним зміщенням відносно робочих місць, а також паралельно до вікон.

На робочому місці забезпечена рівномірна освітленість за допомогою переважно відбитого або розсіяного світлорозподілу. Світлових відблисків з клавіатури, екрана та від інших частин ЕОМ у напрямку очей користувача немає. Дискомфорт від відбиття світла знижується при збільшенні яскравості екрана та зниженні рівня навколишнього освітлення.

Щоб уникнути пульсацій освітленості використовують якісні LED лампи.

Інформація, яку одержує користувач, генерується на екрані, а комфортність її сприймання залежить від чіткості символів. На робочих місцях користувачів використовуються сучасні ноутбуки та ПК обладнані рідкокристалічними моніторами. Завдяки використанню IPS матриць з частотою оновлення не менше 60 Гц очі практично не втомлюються. Матове покриття екрану виключає відблиск сторонніх джерел світла. Роздільна здатність моніторів що використовуються на робочих місцях 1920×1080 точок що забезпечує високу чіткість зображення. Ще одною важливою перевагою даних моніторів відсутнє опромінення користувача.

Отже в даному підрозділі розглянуто вплив середовища на працездатність та здоров’я користувачів комп’ютерів. Як висновок можна сказати, що робоче місце яке використовувалось для написання даного наукового дослідження відповідає вимогам з охорони праці.

Однак необхідно не забувати що надмірна робота з ПК може привезти до порушення роботи організму користувача. Тому необхідно дотримуватись вимог щодо планування робочого часу за ЕОМ і робити перерви. Це дозволить підвищити продуктивність роботи і зменшить втомленість організму.

4.2. Підвищення стійкості роботи об'єктів господарської діяльності у воєнний час

Під стійкістю об’єкту роботи об'єктів господарської діяльності країни в цілому розуміється його здатність в умовах воєнного часу забезпечувати всі галузі необхідною продукцією. Здатність об’єкта народного господарства випускати продукцію залежить від захисту і нормального функціонування чотирьох основних елементів сучасного виробництва, якими є:

– виробничий персонал (робітники та службовці);

– будинки і споруди з технологічним устаткуванням;

– система постачання енергією, водою, паливом, устаткуванням і ремонтною базою;

– система виробничих і кооперативних зв’язків з іншими об’єктами.

Тому стійкість роботи об’єктів і галузі народного господарства в цілому в умовах воєнного стану визначається наступними факторами:

– надійністю захисту робітників та службовців від усіх вражаючих факторів зброї масового ураження;

– здатністю інженерно-технічного комплексу (ІТК) об’єкта протистояти вражаючим факторам ядерного вибуху;

– надійністю системи постачання об’єкта всім необхідним для виробництва продукції (сировиною, паливом, що комплектують виробами, електроенергією, водою, газом тощо.);

– захищеності об’єкта від вторинних вражаючих факторів (пожеж, вибухів, затоплень, зараження місцевості отруйними і сильнодіючими отруйними речовинами);

– стійкістю і безперервністю керування виробництвом і цивільною обороною;

– підготовленість об’єкта до проведення рятувальних та інших невідкладних робіт і робіт з відновленням порушеного виробництва.

Перераховані фактори визначають собою й основні, загальні для всіх об’єктів господарювання, шляхи підвищення стійкості роботи в умовах військового стану, а саме:

– забезпечення надійного захисту робітників та службовців від вражаючих факторів зброї масового ураження;

– захист основних виробничих фондів від вражаючих факторів, у тому числі й від вторинних;

– підвищення надійності й оперативності керування виробництвом;

– забезпечення стійкості постачання всім необхідним для випуску запланованої на час надзвичайних ситуацій продукцією;

– підготовка до відновлення порушеного виробництва.

Захист робітників та службовців в умовах НС воєнного часу. Це найголовніша задача по підвищенню стійкості роботи об’єкта господарювання. Робітники й службовці – головна продуктивна сила і тому стійкість економіки визначається, насамперед, здатністю захистити і зберегти цю силу.

Військові конфлікти супроводжуються руйнуванням будинків, споруджень і знищенням основної продуктивної сили – працюючого населення. Тому серед усіх задач по підвищенню стійкості роботи об’єктів народного господарства основною є задача завчасного вживання заходів по забезпеченню захисту робітників та службовців і членів їхніх родин. Захист робітників та службовців від зброї масової поразки в сучасних умовах здійснюється трьома основними способами:

– укриття людей у захисних спорудженнях (сховищах, протирадіаційних укриттях);

– проведення евакуації робітників, службовців і членів їхніх родин;

– використання засобів індивідуального захисту, а також проведенням заходів щодо протирадіаційного, протихімічного і протибактеріологічного захисту з урахуванням конкретних обставин.

Варто також підкреслити, що найважливішою умовою успішного вирішення задачі захисту людей є навчання їх правилам дії по сигналах оповіщення цивільного

захисту, застосуванню способів і засобів захисту, наданню самодопомоги і взаємодопомоги, діям у складі формувань ЦЗ.

Захист засобів виробництва. полягає в підвищенні фізичної опірності будинків, споруджень і конструкцій об’єкта до впливу вражаючих факторів ядерного вибуху, захисту технологічного і верстатного устаткування, засобів зв’язку й інших засобів, що складають матеріальну основу виробничого процесу.

Основу діяльності керівника виробництва – начальника ЦЗ, а також його штабу складає якісне та професійне керування підлеглими йому структурами в організації їхньої дії і напрямку зусиль на своєчасне й успішне виконання виробничих завдань. Тому, забезпечення надійності й оперативності керування є важливою ланкою в підвищенні стійкості роботи об’єкта, в умовах швидко мінливої обстановки воєнного часу.

