Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Метод вибіркової обробки даних в деградуючих полінгових мережах

(тема)

Виконав:

студент II курсу, групи СПм-21-1

Головін В.Д.
(прізвище, ініціали)

Спеціальність

123 «Комп'ютерна інженерія»
(код і повна назва спеціальності)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма

Системне програмування
(повна назва освітньої програми)

Керівник: доц. Ткачов В.М.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ЕОМ _____ Коваленко А.А.
(підпис) (прізвище, ініціали)

2022 р.

Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління

Кафедра електронних обчислювальних машин

Рівень вищої освіти другий (магістерський)

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

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Системне програмування
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
"____" _____ 20___ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студенту Головіну Вячеславу Денисовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Метод вибіркової обробки даних в деградуючих полінгових мережах

затверджена наказом по університету від " 07 " листопада 2022 р. № 1454 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 13 грудня 2022 р.

3. Вхідні дані до роботи мережеві топології;
стек протоколів TCP/IP;
фрагмент комп'ютерної мережі кафедри ЕОМ;
програмне середовище Network Simulator;
програмне середовище Cisco Packet Tracert;
програмне середовище Graphical Network Simulator 3.

4. Перелік питань, що потрібно опрацювати у роботі
Вступ.
1. Принципи побудови та функціонування деградуючих полінгових мереж.
2. Дослідження методів аналізу мереж.
3. Використання модифікованого квест-методу для аналізу деградуючих полінгових
Висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) Демонстраційна презентація. Слайди – 12 штук.

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Огляд прототипів рішення.	08.11.22-11.11.22	
2.	Розробка алгоритмів.	12.11.22-17.11.22	
3.	Моделювання.	18.11.22-21.11.22	
4.	Проведення експериментальних досліджень.	22.11.22-28.11.22	
5.	Складання висновків.	29.11.22-02.12.22	
6.	Оформлення пояснювальної записки	03.12.22-06.12.22	
	та презентаційних матеріалів.		
7.	Подання кваліфікаційної роботи керівникові та	07.12.22-08.12.22	
	попередній захист		
8.	Подання атестаційної роботи на рецензування	09.12.22-12.12.22	

Дата видачі завдання 07 листопада 2022 р.

Студент _____
(підпис)

Керівник роботи _____ доц. Ткачов В.М.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 61 сторінок, 9 рисунків, 2 додатка, 7 джерел.

КОМП’ЮТЕРНА МЕРЕЖА, ДЕГРАДУЮЧА ПОЛІНГОВА МЕРЕЖА, ІНТЕРНЕТ, ПРОТОКОЛ.

Метою кваліфікаційної роботи є модифікація квест-методу аналізу мереж, як одного з методів вибіркової обробки даних в деградуючих полінгових мереж.

У ході виконання кваліфікаційної роботи було розібрано принципи побудови та функціонування деградуючих полінгових мереж, досліджені основні методи аналізу мереж, а саме: контурний, вузловий та ортогональний методи, проаналізовано використання модифікованого квест-методу для аналізу мереж, створено програмний продукт і досліджено ефективність роботи системи аналізу деградуючих полінгових мереж.

ABSTRACT

Master’s thesis: 61 pages, 9 figures, 2 appendices, 7 sources.

NETWORK, DEGRADING POLLING NETWORK, INTERNET, PROTOCOL.

The major goal of this thesis is to modify the quest method for analyzing networks as one of the methods of selective data processing in degrading polling networks.

In order to analyze the principles of building and functioning of degrading polling networks, investigate main methods of network analysis, namely: contour, nodal and orthogonal methods, analyze the use of the modified quest method for network analysis, create a software product and investigate the efficiency of the analysis system for degrading polling networks.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ 7
ВСТУП 8
1 ПРИНЦИПИ ПОБУДОВИ ТА ФУНКЦІОНУВАННЯ ДЕГРАДУЮЧИХ ПОЛІНГОВИХ МЕРЕЖ 10
1.1 Основні принципи побудови і функціонування сучасних мереж 10
1.2 Програмно-реконфігуровані мережі 14
2 ДОСЛІДЖЕННЯ МЕТОДІВ АНАЛІЗУ МЕРЕЖ 22
2.1 Загальні характеристики і мета аналізу 22
2.2 Контурний метод 23
2.3 Вузловий метод 25
2.4 Ортогональний метод 29
2.5 Накладені обмеження на якість обслуговування 34
2.6 Аналіз мережі на практиці 35
3 ВИКОРИСТАННЯ МОДИФІКОВАНОГО КВЕСТ-МЕТОДУ ДЛЯ АНАЛІЗУ ДЕГРАДУЮЧИХ ПОЛІНГОВИХ МЕРЕЖ 39
ВИСНОВКИ 49
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ 50
ДОДАТОК А Графічний матеріал кваліфікаційної роботи 51
ДОДАТОК Б Фрагмент коду програмного забезпечення для аналізу трафіку мережі 588

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ПКМ - програмно-реконфігуровані мережі

API - прикладний програмний інтерфейс(Application Programming Interface)

CLI - інтерфейс командного рядка (Command Line Interface)

VLAN - віртуальна локальна комп'ютерна мережа (Virtual Local Area Network)

ВСТУП

Організаційне завдання поетапної обробки інформації характеризується низкою практичних застосувань, які постійно доповнюються і, таким чином, займають важливе місце в розробці задач комбінаторної оптимізації. Такими завданнями в найширшому розумінні є поточні планові завдання, під час яких здійснюється відбір даних і розробляються траєкторії їх передачі між опитуваними компонентами мережі. Умова кожної задачі містить опис мережі опитування, яка визначає набір можливих шляхів до кінцевого вузла всіх даних. Структурні параметри мережі, як правило, не змінюються протягом усього процесу вирішення проблеми.

У сфері обчислювальної техніки є галузі, об'єднані завданням організації поетапної обробки даних. У процесі аналізу мережевих даних, наприклад, важливим етапом є попередня обробка даних. Він складається з набору послідовних процедур: очищення, нормалізації та ін.

Полінгові мережі для виконання задачі обробки даних можуть бути двох типів:

Багатофункціональні системи обробки даних, кожна з яких виконує повний цикл обробки кожної з частин даних. Тобто частина даних, потрапляючи в таку комп’ютерну мережу, монополізує підсистему протягом усієї обробки.

Однофункціональні системи(характерні для хмарних архітектур). Тобто, різні етапи обробки даних виконуються різними системами, послідовно переміщуючись між ними. Прикладом є цибульна схема, в якій кожному шару може відповідати власний етап обробки даних.

Отже, маючи на увазі ці різновиди, задачі організації покрокової обробки даних можна представити як модифікацію квест-методу, де може

бути декілька таких потоків даних. Характеристикою цих потоків даних є час аналізу у вузлі полінгової мережі, а метою — мінімізація часу, який витрачається на аналіз даних мережі. Отже, метою даної роботи являється аналіз вибіркової обробки даних в деградуючих полінгових мереж з використанням модифікованого квест-методу аналізу мереж, а також дослідження його ефективності в порівнянні з іншими існуючими методами аналізу мереж.

1 ПРИНЦИПИ ПОБУДОВИ ТА ФУНКЦІОНУВАННЯ ДЕГРАДУЮЧИХ ПОЛІНГОВИХ МЕРЕЖ

1.1 Основні принципи побудови і функціонування сучасних мереж

Сучасні мережеві технології являють собою складний набір взаємопов'язаних інтерфейсів, протоколів і алгоритмів взаємодії різних пристроїв. Їх правильне комбіноване застосування дозволяє вирішити широкий спектр завдань, пов'язаних із забезпеченням обміну інформацією [1].