Забезпечення стійкого постачання підприємств. Для виробництва продукції необхідні: електроенергія, вода, паливо, сировина, матеріали й інші матеріально-технічні засоби. Забезпечення підприємств цими ресурсами багато в чому визначає можливість нормального їхнього функціонування в умовах воєнного часу. Це досягається проведенням таких заходів, що сприяють підвищенню не ураженості комунально-енергетичних мереж, транспортних комунікацій і джерел постачання, надійному захисту необхідних запасів палива, сировини, напівфабрикатів, що комплектують, виробів тощо.

Можливості вражаючою дії сучасних видів зброї такі, що забезпечити абсолютний захист від нього об’єктів і споруд практично неможливо. Вони можуть одержати той чи інший ступінь руйнування. У цих умовах задача зводиться до того, щоб у випадку слабких і середніх руйнувань на об’єкті відбудувати об’єкт і відновити випуск необхідної продукції в мінімальний термін.

Таким чином в даному розділі були розглянуті особливості підвищення стійкості об’єктів господарської діяльності в умовах воєнного стану.

ВИСНОВКИ

У результаті виконання роботи отримані наступні результати:

1) розглянуто найбільш використовувані протоколи маршрутизації мереж, OSPF та EIGRP;

2) виявлені недоліки OSPF і EIGRP, що не дозволяють ефективно використовувати ці протоколи в мережах з високим рівнем трафіку;

3) визначені критерії ефективності роботи мережі;

4) запропоновано модифікацію протоколу EIGRP для перебудови таблиць маршрутизації з урахуванням динамічно змінюваного навантаження на канали зв’язку;

5) проведено експерименти з модифікованим EIGRP на моделі мережі, написаної в OMNeT++,

6) було показано працездатність доопрацьованої версії протоколу. Ефективність запропонованого методу підтверджено з використанням графіків що були отримані в результаті моделювання

Згодом запропоновані напрацювання можуть бути використані при складання і реалізації оптимального алгоритму необхідності перемикання на нові маршрути мереж довільною топології для використання не лише серед моделювання, а й у реальних мережах передачі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1.Wetherall David J.; Tanenbaum Andrew S. Computer networks. Pearson Education, 2013.

2.Enhanced Interior Gateway Routing Protocol // Cisco Systems, Inc. URL: https://www.cisco.com/c/en/us/support/docs/ip/en- hanced-interior-gateway-routing-protocol-eigrp/16406-eigrp-toc.html (дата звернення: 11.12.2023).

3.Pepelnjak I. EIGRP load and reliability metrics / I. Pepelnjak // ipSpace.net: Internetworking perspectives by Ivan Pepelnjak. URL: http://blog.ipspace.net/2009/06/eigrp-load-and-reliability-metrics.html (дата звернення: 11.12.2023).

4.RFC 7868, Cisco's Enhanced Interior Gateway Routing Protocol (EIGRP) // The Internet Engineering Task Force (IETF). URL:: https://datatracker.ietf.org/doc/rfc7868/ (дата звернення: 11.12.2023).

5.RFC 2328, OSPF Version 2 // The Internet Engineer- ing Task Force (IETF). URL: https://datatracker.ietf.org/doc/rfc2328/ (дата звернення: 11.12.2023).

6.RFC 2328 URL: https://www.rfc-editor.org/rfc/rfc2328.html (дата звернення: 11.12.2023).

7.Manzoor A., Hussain M., Mehrban S. Performance analysis and route optimization: redistribution between EIGRP, OSPF & BGP routing protocols. Computer Standards & Interfaces, 2020, 68: 103391.

8.Introduction to EIGRP // Cisco Systems, Inc. URL: https://www.cisco.com/c/en/us/support/docs/ip/enhanced-interior-gateway- routing-protocol-eigrp/13669-1.html (дата звернення: 11.12.2023).

9.An Introduction to IGRP // Cisco Systems, Inc. URL: https://www.cisco.com/c/en/us/support/docs/ip/interior-gateway-rout- ing-protocol-igrp/ 26825-5.html (дата звернення: 11.12.2023).

10.Dumitrache C. G., et al. Comparative study of RIP, OSPF and EIGRP protocols using Cisco Packet Tracer. In: 2017 5th International Symposium on Electrical and Electronics Engineering (ISEEE). IEEE, 2017. p. 1-6.

11.Enhanced Interior Gateway Routing Protocol (EIGRP) Informational RFC Frequently Asked Questions // Cisco Systems, Inc. URL: https://www.cisco.com/c/en/us/products/collateral/ios-nx-os-software/en- hanced-interior-gateway-routing-protocol-eigrp/qa_C67-726299.html (дата звернення: 11.12.2023).

12.Burke A. Why Is Cisco Bothering with «Open» EIGRP? // Packet Pushers Interactive, LLC. URL: http://pack- etpushers.net/why-is-cisco-bothering-with-open-eigrp/ (дата звернення: 11.12.2023).

13.Ватаманеску С. В., Луценко А. В. Про застосування графів у комп'ютерних інформаційних технологіях. Прикладні інформаційні технології, 2023, 28-30.

14.Snihurov A. Improvement of EIGRP Protocol Routing Algorithm with the Consideration of Information Security Risk Parameters / A. Snihurov, V. Chakrian //. URL: http://openarchive.nure.ua/bitstream/document/2243/1/SJET38707-714.pdf (дата звернення: 11.12.2023).

15.Improvement of Performance of EIGRP Network by Using a Supervisory Controller with Smart Congestion Avoidance Algorithm //ResearchGateGmbH.URL: https://www.re- searchgate.net/publication/306925828_Improvement_of_perfor- mance_ of_EIGRP_ network_by_using_a_supervisory_controller_with_smart_conges- tion_ avoidance_ algorithm (дата звернення: 11.12.2023).

16.Мартовицький В., Акіменко Б. Порівняння двох алгоритмів пошуку найкоротших шляхів між вузлами комп'ютерної мережі. 2019.