Для опису взаємодії вузлів комп'ютерних мереж прийнято використовувати багаторівневі моделі, такі як OSI. OSI (Open System Interconnection) – це модель взаємодії відкритих систем, яка описує узгоджений набір протоколів(стек протоколів), використання яких забезпечує передачу даних призначених для користувача разом зі службовими мережними повідомленнями, які необхідні для функціонування самої мережі.

Найважливішим стеком протоколів для мереж ж TCP/IP. TCP (Transfer Control Protocol) — спеціальний протокол, який керує передачею даних в мережі. Він створений для встановлення та підтримання надійного з'єднання між пристроями. Саме він відповідає за передачу даних, контролює обсяг файлів, що передаються, і виконує нове відправлення при виникненні збоїв [2].

Основними функціями протоколу TCP є:

- реалізація взаємодії на рівні логічного з’єднання;
- організація потокового відсилання даних;
- встановлення двонаправленого взаємозв'язку;
- надсилання окремих пакетів даних;
- використання принципу «ковзного вікна» збільшення швидкості

передачі.

Але зі стеку TCP/IP більш важливим для роботи мереж є протокол IP (Internet Protocol), який забезпечує всі взаємодії і в локальних мережах, і в Інтернеті. Його основними характеристиками є:

- реалізація обміну даних за допомогою сегментів;
- взаємодія без використання логічного з'єднання;
- фрагментація IP-сегментів у разі потреби;
- відсутність коштів на управління швидкістю передачі сегментів.

IP не був розроблений для роботи в сучасних мережах Інтернету, і тому у певний момент розвитку зіткнувся з такими проблемами:

- стрімке зростання складності маршрутизації великого числа мереж;
- складність масового зміни адрес, через їх залежність від провайдера;
- ризик того, що IP-адреси скоро закінчаться.

Трансляція мережевих адрес (NAT - Network Address Translation) є найефективнішим і найпоширенішим способом вирішення проблеми кількості IP-адрес. Ця техніка дозволяє замінити адреси великої кількості комп’ютерів у вашій локальній мережі однією адресою шлюзу у зовнішній мережі(зазвичай в Інтернеті). Використання NAT не лише зберігає IP-адреси, але й підвищує безпеку, приховуючи внутрішню мережеву інфраструктуру. Однак ця технологія теж має недоліки, одним з яких є ізольованість внутрішньої мережі від зовнішніх з'єднань і абсолютна «невидимість» для Інтернету. Така ситуація неприйнятна для територіально розосереджених організацій, інформаційні ресурси яких можуть бути розподілені між декількома внутрішніми мережами, і для забезпечення зв'язку між ними була створена кругова мережа на основі технологій шифрування та тунелювання:

- тунелювання — це процес передачі комунікацій приватних мереж через загальнодоступну мережу, наприклад Інтернет, через процес, який називається інкапсуляцією;

- інкапсуляція — це процес, що дозволяє пакетам даних виглядати так, ніби вони мають публічний характер для публічної мережі, коли вони є фактично приватними пакетами даних, дозволяючи їм проходити непомітно.

В процесі інкапсуляції виділяють наступні типи протоколів:

- транспортний протокол, дані якого потрібно передати через тунель;

- протокол інкапсуляції, який потрібен для позначення факту тунелювання і передачі параметрів тунелю;

- несучий протокол, який передає перераховану вище інформацію.

Тунелювання в полінгових мережах відрізняється від традиційних багаторівневих моделей мереж (таких як OSI або TCP/IP), оскільки транспортный протокол належить до того самого або нижчого рівня, ніж протокол, який використовується для тунелю (рисунок 1.1).

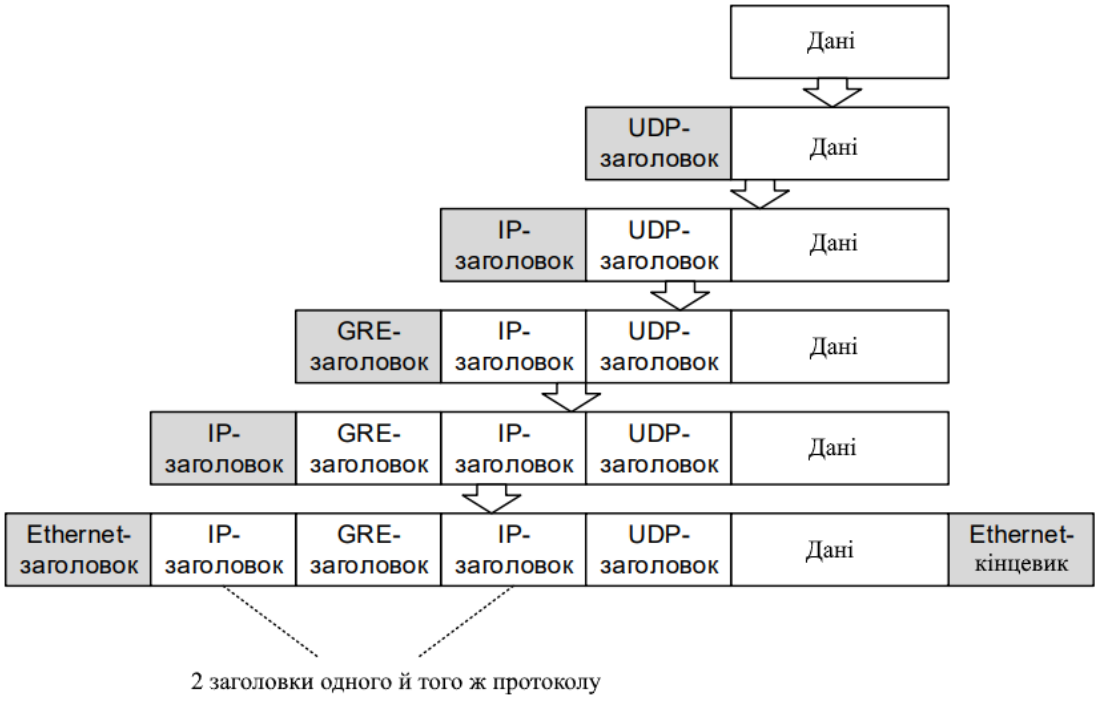

Рисунок 1.1 – Інкапсуляція пакетів при тунелюванні

Стандартним підходом інкапсуляції є використання UDP, як транспортним протоколом, GRE – протоколом інкапсуляції, а IP, як несучим і транспортним протоколов водночас (рисунок 1.1). Цей метод легко

ідентифікується системами аналізу трафіку та, без додаткового захисту, піддається подальшому аналізу для отримання корисної інформації. Також, досить перспективним є метод, що використовує принципи мережевої стеганографії(сукупності методів прихованої передачі даних, при якому повідомлення, закодовується так, що не виглядає як повідомлення).

Усі мережеві протоколи можна умовно розділити на дві групи:

- протоколи, які використовуються для передачі корисного навантаження (зазвичай даних користувача). До цієї групи входять усі протоколи прикладного рівня, UDP, IP, TCP та Ethernet;

- службові протоколи, які необхідні для обміну інформацією про топології мережі, її стан, зіставлення адрес, а також виконують велику кількість інших функцій, які безпосередньо не пов’язані з передачею даних користувача, але необхідні для належного функціонування мереж. До них відносяться DNS, ICMP, DHCP, ARP, усі протоколи маршрутизації та багато інших.