17.Кульчинський І. Аналіз роботи протоколів динамічної маршрутизації. Збірник тез V Всеукраїнської студентської науково-технічної конференції „Природничі та гуманітарні науки. Актуальні питання ", 2012, 1: 67-67..

18.Шевченко Н. Аналіз протоколів маршрутизації у сучасних комп'ютерних мережах для швидкості поширення маршрутної інформації і обчислення оптимальних шляхів. MS thesis. 2021.

19.Бігуняк А., Жаровський Р. Особливості протоколів маршрутизації в комп'ютерних мережах. Матеріали II науково-технічної конференції „Інформаційні моделі, системи та технології ", 2012, 40-40.

20.Daniluk K. Energy-Efficient Protocol in OMNeT++ Simulation Environ- ment / K. Daniluk // ITHEA International Scientific Journals. URL: http://foibg.com/ibs_isc/ibs-27/ibs-27-p24.pdf (дата звернення: 11.12.2023).

21.Saenko I., Kotenko I. Design of Virtual Local Area Network Scheme Based on Genetic Optimization and Visual Analysis. J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl., 2014, 5.4: 86-102.

22.OMNeT++ // OMNeT++ Discrete Event Simulator. URL: https://omnetpp.org (дата звернення: 11.12.2023).

23.INET Framework // INET Framework. URL: https://inet.omnetpp.org/ (дата звернення: 11.12.2023).

24.ANSAINET // ANSA by Brno University of Technology. URL: https://ansa.omnetpp.org/ (дата звернення: 11.12.2023).

25.Буранич І., Жаровський Р. Протокол EIGRP. Збірник тез VIII всеукраїнської студентської науково-технічної конференції „Природничі та гуманітарні науки. Актуальні питання “, 2015, 1: 69-69.

26.Wallace K. CCNP Routing and Switching ROUTE 300-101 Official Cert Guide. – Cisco Press, 2014.

27.“How Does Unequal Cost Path Load Balancing (Variance) Work in IGRP and EIGRP?” URL: http://www.cisco.com/c/en/us/support/docs/ip/enhanced-interior-gateway-routing-protocol-eigrp/13677-19.html (дата звернення: 11.12.2023).

28.“Cisco Express Forwarding Overview” URL: http://www.cisco.com/c/en/us/support/docs/ip/enhanced-interior-gateway-routing-protocol-eigrp /13677-19.html (дата звернення: 11.12.2023).

29.Adomnicăi C. Routing protocols behaviour under bandwidth limitation //Proceedings of International Conference on Information and Computer Networks. – 2012. – Т. 27. – С. 52-57.

30.Anvitha P., Shashank S., Shridhar D. “CEF Polarization” –URL: http://www.cisco.com/c/en/us/support/docs/ip/express-forwarding-cef/116376-technote-cef-00.html (дата звернення: 11.12.2023).

31. Чайковський А. В., Жаровський Р. О., Лещишин Ю. З. "Конспект лекцій з дисципліни «Дослідження і проєктування комп’ютерних систем та мереж» для студентів спеціальності 123–Комп’ютерна інженерія." 2021. 343с.

32. Жаровський Руслан Олегович. "Конспект лекцій з дисципліни Захист інформації у комп’ютерних системах." 2019 268 с.

33. Лупенко С.А., Луцик Н.С., Луцків А.М., Осухівська Г.М., Тиш Є.В. Методичні рекомендації до виконання кваліфікаційної роботи магістра для студентів спеціальності 123 «Комп’ютерна інженерія» другого (магістерського) рівня вищої освіти усіх форм навчання. Тернопіль. 2021. 34 с.

34. Озарків Т., Жаровський Р. Метод оптимізації EIGRP протоколу для підвищення продуктивності передачі даних в комп’ютерних мережах. Матеріали XI науково-технічної конференції Тернопільського національного технічного університету імені Івана Пулюя «Інформаційні моделі системи та технології» (13-14 грудня 2023 року). Тернопіль: ТНТУ. 2023. С.167.

35. Озарків Т., Жаровський Р. Оптимізація роботи протоколу EIGRP в умовах великих мереж зі складною топологією. Матеріали XII Міжнародна науково-технічна конференція молодих учених та студентів «Актуальні задачі сучасних технологій» (6-7 грудня 2023 року). Тернопіль: ТНТУ. 2023. С. 442.

Додаток А.

Тези конференцій

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Тернопільський національний технічний університет імені Івана Пулюя (Україна)
Університет імені П'єра і Марії Кюрі (Франція)
Марібурський університет (Словенія)
Технічний університет у Кошице (Словаччина)
Вільнюський технічний університет ім. Гедиміната (Литва)
Міжнародний університет цивільної авіації (Марокко)
Наукове товариство ім. Т.Шевченка

АКТУАЛЬНІ ЗАДАЧІ СУЧАСНИХ ТЕХНОЛОГІЙ

Збірник
тез доповідей

XII Міжнародної науково-практичної конференції молодих учених та студентів
6-7 грудня 2023 року

УКРАЇНА
ТЕРНОПІЛЬ – 2023

УДК 004.45

Т. А. Озарків; Р. О. Жаровський , к.т.н.
(Тернопільський національний технічний університет імені Івана Пулюя, Україна)

ОПТИМІЗАЦІЯ РОБОТИ ПРОТОКОЛУ EIGRP В УМОВАХ ВЕЛИКИХ МЕРЕЖ ЗІ СКЛАДНОЮ ТОПОЛОГІЄЮ

T. A. Ozarkiv; R. O. Zharovskyi, Ph.D.

OPTIMIZATION OF THE EIGRP PROTOCOL IN LARGE NETWORKS WITH COMPLEX TOPOLOGY

У великих мережах з складною топологією та багатьма альтернативними маршрутами використання протоколів маршрутизації автоматизує створення таблиць маршрутизації та дозволяє знаходити нові маршрути при змінах у мережі, таких як відмови або появи нових ліній зв'язку та маршрутизаторів.