Найбільш простим для інкапсуляції полінгових даних є протокол ICMP. В його пакетах передбачається місце для розміщення будь-яких даних, які безпосередньо не використовуються самим протоколом.

Також набув досить широкого розповсюдження службовий протокол ARP (Address Resolution Protocol), який призначений для зіставлення MAC і IP адрес. В ньому не виділяється окреме місце для запису даних, натомість можна використовувати поля його заголовка, виділені для однієї з адрес відправника.

Зазвичай під час конфігурації системи аналізу трафіку правила сервісного протоколу не такі суворі, як транспортного протоколу, а іноді навіть повністю ігноруються задля економії обчислювальних ресурсів і уявленням про те, що в службових протоколах не повинно бути даних, що призначені для користувача, які могли б представляти інтерес для аналізу.

При використанні динамічної маршрутизації в мережі, через пакети протоколу, що використовується, також можуть передаватися повідомлення

користувача. Ці пакети мають велику кількість даних, наприклад, про існуючі маршрути(для протоколу RIP) або про топологію мережі(для протоколу ICMP). Додавання в такі пакети великих обсягів даних не являється маршрутною інформацією, адже проміжний вузол, що здійснює аналіз трафіку, не може перевірити їх зміст, а довжина даних не обмежена.

Використання принципів стеганографії при впровадженні даних в мережеві пакети заключається в:

- визначенні в структурі пакета інформації, яка є потенційно надлишковою;

- компресії структури;

- запису необхідної інформації в місце, що вивільнилось [3].

Даний принцип показаний на рисунку 1.2.

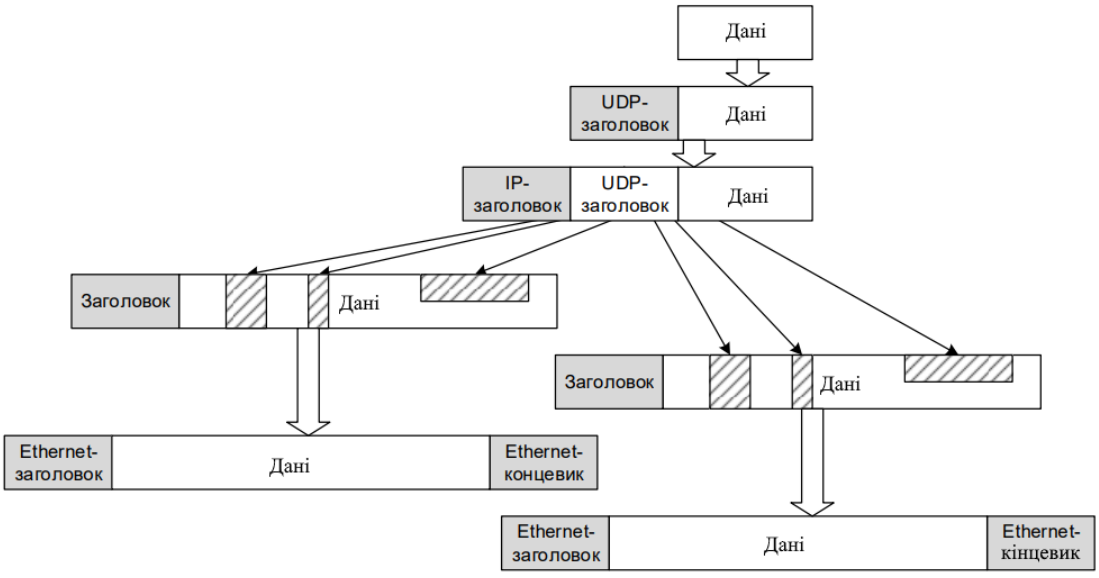

Рисунок 1.2 – Інкапсуляція даних з використанням стенографії

1.2 Програмно-реконфігуровані мережі

Ще одним різновидом полінгових мереж є програмно-реконфігуровані мережі(ПКМ). Можно виділити три способи реалізації ПКМ:

- ПКМ, що створені на основі оригінальної версії, яка була запропонована фахівцями Стенфордського університету (США) в 2007 р. Цей варіант передбачає перенесення функцій управління мережею з мережевих пристроїв на централізований контролер за допомогою протоколу OpenFlow. Така ПКС повинна мати такі п'ять основних атрибутів: мати окремий рівень управління, використовувати автоматизацію, віртуалізацію і більш просте мережеве обладнання, мати централізоване управління мережею, бути відключеною від мережі та бути відкритою для експертів і розробників. Для мереж з перерахованими властивостями використовується окремий термін - "Open SDN". Для такого ПКМ обов’язковим є наявність контролера, що підтримує інтерфейс OpenFlow для роботи з мережею.

- ПКМ, що створені на основі вже існуючих API. При створенні ПКМ використовуються функції API, які часто можна викликати віддалено за допомогою традиційних механізмів, таких як CLI та SNMP, або при використанні нових гнучких механізмів, таких як RESTful API.

- ПКМ, що створені на основі полінгових мереж і гіпервізора. Реалізація ПКМ при такому способі не залежить від інфраструктури мережі, бо ПКМ накладається поверх вже існуючої фізичної мережі.

Другий і третій варіанти створення ПКМ є альтернативами Open SDN, але іноді краще підходять для створення полінгових комп'ютерних мереж.

Завдяки тому, що з комутаторів знімається завдання управляти просуванням даних трафік починає пересуватися швидше, що значно підвищує продуктивність. До того ж відбувається зниження витрат на їх побудову і супровід, завдяки віртуалізації управління мережею. Системні адміністратори на централізованому контролері Open SDN можуть бачити всю мережу в одному поданні, що підвищує простоту адміністрування, забезпечує безпеку та спрощує виконання багатьох інших задач. В теорії Open SDN має необмежений потенціал до розширення, що дозволяє будувати хмари з можливістю масштабування в залежності від типу розв'язуваних завдань. При цьому мережа має необхідний «інтелект», що особливо важливо

для координації роботи великої кількості груп комутаторів [4].

Але дуже скоро з'явилося чимало критики, яка пов'язана і з самоє архітектурою Open SDN, і з питаннями щодо практичності впровадження таких мереж. Більш за все критики було від виробників мережевого обладнання, через радикальні, на їх думку, змінами, які принесла з собою нова мережева технологія:

- велика вартість необхідного обладнання(комутаторів з підтримкою OpenFlow);

- ризики, що виникають через недостатнє тестування нового обладнання, що повинно массово впроваджуватися у клієнтів.

З точки зору архітектури, найсерйозніший аргумент критиків Open SDN полягає в тому, що такі мережі є вразливими через наявність лише однієї точки відмови. Схема переміщення даних у мережі Open SDN з одним контролером зображена на рисунку 1.3. Коли цей контролер виходить з ладу, вся мережа не може працювати. Тобто зображений контролер ПКМ є єдиною точкою відмови.

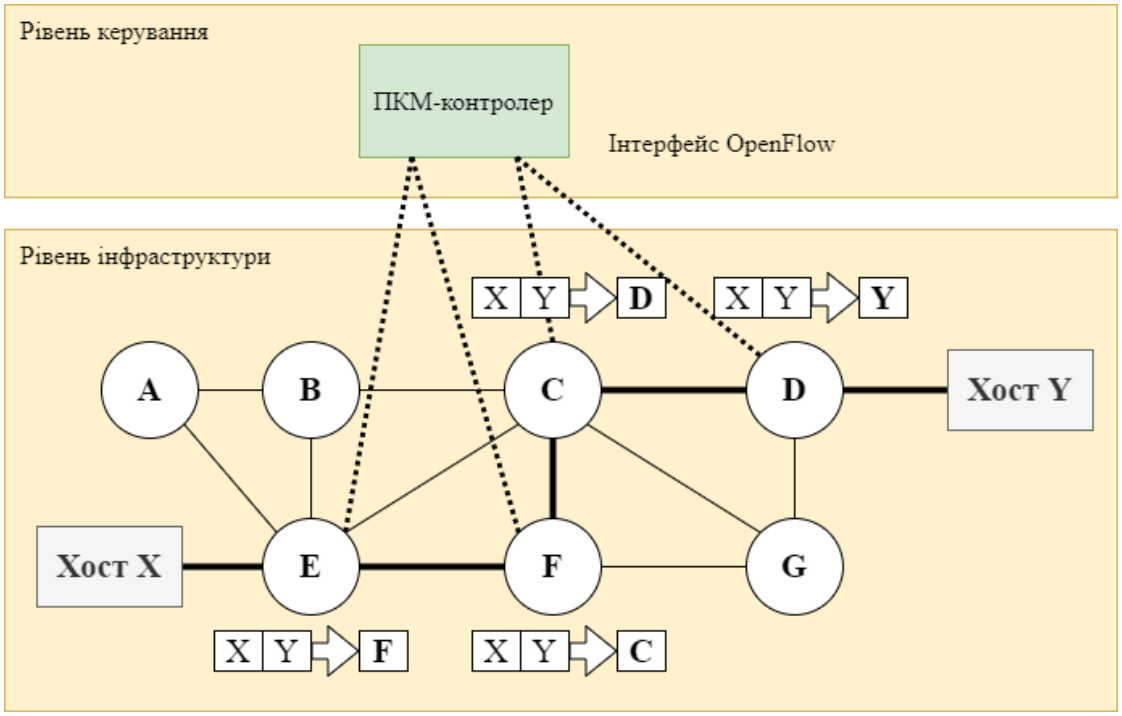

Рисунок 1.3 – Переміщення даних в мережі OpenSDN

Для подолання цього недоліку, треба мати декілька контролерів в мережі, які взаємодіють між собою за допомогою особливо надійних ліній зв'язку. Це робить архітектуру Open SDN складніше і дорожче.

Розглянемо ПКМ, що створені на основі існуючих API. Якщо надати мережевим пристроям можливість розпізнавати більш широкий набір команд API, за допомогою яких контролери можуть гнучко керувати пристроями та всією мережею, то це буде ПКМ на основі існуючих API (рисунок 1.4).

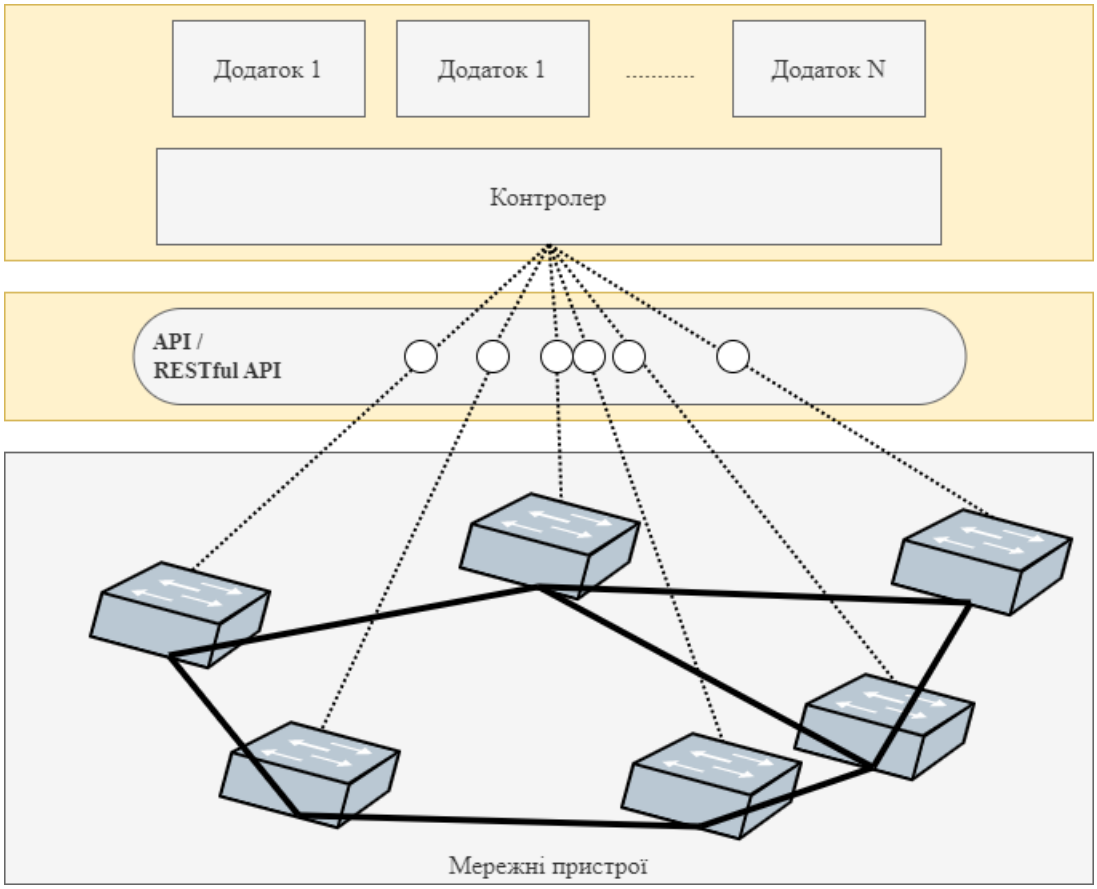

Рисунок 1.4 – ПКМ, створена на основі існуючих API

Для створення ПКМ по такій схемі виробники пристроїв модернізували вже існуючі API(наприклад, використовуючи RESTful API, а не традиційні CLI або SNMP). Механізми CLI і SNMP розроблені давно і використовуються під час налаштування мережі. Але сьогодні ці механізми

громіздкі і незручні, коли потрібне швидке динамічне управління великими мережами або центрами обробки даних. На зміну їм прийшла нова технологія – RESTful API. В останній час цей механізм став найпоширенішим механізмом при передачі запитів API в мережі. RESTful API працює за допомогою HTTP(протоколу передачі гіпертексту), який зазвичай використовується для передачі веб-трафіку. RESTful API є відносно простою технологією та легко розширюється. Простий у використанні, оскільки він використовує стандартні порти TCP і не потребує спеціальних налаштувань брандмауера для передачі запитів API. ПКМ, створені на основі існуючих API, мають багато переваг:

- не потрібно впроваджувати комутатори нового типу, що підтримують стандарт OpenFlow, адже такі ПКМ можуть працювати зі звичайними комутаторами;

- використання існуючих API дозволяє побудувати централізовано керовану мережу в масштабі. Це призводить до більш відкритої архітектури мережі, оскільки виробники змушені відкривати специфікації інтерфейсу для своїх пропрієтарних пристроїв;

- доступні API дозволяють легко писати програмне забезпечення оркестровки мережевих подій, яке швидко й автоматично реагує на зміни в мережі, такі як динамічний рух віртуальних машин у центрі обробки даних.