Протоколи маршрутизації використовуються для пошуку та фіксації маршрутів передачі даних через складову мережу TCP/IP. Більшість таких протоколів формують таблиці маршрутизації та відмічаються адаптивною (динамічною) або статичною маршрутизацією [1].

Сучасні протоколи маршрутизації в IP-мережах відносяться до адаптивних розподілених протоколів. Популярними дистанційно-векторними протоколом є RIP, а серед інших протоколів цієї групи варто відзначити удосконалений протокол EIGRP, розроблений компанією Cisco як наступника IGRP.

EIGRP придатний для різних топологій та середовищ. У добре спроектованих мережах EIGRP добре масштабується та має невеликий час узгодження при мінімальному мережевому трафіку. Для обчислення найкоротшого шляху використовується алгоритм дифузного оновлення (DUAL).

До переваг алгоритму роботи мережі EIGRP можна віднести:

- низьке використання мережевих ресурсів у режимі нормальної експлуатації: тільки пакети HELLO передаються за умов стабільної мережі;
- при виникненні змін через мережу передаються лише зміни, що відбулися в маршрутній таблиці, а не вся таблиця повністю; це дозволяє знизити навантаження на мережу, що створюється протоколом маршрутизації;
- малий час конвергенції (або збіжності) у разі зміни у топології мережі (в окремих випадках збіжність забезпечується майже миттєво);
- можливість використання до 5-ти компонентів при розрахунку метрики маршрутизації.

У порівнянні з іншими протоколами, EIGRP має вбудований механізм обліку навантаження на лінію, але він обмежений у гнучкості. Протокол може бути налаштований за допомогою різних параметрів, але перерахунок маршрутів відбувається тільки при зміні топології мережі. Для протоколу OSPF, який часто використовується в магістральних IP-мережах, існує багато способів оптимізації [2] для розподілу навантаження на мережу. Навпаки, протокол EIGRP ще вимагає досліджень для оптимізації його роботи в умовах великого рівня трафіку.

Література

1. Katumela, Noelia; Kattan, Dimitrios A. A Comparative Analysis of OSPF and EIGRP Routing Protocol Evaluation. Journal of Transactions in Systems Engineering, 2023, 1.2: 73-103.

2. Tamgno, James Kouawa, et al. Optimization Of Eigrp Dynamic Routing Protocol Based On Artificial Intelligence Algorithm. In: 2022 24th International Conference on Advanced Communication Technology (ICACT). IEEE, 2022. p. 370-379.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ
УНІВЕРСИТЕТ ІМЕНІ ІВАНА ПУЛЮЯ

МАТЕРІАЛИ

XI НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ

«ІНФОРМАЦІЙНІ МОДЕЛІ, СИСТЕМИ ТА ТЕХНОЛОГІЇ»

13-14 грудня 2023 року

ТЕРНОПІЛЬ
2023

УДК 004.45

Т.А. Озарків; Р.О. Жаровський , к.т.н.
(Тернопільський національний технічний університет імені Івана Пулюя, Україна)

МЕТОД ОПТИМІЗАЦІЇ EIGRP ПРОТОКОЛУ ДЛЯ ПІДВИЩЕННЯ ПРОДУКТИВНОСТІ ПЕРЕДАЧІ ДАНИХ В КОМП'ЮТЕРНИХ МЕРЕЖАХ

T. A. Ozarkiv; R.O. Zharovskyi, Ph.D.

THE METHOD OF OPTIMIZING THE EIGRP PROTOCOL TO INCREASE THE PRODUCTIVITY OF DATA TRANSMISSION IN COMPUTER NETWORKS

Протоколи маршрутизації EIGRP та OSPF не адаптовані до різкого збільшення навантаження на канали на тривалий час. Вирішення (хоча б часткове) цієї проблеми має теоретичне та практичне значення для більш ефективного використання ресурсів мережі, та дозволить зменшити навантаження на окремі вузли мережі та збільшити їх стійкість до відмов, наприклад за рахунок зміни маршруту, по якому передається інформація, на альтернативний, менш завантажений, і водночас надасть додаткові можливості для подальшого аналізу та покращення роботи протоколів.

Багато дослідників пропонують свої способи пом'якшення наслідків виникнення навантажень у мережах. Такі як алгоритм Беллмана Форда, який для побудови резервних шляхів використовує дані про ребра.

Є пропозиції підвищити продуктивність традиційного EIGRP шляхом додавання деяких SDN-функцій. Також існує ряд робіт з методами удосконалення алгоритму Дейкстри, що дозволяє одночасно з вирішенням задачі пошуку найкоротших шляхів сформувати резервні шляхи до вузлів мережі. Підхід цієї роботи було взято за основу під час модифікації алгоритму Беллмана-Форда

У ході аналізу розробок у цій галузі було виявлено, що для EIGRP дослідження ведуться не так активно, як, наприклад, для OSPF.

Пропонується наступна модифікація роботи протоколу EIGRP. Після повної ініціалізації роботи EIGRP у мережі:

- на кожному інтерфейсі роутера, який використовує EIGRP як протокол маршрутизації, проводити розрахунок поточної пропускної спроможності (завантаження) за певний інтервал або за певну кількість пакетів у бітах за секунду;
- якщо завантаження на одному з інтерфейсів, наприклад, протягом 1 секунди менше певного порогу, то необхідно змінити завантаження EIGRP інтерфейсу на величину, що відповідає падінню пропускної здатності;
- при зміні стану каналу викликати перерахунок маршрутів, до яких він входить, після чого сповістити про це маршрутизатори-сусіди. У цьому випадку (тобто при зміні топології) на кожному з маршрутизаторів, що залишилися, також повинен відбутися перерахунок необхідних маршрутів (при необхідності).