ПКМ, створені з існуючих API, також мають свої недоліки. По-перше, в такій мережі в більшості випадків взагалі відсутній контролер. Тому мережеві програмісти змушені програмувати кожен комутатор безпосередньо. Але навіть з контролерами програмістам не вистачає загального стандартного механізму взаємодії з мережевими пристроями. Тобто програмісти змушені розбиратися в технічних характеристиках кожного інтерфейсу комутатора. Також недоліком є те, що програмне забезпечення для цього ПКМ можна використовувати лише для певних конфігурацій мережі. Це зрозуміло, оскільки API для пристроїв різних виробників не є частиною загального стандарту (на відміну від протоколу

OpenFlow). Таким чином, цей тип ПКМ зможе працювати з обладнанням певного виробника або з невеликим набором сумісного обладнання [5].

Інший недолік пов’язаний з фактом, що архітектура ПКМ вимагає перенесення функції керування з комутатора на контролер, щоб створити простіший і менш дорогий комутатор. Ця задача залишилася невирішеною: для створення ПКС потрібні складні і дорогі комутатори. І нарешті, хоча ПКС, створені на основі існуючих API, дозволяють певною мірою керувати пересиланням даних, особливо при побудові VPN чи VLAN, вони не дозволяють більш детального контролю над кожним інформаційним потоком, який можливий за допомогою протоколу OpenFlow.

Якщо досить радикальне рішення на основі OpenFlow ще не створено або не має сенсу з будь-яких причин, то можно розглянути ПКМ на основі існуючих API як практичний засіб розширити вже існуючий функціонал полінгової комп'ютерної мережі. Загалом, створення ПКМ на основі існуючого API є кроком у правильному напрямку, кроком до створення полінгової мережі з повністю централізованим керуванням.

Іншим, більш інноваційним способом створення ПКМ є використання накладеної мережі та керування нею через гіпервізор. При цьому способі фізична мережа залишається незмінною, але на цій фізичній мережі створюється віртуальна мережа з гіпервізором. Програми на мережевих вузлах взаємодіють із цими віртуальними мережами без необхідності знати фізичні характеристики мережі, яка передає дані (рис. 1.5).

Через те, що віртуальні мережі розташовані поверх фізичної інфраструктури, вони можуть керуватися системами (або пристрої), розташованими в кінцевих точках мережі. У центрі обробки даних така система є гіпервізором для віртуальних машин, які існують на кожному сервері. Трафік у віртуальній мережі тунелюється за допомогою інкапсуляції. Тобто, коли пакет надходить до вузла віртуальної мережі для передачі, мережевий пристрій (зазвичай гіпервізор) інкапсулює пакет в інший кадр. ПКМ, створені на основі віртуальних мереж і гіпервізорів, добре працюють у

центрах обробки даних, де програмне забезпечення віртуалізації серверів вже встановлено. Вони допомагають усунути багато проблем, які виникають під час роботи центру обробки даних:

- відоме обмеження на кількість VLAN, підтримуваних у локальній мережі, більше не має значення, оскільки тут для розподілу трафіку масового трафіку використовуються тунелювання (замість VLAN);

- побудова центрів обробки даних на базі ПКМ дозволяє дуже швидко і гнучко змінювати характеристики мереж, залучених до обчислювального процесу центру обробки даних, завдяки можливості централізованого програмного керування віртуальною мережею ПКМ;

- різке зростання MAC-адрес вузлів усувається, оскільки тепер MAC-адреси приховані в інкапсульованому кадрі.

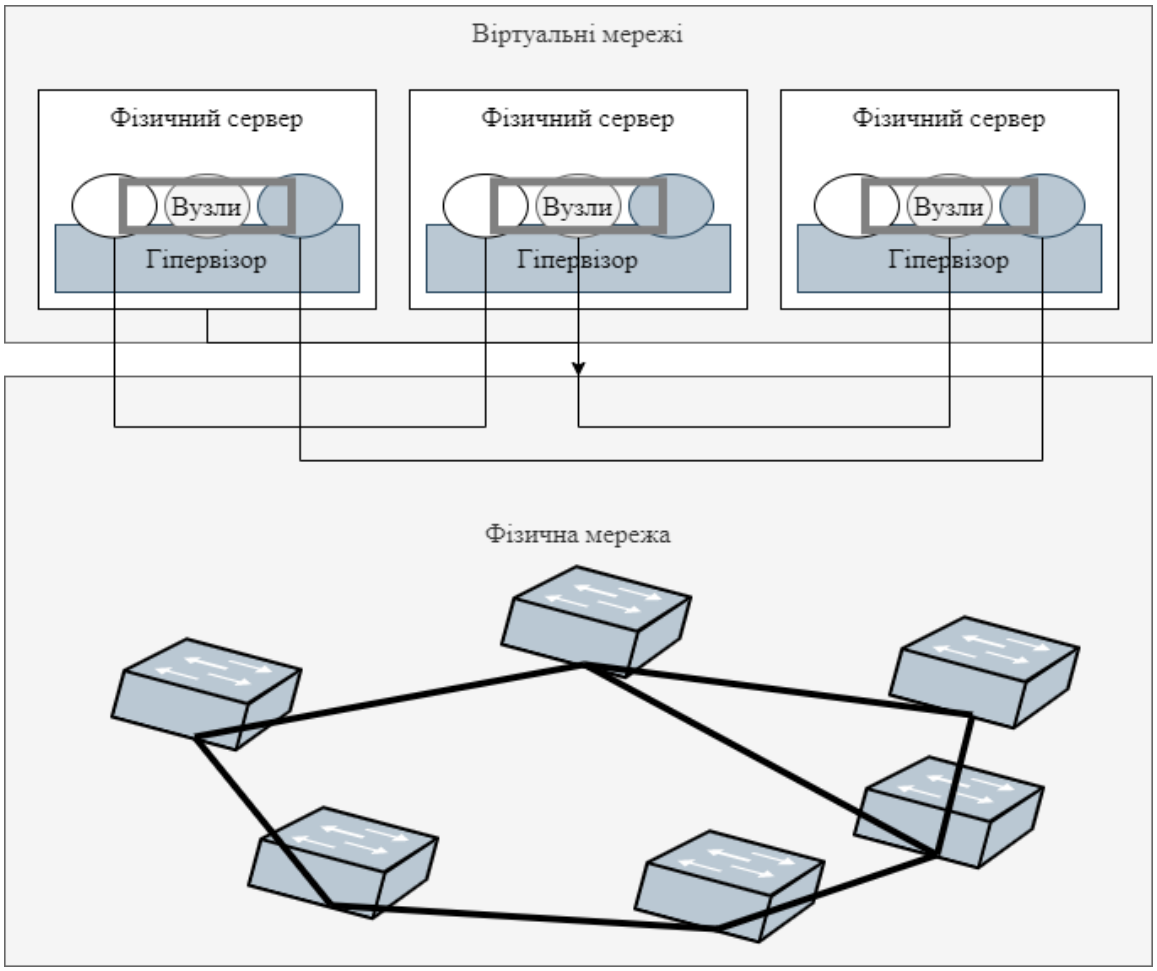

Рисунок 1.5 – Віртуальні мережі, накладені на фізичну мережу

ПКМ на основі віртуальних мереж і гіпервізорів не вирішують усіх проблем. Зокрема, інфраструктура фізичної мережі все ще вимагає ручного налаштування та обслуговування(наприклад, це стосується протоколів STP та QoS). Також, як і в попередньому способі, мережні пристрої залишаються незмінними. Натомість полінгова мережа може бути динамічною, зі змінною кількістю вузлів.

2 ДОСЛІДЖЕННЯ МЕТОДІВ АНАЛІЗУ МЕРЕЖ

2.1 Загальні характеристики і мета аналізу

Для аналізу мереж використовуються такі вхідні дані:

- топологія мережі провайдера;
- матриця запитів між вузлами мережі;
- потоки між вузлами мережі, які зазвичай представляються у вигляді матриці запитів NxN, де nij – інтенсивність потоку з вузда і до вузла j всередині мережі.

Під час дослідження мережі, увага звертається на такі характеристики:

- інтенсивність потоку трафіку;
- навантаження на канали з урахуванням обмежень на параметри якості обслуговування.

Як результат аналізу буде виведено маршрут пересування трафіку між вузлами мережі.

Основна ідея тензорного методу полягає в тому, що всі топології, що містять однакову кількість гілок, з'єднані тензором перетворення, роль якого може виконувати матриця лінійно незалежних ділянок, матриця лінійно незалежних контурів або комбінована лінійна матриця для незалежні контури і розрізи. Завдяки цій пов’язанності тензором перетворення серед багатьох проекцій мереж можно виділити одну – примітивну мережу. Ця мережа має стільки ж гілок, слільки і в досліджувана мережа, але в той же час у неї кількість не пов'язаних компонент така ж, як і кількість гілок, що робить потоки в кожній із гілок примітивної мережі незалежними.

Для визначення математичної моделі найпростішого елемента мережі можно скористатися узагальненням Крона, який стверджує, що математична модель примітивної мережі має тривіальний вид і показує, як пов’язані між собою контурний час обслуговування, контурні навантаження і контурні

інтенсивності у матричній формі:

$$\begin{bmatrix} \rho_1 \\ \rho_2 \\ \dots \\ \rho_n \end{bmatrix} = \begin{bmatrix} t_1 & 0 & \dots & 0 \\ 0 & t_2 & \dots & 0 \\ \dots & \dots & \dots & 0 \\ 0 & 0 & \dots & t_n \end{bmatrix} \star \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \dots \\ \lambda_n \end{bmatrix}, \quad (2.1)$$

де: ρ_i – завантаження і-го елемента;

λ_i – інтенсивність надходження;

t_i – середній час обслуговування.

2.2 Контурний метод

Якщо взяти лінійно незалежні контури як базисні елементи у кожній топології, то тензор перетворення буде встановлювати зв’язок між контурними інтенсивностями двох мереж. Таким чином тензорне перетворення встановлює зв’язок між примітивною і досліджуваною мережами. Цикломатичне число використовується для визначення кількості контурних інтенсивностей аналізованої мережі і дорівнює p=n-m+1.

Тензор перетворення C представляється у вигляді матриці лінійно незалежних контурів. Існує два основних методи для отримання C:

- граф-комбінаторний. Цей метод базується на алгоритмах пошуку в глибину та пошуку в ширину. Важливим моментом є те, що складність цих алгоритмів зростає лінійно разом зі збільшенням кількості гілок в мережі і дорівнює O(N);

- аналітичний. Цей метод допускає що матриця контурів ортогональна до матриці інцидентності, адже вона також виступає матрицею розрізів, за виключенням того, що вона має один лінійно залежний рядок. Це означає, що можна отримати матрицю з нульовими елементами, перемноживши матрицю інцидентності без одного рядка (який можно обрати довільно) з матрицею

лінійно незалежних контурів.

Якщо контури, які розташовані у кожному стовпці, лінійно незалежні, то значення елементів матриці контурів не важливе. Але для розрахунків найзручнішою являється матриця не тільки з лінійно незалежними контурами, але і з фундаментальними циклами, яка і була обрана в ролі тензора перетворення. Завдяки матриці фундаментальних циклів можно явно побачити, що при перетворенні примітивної мережі у досліджувану, якщо усі вказані в контурах примітивної мережі потоки дорівнюють нулю, потоки в останніх гілках будуть визначені як лінійна комбінація вказаних потоків.

Щоб записати зв'язок між контурними інтенсивностями примітивної і досліджуваної мереж, можно використати наступну формулу:

$$\lambda^{\bar{i}} = C_{i}^{\bar{i}} \lambda^{i}, \tag{2.2}$$

де $\lambda^{\bar{i}}$ являється контурною інтенсивністю примітивної мережі;

$C_{i}^{\bar{i}}$ являється тензором перетворення;

λ^{i} являється контурною інтенсивністю досліджуваної мережі.

Скориставшись простими матричними перетвореннями можно представити вектор контурних навантажень досліджуваної мережі у вигляді $\rho_i = \tilde{C_i^i} \rho_i$, а матрицю тривалостей обслуговування досліджуваної мережі, як:

$$t_{\bar{j}\bar{i}} = t_{\bar{j}\bar{i}} C_{j}^{\bar{j}'} C_{i}^{\bar{i}'}, \tag{2.3}$$

де C' – тензор перетворення без лінійно залежних рядків.

Відповідно до отриманих контурних часів обслуговування і контурних завантажень можно представити значення контурних інтенсивностей досліджуваної мережі у такому вигляді:

$$\lambda^i = (t_{ji})^{-1}\rho_j \quad , \tag{2.4}$$

Значення потоків у кожній гілці, в такому випадку буде дорівнювати:

$$\lambda^{\bar{i}}_{\text{гілок}} = C^{\bar{i}}_i \lambda^i \quad , \tag{2.5}$$

При проведенні тензорного аналізу характеристик кожної з відомих мереж окремо, маючи для них відповідні матриці запитів, тобто кожна з мереж має відповідне рівняння виду (2.4), і підсумовуючи однойменні рядки кожного з рівнянь ми можемо отримати систему рівнянь, яка буде відображати залежність між інтенсивностями потоків та завантаженням кожної зі створюваних мереж. Результуюча система рівнянь для визначення потоків мереж у кожній гілці, при кількості мереж N, буде мати вигляд:

$$\lambda^{\bar{i}}_{\text{гілок}-\text{загальне}} = C^{\bar{i}}_i \sum_{q=1}^{N}\sum_{p=1}^{K} \lambda^i_{qp} \quad , \tag{2.6}$$

де N – число гілок досліджуваної мережі;

S·p – число невідомих;

S – загальна кількість потоків мереж.

2.3 Вузловий метод

Проблема тензорних перетворень за Г.Кроном полягає в тому. що тензор перетворення між топологіями примітивної та аналізованої мереж є сингулярною матрицею, що неможливо, тому що тоді не існує зворотного перетворення.

Якщо в якості базисних елементів кожної топології будуть використовуватися лінійно-незалежні розрізи, то тензор перетворення встановлює зв'язок між вузловими завантаженнями однієї мережі з вузловими завантаженнями іншої мережі, так зв'язок вузлових навантажень примітивної мережі пов'язана з вузловими завантаженнями досліджуваної мережі тензором перетворення А. Цикломатичне число використовується для визначення кількості контурних інтенсивностей аналізованої мережі і дорівнює p=n - 1.