Максимальний поріг поточної пропускної спроможності, після перебільшення якого має бути змінено завантаження інтерфейсу, інші важливі для роботи алгоритму значення та робота елементів, що відповідають за зміну завантаження та перебудову маршрутів, будуть розглянуті та визначені при практичному дослідженні запропонованого методу.

В результаті виконання роботи було розроблено метод підвищення продуктивності мережі при великих навантаженнях, де протокол EIGRP використовується як основний протокол маршрутизації.

Були проведені експерименти з модифікованим EIGRP на моделі мережі та була показана працездатність доопрацьованої версії протоколу.

Додаток Б.

Визначення пропускної здатності

```
// measure throughput only on an EIGRP router
    if (strcmp(ethIface->getParentModule()->getModuleType()-
            >getName(), "ANSA_EIGRPRouter") == 0)
{
    bool isThresholdPassed = ethIface->getParentModule()
                                  ->par("isThresholdPassed");
    simtime_t now = simTime();
    unsigned long bits = msg->getBitLength();

    // packet should be counted to new interval if (packetsPerIntvl
        >= batchSize
        || now - intvlStartTime >= maxInterval)
    {
        simtime_t duration = now - intvlStartTime;
        // record measurements
        double bitpersec = bitsPerIntvl / duration.dbl();
        //double pkpersec = packetsPerIntvl / duration.dbl();
        EV << "bitpersec = " << bitpersec
           << " at t = " << simTime() << "\n";
        // threshold passed, set necessary values if (threshold &&
            bitpersec <= threshold
            && !isThresholdPassed)
        {
            timestamp = simTime(); ethIface->getParentModule()
                        ->par("isThresholdPassed").setBoolValue(true);
                ethIface->getParentModule()
                    ->par("ifIndex").setLongValue(ethIface
                        ->getIndex()); shouldChangeLoad = true;
            // determining a maximum channel data rate
            cGate *ethIfaceOutGate = ethIface->getParentModule()
                                              ->gate("ethg$o",
                                                ethIface
                                          ->getIndex());
            ASSERT(ethIfaceOutGate);
                cDatarateChannel *channel =
                    check_and_cast<cDatarateChannel*>(
                    ethIfaceOutGate->getChannel());
            double channelDatarate = channel->getDatarate();
            // new EIGRP interface load normalization (feature
            // scaling) from <0-channel data rate> to <1-255>
                newEigrpIfaceLoad = (bitpersec - 0.) /
                                (channelDatarate - 0.) * (255. - 1.) +
                                    1.;
        }
        oldThreshold = bitpersec;
        // 30 %
        threshold = 0.70 * bitpersec;
        // restart counters intvlStartTime = now;
```

```
        packetsPerIntvl = bitsPerIntvl = 0;
    }
    packetsPerIntvl++; bitsPerIntvl += bits;
    // send TCN (Topology Change Notification)
    // (to ANSA_MultiNetworkLayerLowerMultiplexer) if
        (isThresholdPassed && shouldChangeLoad
        && simTime() >= timestamp)
    {
        // (EigrpProcessDS) eigrp
        cModule *eigrp = ethIface->getParentModule()
                                ->getSubmodule("eigrp");
        char msgname[20];
            sprintf(msgname, "%s-to-%s-TCN", ethIface->getFullName(),
                    eigrp->getName());
        // set TCN fields
        TplgyChngNtfctn *tcnMsg = new TplgyChngNtfctn(msgname);
            tcnMsg->setSource(ethIface->getFullName());
        tcnMsg->setDestination(eigrp->getName());
        tcnMsg->setShouldChangeTopology(shouldChangeLoad); tcnMsg-
            >setIfIndex(ethIface->getIndex());
        tcnMsg->setNewEigrpIfaceLoad(newEigrpIfaceLoad);
        cGate *outGate = gate("upperLayerOut"); send(tcnMsg,
            outGate);
        shouldChangeLoad = false;
    }
}
```

Додаток В.

Конфігурація топології моделі тестової EIGRP-мережі (EigrpNet.ned)

```
package eigrp_for_masters_thesis;
import inet.common.misc.ThruputMeteringChannel; import
    inet.linklayer.ethernet.EtherHub;
import inet.networklayer.configurator.ipv4.IPv4NetworkConfigurator;
    import inet.common.scenario.ScenarioManager;
import inet.common.lifecycle.LifecycleController; import
    ansa.node.ANSA_EIGRPRouter;
import ansa.node.ANSA_Host;
channel C extends ThruputMeteringChannel
{
    delay = 0.1us; datarate = 100Mbps;
    thruputDisplayFormat = "# N";
}
module EigrpLan
{
    parameters:
        int h; // number of hosts on the hub
            @display("i=device/lan");
    gates:
            inout ethg []; submodules:
            hub: EtherHub { parameters:
                @display("is=s");
        }
            host[h]: ANSA_Host { parameters:
                @display("is=s");
        }
    connections:
            for i=0..sizeof(ethg)-1 { hub.ethg++ <--> ethg [i];
        }
        for i=0..h-1 {
            hub.ethg++ <--> C <--> host[i].ethg++;
        }
}
network EigrpNet
{
    @display("bgb = 385,240; bgl = 2"); submodules:
            R1: ANSA_EIGRPRouter { parameters:
                    @display("p = 71,137"); gates:
                ethg [3];
        }
            R2: ANSA_EIGRPRouter { parameters:
                    @display("p=125,89"); gates:
                ethg [3];
        }
            R3: ANSA_EIGRPRouter { parameters:
                    @display("p = 233,89"); gates:
                ethg [3];
        }
```

```
            R7: ANSA_EIGRPRouter { parameters:
                    @display("p = 301,137"); gates:
                ethg [3];
        }
            R4: ANSA_EIGRPRouter { parameters:
                    @display("p = 125,197"); gates:
                ethg [2];
        }
            R5: ANSA_EIGRPRouter { parameters:
                    @display("p = 179,197"); gates:
                ethg [2];
        }
            R6: ANSA_EIGRPRouter { parameters:
                    @display("p = 233,197"); gates:
                ethg [2];
        }
            N1: EigrpLan { parameters:
                h = 2;
                @display("p=124,22");
        }
            N2: EigrpLan { parameters:
                h = 2;
                @display("p = 232,22");
        }
            N3: EigrpLan { parameters:
                h = 1;
                @display("p = 22,136");
        }
            N4: EigrpLan { parameters:
                h = 1;
                @display("p=361,136");
        }
            configurator: IPv4NetworkConfigurator { parameters:
                config = xml("<config></config>"); assignAddresses =
                    false; assignDisjunctSubnetAddresses = false;
                    addStaticRoutes = false; addDefaultRoutes =
                    false; addSubnetRoutes = false; optimizeRoutes =
                    false; @display("p=44.615387,39.23077");
        }
    connections:
        R1.ethg[0] <--> C <--> R2.ethg[0];
        R2.ethg[1] <--> C <--> R3.ethg[0];
        R3.ethg[1] <--> C <--> R7.ethg[0];
        R1.ethg[1] <--> C <--> R4.ethg[0];
        R4.ethg[1] <--> C <--> R5.ethg[0];
        R5.ethg[1] <--> C <--> R6.ethg[0];
        R6.ethg[1] <--> C <--> R7.ethg[1];
        N1.ethg++ <--> C <--> R2.ethg [2];
        N2.ethg++ <--> C <--> R3.ethg[2];
        N3.ethg++ <--> C <--> R1.ethg [2];
        N4.ethg++ <--> C <--> R7.ethg[2];
}
```

Додаток Д.

Параметри виконання моделі тестової мережі EIGRP (omnetpp.ini)

```
[General]
network = EigrpNet #record-eventlog = true #debug-on-errors = true
#tkenv-plugin-path =../../../etc/plugins sim-time-limit =
    200s#200s#600s
**.enableIPv6 = false
**.enableCLNS = false
**.*.networkLayer.enableANSAConfig = true
**.R1.configData = xmldoc("config.xml",
                    "Devices/Router[@id='10.0.0.9']/")
**.R2.configData = xmldoc("config.xml",
                    "Devices/Router[@id='10.0.0.17']/")
**.R3.configData = xmldoc("config.xml",
                    "Devices/Router[@id='10.0.0.25']/")
**.R4.configData = xmldoc("config.xml",
                    "Devices/Router[@id='10.0.0.37']/")
**.R5.configData = xmldoc("config.xml",
                    "Devices/Router[@id='10.0.0.41']/")
**.R6.configData = xmldoc("config.xml",
                    "Devices/Router[@id='10.0.0.42']/")
**.R7.configData = xmldoc("config.xml",
                    "Devices/Router[@id='10.0.0.33']/")
**.N1.host[0].configData = xmldoc("config.xml",
                    "Devices/Host[@id='10.0.0.18']/")
**.N1.host[1].configData = xmldoc("config.xml",
                    "Devices/Host[@id='10.0.0.19']/")
**.N2.host[0].configData = xmldoc("config.xml",=
                    "Devices/Host[@id='10.0.0.26']/")
**.N2.host[1].configData = xmldoc("config.xml",
                    "Devices/Host[@id='10.0.0.27']/")
**.N3.host[0].configData = xmldoc("config.xml",
                    "Devices/Host[@id='10.0.0.10']/")
**.N4.host[0].configData = xmldoc("config.xml",
                    "Devices/Host[@id='10.0.0.34']/")
# hookType settings
**.eth[*].numOutputHooks = 1
    **.eth[*].outputHook[0].typename = "ThruputMeter" # Nop |
         ThruputMeter | OrdinalBasedDropper | OrdinalBasedDuplicator
[Config TCPload]
description = "Use of TCP apps to decrease network throughput" # tcp
    apps
#
# from N3.host[0] (R1) to N4.host[0] (R7) #
**.N3.host[0].numTcpApps = 1
**.N3.host[0].tcpApp[0].typename = "TCPSessionApp"
**.N3.host[0].tcpApp[0].sendBytes = 2000000 bytes
**.N3.host[0].tcpApp[0].active = true
**.N3.host[0].tcpApp[0].connectAddress = "N4.host[0]"
**.N3.host[0].tcpApp[0].connectPort = 10021
```

```
**.N3.host[0].tcpApp[0].tOpen = 50s
**.N3.host[0].tcpApp[0].tSend = 50.10s
**.N3.host[0].tcpApp[0].tClose = 50s
**.N4.host[0].numTcpApps = 1
**.N4.host[0].tcpApp[0].typename = "TCPEchoApp"
**.N4.host[0].tcpApp[0].localPort = 10021
#
# from N1.host[0] (R2) to N2.host[0] (R3) #
**.N1.host[0].numTcpApps = 1
**.N1.host[0].tcpApp[0].typename = "TCPSessionApp"
**.N1.host[0].tcpApp[0].sendBytes = 1000000 bytes
**.N1.host[0].tcpApp[0].active = true
**.N1.host[0].tcpApp[0].connectAddress = "N2.host[0]"
**.N1.host[0].tcpApp[0].connectPort = 10021
**.N1.host[0].tcpApp[0].tOpen = 50s
**.N1.host[0].tcpApp[0].tSend = 50.15s
**.N1.host[0].tcpApp[0].tClose = 50s
**.N2.host[0].numTcpApps = 1
**.N2.host[0].tcpApp[0].typename = "TCPEchoApp"
**.N2.host[0].tcpApp[0].localPort = 10021
```

Додаток Е.

Параметри пристроїв тестової EIGRP-мережі (config.xml)

```
<Devices>
     <!-- R1 -- >
     <Router id="10.0.0.9">
       <Interfaces>
           <Interface name="eth0">
               <IPAddress>10.0.0.1</IPAddress>
               <Mask>255.255.255.252</Mask>
               <EIGRP-IPv4 asNumber='1'>
               </EIGRP-IPv4> _
           </Interface>
           <Interface name="eth1">
               <IPAddress>10.0.0.5</IPAddress>
               <Mask>255.255.255.252</Mask>
               <EIGRP-IPv4 asNumber='1'>
               </EIGRP-IPv4> _
           </Interface>
           <Interface name="eth2">
               <IPAddress>10.0.0.9</IPAddress>
               <Mask>255.255.255.252</Mask>
               <EIGRP-IPv4 asNumber='1'>
               </EIGRP-IPv4> _
           </Interface>
       </Interfaces>

       <Routing>
          <EIGRP>
               <ProcessIPv4 asNumber="1">
                    <Networks>
                         <Network>
                              <IPAddress>10.0.0.1</IPAddress>
                              <Wildcard>0.0.0.3</Wildcard>
                         </Network>
                         <Network>
                              <IPAddress>10.0.0.5</IPAddress>
                              <Wildcard>0.0.0.3</Wildcard>
                         </Network>
                         <Network>
                              <IPAddress>10.0.0.9</IPAddress>
                              <Wildcard>0.0.0.3</Wildcard>
                         </Network>
                    </Networks>
               </ProcessIPv4>
          </EIGRP>
       </Routing>
       <Routing6>
       </Routing6>

    </Router>
```

```
<!-- R2 -- >
<Router id="10.0.0.17">
    <Interfaces>
        <Interface name="eth0">
            <IPAddress>10.0.0.2</IPAddress>
            <Mask>255.255.255.252</Mask>
            <EIGRP-IPv4 asNumber='1'>
            </EIGRP-IPv4> _
        </Interface>
        <Interface name="eth1">
            <IPAddress>10.0.0.13</IPAddress>
            <Mask>255.255.255.252</Mask>
            <EIGRP-IPv4 asNumber='1'>
            </EIGRP-IPv4> _
        </Interface>
        <Interface name="eth2">
            <IPAddress>10.0.0.17</IPAddress>
            <Mask>255.255.255.248</Mask>
            <EIGRP-IPv4 asNumber='1'>
            </EIGRP-IPv4> _
        </Interface>
    </Interfaces>

    <Routing>
       <EIGRP>
            <ProcessIPv4 asNumber="1">
                 <Networks>
                      <Network>
                           <IPAddress>10.0.0.2</IPAddress>
                           <Wildcard>0.0.0.3</Wildcard>
                      </Network>
                      <Network>
                           <IPAddress>10.0.0.13</IPAddress>
                           <Wildcard>0.0.0.3</Wildcard>
                      </Network>
                      <Network>
                           <IPAddress>10.0.0.17</IPAddress>
                           <Wildcard>0.0.0.7</Wildcard>
                      </Network>
                 </Networks>
            </ProcessIPv4>
       </EIGRP>
    </Routing>

    <Routing6>
    </Routing6>

</Router>

<!-- R3 -- >
<Router id="10.0.0.25">
    <Interfaces>
```

```
        <Interface name="eth0">
            <IPAddress>10.0.0.14</IPAddress>
            <Mask>255.255.255.252</Mask>
            <EIGRP-IPv4 asNumber='1'>
            </EIGRP-IPv4> _
        </Interface>
        <Interface name="eth1">
            <IPAddress>10.0.0.21</IPAddress>
            <Mask>255.255.255.252</Mask>
            <EIGRP-IPv4 asNumber='1'>
            </EIGRP-IPv4> _
        </Interface>
        <Interface name="eth2">
            <IPAddress>10.0.0.25</IPAddress>
            <Mask>255.255.255.248</Mask>
            <EIGRP-IPv4 asNumber='1'>
            </EIGRP-IPv4> _
        </Interface>
    </Interfaces>

    <Routing>
       <EIGRP>
            <ProcessIPv4 asNumber="1">
                 <Networks>
                      <Network>
                           <IPAddress>10.0.0.14</IPAddress>
                            <Wildcard>0.0.0.3</Wildcard>
                      </Network>
                      <Network>
                           <IPAddress>10.0.0.21</IPAddress>
                           <Wildcard>0.0.0.3</Wildcard>
                      </Network>
                      <Network>
                           <IPAddress>10.0.0.25</IPAddress>
                           <Wildcard>0.0.0.7</Wildcard>
                      </Network>
                 </Networks>
            </ProcessIPv4>
       </EIGRP>
    </Routing>
    <Routing6>
    </Routing6>

</Router>

<!-- R4 -- >
<Router id="10.0.0.37">
    <Interfaces>
        <Interface name="eth0">
            <IPAddress>10.0.0.6</IPAddress>
            <Mask>255.255.255.252</Mask>
            <EIGRP-IPv4 asNumber='1'>
            </EIGRP-IPv4> _
```

```
        </Interface>
        <Interface name="eth1">
            <IPAddress>10.0.0.37</IPAddress>
            <Mask>255.255.255.252</Mask>
            <EIGRP-IPv4 asNumber='1'>
            </EIGRP-IPv4> _
        </Interface>
    </Interfaces>

    <Routing>
      <EIGRP>
           <ProcessIPv4 asNumber="1">
                <Networks>
                     <Network>
                          <IPAddress>10.0.0.6</IPAddress>
                          <Wildcard>0.0.0.3</Wildcard>
                     </Network>
                     <Network>
                          <IPAddress>10.0.0.37</IPAddress>
                          <Wildcard>0.0.0.3</Wildcard>
                     </Network>
                </Networks>
           </ProcessIPv4>
      </EIGRP>
    </Routing>
    <Routing6>
    </Routing6>
</Router>

<!-- R5 -- >
<Router id="10.0.0.41">
    <Interfaces>
        <Interface name="eth0">
            <IPAddress>10.0.0.38</IPAddress>
            <Mask>255.255.255.252</Mask>
            <EIGRP-IPv4 asNumber='1'>
            </EIGRP-IPv4> _
        </Interface>
        <Interface name="eth1">
            <IPAddress>10.0.0.41</IPAddress>
            <Mask>255.255.255.252</Mask>
            <EIGRP-IPv4 asNumber='1'>
            </EIGRP-IPv4> _
        </Interface>
    </Interfaces>

    <Routing>
      <EIGRP>
           <ProcessIPv4 asNumber="1">
                <Networks>
                     <Network>
                          <IPAddress>10.0.0.38</IPAddress>
                          <Wildcard>0.0.0.3</Wildcard>
```

```
                    </Network>
                    <Network>
                         <IPAddress>10.0.0.41</IPAddress>
                         <Wildcard>0.0.0.3</Wildcard>
                    </Network>
               </Networks>
          </ProcessIPv4>
      </EIGRP>
    </Routing>
    <Routing6>
    </Routing6>

</Router>
<!-- R6 -- >
<Router id="10.0.0.42">
    <Interfaces>
        <Interface name="eth0">
            <IPAddress>10.0.0.42</IPAddress>
            <Mask>255.255.255.252</Mask>
            <EIGRP-IPv4 asNumber='1'>
            </EIGRP-IPv4> _
        </Interface>
        <Interface name="eth1">
            <IPAddress>10.0.0.30</IPAddress>
            <Mask>255.255.255.252</Mask>
            <EIGRP-IPv4 asNumber='1'>
            </EIGRP-IPv4> _
        </Interface>
    </Interfaces>
    <Routing>
      <EIGRP>
          <ProcessIPv4 asNumber="1">
               <Networks>
                    <Network>
                         <IPAddress>10.0.0.30</IPAddress>
                         <Wildcard>0.0.0.3</Wildcard>
                    </Network>
                    <Network>
                         <IPAddress>10.0.0.42</IPAddress>
                         <Wildcard>0.0.0.3</Wildcard>
                    </Network>
               </Networks>
          </ProcessIPv4>
      </EIGRP>
    </Routing>
    <Routing6>
    </Routing6>
</Router>
<!-- R7 -- >
 <Router id="10.0.0.33">
    <Interfaces>
        <Interface name="eth0">
            <IPAddress>10.0.0.22</IPAddress>
```

```
            <Mask>255.255.255.252</Mask>
            <EIGRP-IPv4 asNumber='1'>
            </EIGRP-IPv4> _
        </Interface>
        <Interface name="eth1">
            <IPAddress>10.0.0.29</IPAddress>
            <Mask>255.255.255.252</Mask>
            <EIGRP-IPv4 asNumber='1'>
            </EIGRP-IPv4> _
        </Interface>
        <Interface name="eth2">
            <IPAddress>10.0.0.33</IPAddress>
            <Mask>255.255.255.252</Mask>
            <EIGRP-IPv4 asNumber='1'>
            </EIGRP-IPv4> _
        </Interface>
    </Interfaces>

    <Routing>
      <EIGRP>
            <ProcessIPv4 asNumber="1">
                 <Networks>
                      <Network>
                           <IPAddress>10.0.0.22</IPAddress>
                           <Wildcard>0.0.0.3</Wildcard>
                      </Network>
                      <Network>
                           <IPAddress>10.0.0.29</IPAddress>
                           <Wildcard>0.0.0.3</Wildcard>
                      </Network>
                      <Network>
                           <IPAddress>10.0.0.33</IPAddress>
                           <Wildcard>0.0.0.3</Wildcard>
                      </Network>
                 </Networks>
            </ProcessIPv4>
      </EIGRP>
    </Routing>
    <Routing6>
    </Routing6>

</Router>
      <Host id="10.0.0.10">
      <Interfaces>
            <Interface name="eth0">
                 <IPAddress>10.0.0.10</IPAddress>
                 <Mask>255.255.255.252</Mask>
            </Interface>
      </Interfaces>
      <DefaultRouter>10.0.0.9</DefaultRouter>
 </Host>
      <Host id="10.0.0.18">
      <Interfaces>
```

```
          <Interface name="eth0">
               <IPAddress>10.0.0.18</IPAddress>
               <Mask>255.255.255.248</Mask>
          </Interface>
     </Interfaces>
     <DefaultRouter>10.0.0.17</DefaultRouter>
</Host>
     <Host id="10.0.0.19">
     <Interfaces>
          <Interface name="eth0">
               <IPAddress>10.0.0.19</IPAddress>
               <Mask>255.255.255.248</Mask>
          </Interface>
     </Interfaces>
     <DefaultRouter>10.0.0.17</DefaultRouter>
</Host>
     <Host id="10.0.0.26">
     <Interfaces>
          <Interface name="eth0">
               <IPAddress>10.0.0.26</IPAddress>
               <Mask>255.255.255.248</Mask>
          </Interface>
     </Interfaces>
     <DefaultRouter>10.0.0.25</DefaultRouter>
</Host>
     <Host id="10.0.0.27">
     <Interfaces>
          <Interface name="eth0">
               <IPAddress>10.0.0.27</IPAddress>
               <Mask>255.255.255.248</Mask>
          </Interface>
     </Interfaces>
     <DefaultRouter>10.0.0.25</DefaultRouter>
</Host>
     <Host id="10.0.0.34">
     <Interfaces>
          <Interface name="eth0">
               <IPAddress>10.0.0.34</IPAddress>
               <Mask>255.255.255.252</Mask>
          </Interface>
     </Interfaces>
     <DefaultRouter>10.0.0.33</DefaultRouter>
</Host>
   </Devices>
```