Тензор перетворення А визначається як матриця лінійно-незалежних розрізів, що у свою чергу дорівнює матриці інцидентності без одного рядка. Даний метод передбачає, що аналізована мережа не містить контурів, тому перед початком дослідження мережі її необхідно перетворити на вузлову. Можна запропонувати два варіанти перетворення мережі з довільною топологією у суто вузлову мережу:

- використання розриву вузла для перетворення мережі з довільною топологією. При цьому слід зазначити, що від вузла відриваються лише хорди;

- використання додаткових систем масового обслуговування, інтенсивність обслуговування яких необхідно прийняти рівну нескінченності, а інтенсивність трафіку, що проходить через неї, необхідно залишити такою самою, як у тій мережі, яку дана мережа доповнює. Кількість додаткових схем визначається числом хорд графа, які інцидентні вузлам, що з'єднує більше двох гілок.

Слід зазначити, що перетворення у вузлову мережу еквівалентно збільшенню розмірності матриці інцидентності, у першому випадку додаються лише нові вузли (нові строки), в той час як у другому випадку дадаються ще і нові гілки (стовпці). Більш раціональним способом є розірвання контурів у вузлах, до яких інцидентні лише два ребра, оскільки при цьому не будуть додаватися додаткові мережі. Якщо ж розрив необхідно зробити у вузлі, якому інцидентні більше ніж дві гілки, то додаткові схеми

будуть необхідні.

Використання першого методу забезпечує мінімум перетворень над мережею, тобто матриця інцидентності, якій була задана вихідна мережа, буде перетворена на нову матрицю шляхом додавання стількох рядків, скільки було зроблено розривів.

У другому варіанті в матриці інцидентності з'являються не тільки рядки, зумовлені появою нових вузлів, а й стовпці, що відображають появу додаткових ребер. Але у зв'язку з тим, що кількість гілок, які інцидентні вузлам, що належать до не перетвореної мережі, залишається не незмінною, то після перетворення сума потоків у вузлах, яким інцидентні два і більше ребра, залишається рівною нулю, що призведе до більш простого виду матриці вузлових інтенсивностей надходження, отриманої за формулою:

$$\lambda^{\bar{i}} = A_i^{\bar{i}} \lambda^i, \tag{2.7}$$

Зв'язок вузлових завантажень примітивної мережі з досліджуваної можна, як було зазначено, записати так:

$$\rho_{\bar{j}} = A_j^{\bar{j}} \rho_j, \tag{2.8}$$

де $\rho_{\bar{j}}$ – вузлові завантаження примітивної мережі;

$A_j^{\bar{j}}$ – тензор перетворення вузлових навантажень досліджуваної мережі в вузлові інтенсивності примітивної мережі;

ρ_j – значення вузлових навантажень досліджуваної мережі.

Виконав прості матричні перетворення можно показати, що:

$$\mu_{ij} = \mu_{\bar{ij}} A_j^{\bar{j}} A_i^{\bar{i'}}, \quad (2.9)$$

де $A_j^{\bar{j}}$ – вектор вузлових інтенсивностей надходження досліджуваної мережі;

$A_i^{\bar{i'}}$ – вектор вузлових інтенсивностей надходження примітивної мережі;

μ_{ij} – матриця вузлових інтенсивностей обслуговування досліджуваної мережі;

$\mu_{\bar{ij}}$ – матриця вузлових інтенсивностей обслуговування примітивної мережі.

На підставі отриманих значень вузлових інтенсивностей надходження та вузлових інтенсивностей обслуговування значення вузлових навантажень досліджуваної мережі будуть визначатися як:

$$\rho_j = (\mu_{ij})^{-1} \lambda^i, \quad (2.10)$$

А значення навантажень у кожній гілки можна визначити так:

$$\rho_{\text{гілок}} = A_i^{\bar{i}} \rho_i \quad (2.11)$$

При проведенні тензорного аналізу характеристик кожної з відомих мереж окремо, маючи для них відповідні матриці запитів, тобто кожна з мереж має відповідне рівняння виду (2.11), і підсумовуючи однойменні рядки кожного з рівнянь ми можемо отримати систему рівнянь, яка буде відображати залежність між інтенсивностями потоків та завантаженням

кожної зі створюваних мереж. Результуюча система рівнянь для визначення потоків мереж у кожній гілці, при кількості мереж N, буде мати вигляд:

$$\rho_{\text{тілок}-\text{загальне}} = A_i^{\bar{i}} \sum_{q=1}^{N} \sum_{p=1}^{K} \rho_{qp} \quad , \qquad (2.12)$$

де ρ_{qp} - Вектор вузлових навантажень, створювана p-м сайтом q-й VPN мережі.

2.4 Ортогональний метод

Поряд із вищевикладеними двома підходами можна запропонувати метод аналізу ортогональних мереж, в якому не потрібні перетворення над вихідною мережею. У такому разі примітивна мережа для ортогональної мережі складатиметься з набору примітивних вузлових елементів та набору контурних елементів.

Таким чином, в ортогональній мережі базисними елементами буде сукупність лінійно незалежних контурів та лінійно незалежних розрізів. Отже, необхідно встановити таке перетворення:

$$\lambda = X\bar{\lambda} \quad , \qquad (2.13)$$

де λ – вектор, що містить як вузлові, так і контурні інтенсивності досліджуваної мережі;

$\bar{\lambda}$ – вектор примітивних елементів, що містить як контурні, і вузлові примітивні елементи;

X – тензор перетворення.

Тензор перетворення *X* складається з двох складових, тензора *A*, який зв'язує вузлові інтенсивності примітивної мережі з вузловими інтенсивностями досліджуваної мережі, та тензора перетворення *C'*, який пов'язує контурні інтенсивності примітивної мережі з контурними інтенсивностями досліджуваної.

Правило отримання матриці *C'*, полягає в тому, що необхідно однозначно зіставити до однієї контурної інтенсивності примітивної мережі лише одну контурну інтенсивність досліджуваної мережі. Як відомо з теорії графів, сукупність всіх ребер графа ділиться на гілки і хорди, тобто контурна інтенсивність у досліджуваній мережі відповідає інтенсивності у тій гілці примітивної мережі, яка після перетворення на досліджувану мережу стала хордовим ребром.

Нижче наведемо приклад отримання матриці *X*. Давайте розглянемо мережу (рисунок 2.1), для якої введені лінійно-незалежні контурні інтенсивності λ_α, λ_β, λ_γ, та лінійно-незалежні вузлові інтенсивності $i \in (1..9)$.

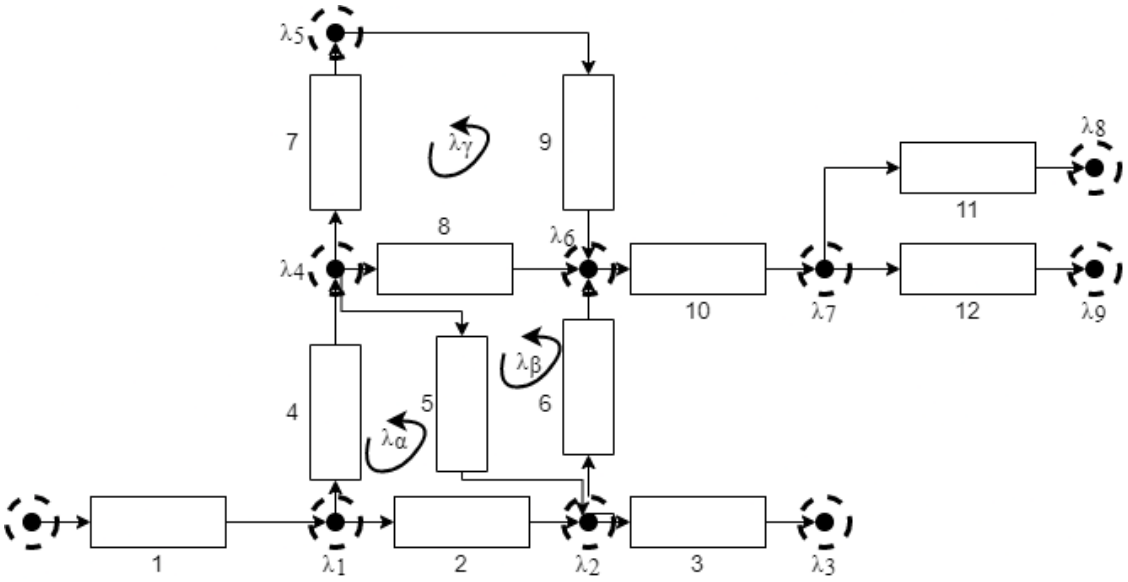

Рисунок 2.1 – Аналізована мережа

Система рівнянь для досліджуваної мережі буде виглядати так:

$$
\begin{cases}
\text{Вузлові інтенсивності} \\
\lambda_1 = \bar{\lambda}_1 - \bar{\lambda}_2 - \bar{\lambda}_4 \\
\lambda_2 = \bar{\lambda}_2 - \bar{\lambda}_3 + \bar{\lambda}_5 - \bar{\lambda}_6 \\
\lambda_3 = \bar{\lambda}_3 \\
\lambda_4 = \bar{\lambda}_4 - \bar{\lambda}_5 - \bar{\lambda}_7 - \bar{\lambda}_8 \\
\lambda_5 = \bar{\lambda}_7 - \bar{\lambda}_9 \\
\lambda_6 = \bar{\lambda}_6 + \bar{\lambda}_8 + \bar{\lambda}_9 - \bar{\lambda}_{10} \\
\lambda_7 = \bar{\lambda}_{10} - \bar{\lambda}_{11} - \bar{\lambda}_{12} \\
\lambda_8 = \bar{\lambda}_{11} \\
\lambda_9 = \bar{\lambda}_{12} \\
\text{Контурні інтенсивності} \\
\lambda_\alpha = \bar{\lambda}_5 \\
\lambda_\beta = \bar{\lambda}_6 \\
\lambda_\gamma = \bar{\lambda}_7
\end{cases}, \qquad (2.14)
$$

У матричному вигляді система рівнянь виглядатиме так:

$$
\begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \\ \lambda_5 \\ \lambda_6 \\ \lambda_7 \\ \lambda_8 \\ \lambda_9 \\ \lambda_\alpha \\ \lambda_\beta \\ \lambda_\gamma \end{bmatrix} = \begin{bmatrix} 1 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} \bar{\lambda}_1 \\ \bar{\lambda}_2 \\ \bar{\lambda}_3 \\ \bar{\lambda}_4 \\ \bar{\lambda}_5 \\ \bar{\lambda}_6 \\ \bar{\lambda}_7 \\ \bar{\lambda}_8 \\ \bar{\lambda}_9 \\ \bar{\lambda}_{10} \\ \bar{\lambda}_{11} \\ \bar{\lambda}_{12} \end{bmatrix}, \qquad (2.15)
$$

Для отримання тензора перетворення між примітивною та аналізованою мережею можна скористатися математичним апаратом теорії графів, згідно з яким перехід між вузловими інтенсивностями примітивної до вузлових інтенсивностей аналізованої мережі забезпечується за допомогою матриці інцидентності без одного рядка, а зв'язок між контурними інтенсивностями забезпечується за допомогою матриці хорд. Відповідно,

якщо дотримуватися позначень теорії графів, то систему рівнянь можна записати як:

$$\lambda = \begin{bmatrix} I' \\ H \end{bmatrix} \bar{\lambda}, \tag{2.16}$$

де I' – матриця інцидентності без лінійно-незалежного рядка;

H – матриця хорд графа.

Алгоритми отримання цих матриць відомі, і складність цих алгоритмів зростає лінійно зі зростанням кількості гілок у мережі. Позначимо тензор перетворення $\begin{bmatrix} I' \\ H \end{bmatrix}$ за X, тоді рівняння (2.16) можна привести до такого вигляду:

$$\lambda^{i} = X_{\bar{i}}^{\bar{i}} \lambda^{\bar{i}}, \tag{2.17}$$

Відповідно, якщо провести аналіз по Крону, можно вивести наступні відношення:

$$\rho_{\bar{j}} = X_{\bar{j}}^{j} \rho_{j}, \tag{2.18}$$

де $\rho_{\bar{j}}$ – навантаження примітивної мережі;

ρ_{j} – навантаження досліджуваної мережі.

$$\mu^{\bar{j}\bar{i}} = \mu^{\bar{j}\bar{i}} X_{j}^{\bar{j}} X_{i}^{\bar{i}}, \tag{2.19}$$

де μ^{ji} – матриця фазових інтенсивностей обслуговування досліджуваної мережі;

$\bar{\mu^{ji}}$ – матриця фазових інтенсивностей обслуговування примітивної мережі.

На підставі отриманих значень фазових інтенсивностей надходження та фазових інтенсивностей обслуговування значення фазових завантажень досліджуваної мережі будуть визначатися як:

$$\rho_j = (\mu^{ji})^{-1}\lambda^i, \tag{2.20}$$

А значення навантажень у кожній гілці можна визначити наступним чином:

$$\rho_{\text{гілок}} = X_{\bar{j}}^{j}\rho_j, \tag{2.21}$$

При проведенні тензорного аналізу характеристик кожної з відомих мереж окремо, маючи для них відповідні матриці запитів, тобто кожна з мереж має відповідне рівняння виду (2.21), і підсумовуючи однойменні рядки кожного з рівнянь ми можемо отримати систему рівнянь, яка буде відображати залежність між інтенсивностями потоків та завантаженням кожної зі створюваних мереж. Результуюча система рівнянь для визначення потоків мереж у кожній гілці, при кількості мереж N, буде мати вигляд:

$$\rho_{\text{гілок}-\text{загальне}} = X_{\bar{j}}^{j}\sum_{q=1}^{N}\sum_{p=1}^{K}\rho_{qp}, \tag{2.22}$$

де ρ_{qp} – вектор вузлових навантажень, створюваних p-м сайтом q-ої мережі.

2.5 Накладені обмеження на якість обслуговування

Представлені вище результуючі системи рівнянь можуть мати нескінченну кількість рішень. Це викликано тим фактом, що маршрутів пересування інформації між потоками може бути безліч, а рішенням систем рівнянь (2.6), (2.12) та (2.22) являється визначення одного з таких маршрутів. Існує два основних способи використання цих систем рівнянь:

- завдяки тому, що потоки в гілках систем виражені у вигляді лінійно незалежних завантажень, після визначення завантаження/потоків у лінійно незалежних гілках, також автоматично будуть визначені завантаження/потоки в усіх інших;

- замість підбора значень лінійно незалежних компонент, можно знайти будь-яке рішення, що буде описувати можливі маршрути між потоками. Але в такому варіанті необхідним для вирішення системи рівнянь являється використання обмежень, які представляють собою системи нерівностей.

Серед обов’язкових обмежень для вирішення систем рівнянь другим способом можно виділити такі:

- величина сумарного потоку в каналі зв'язку завжди повинна бути менша ніж величина пропускної здатності каналу;

- необхідно забезпечити невід’ємність потоків, які створені каналом q-ї мережі;

- необхідно запобігти появу петльових маршрутів.

Як додаткові обмеження для системи рівнянь можно виділити обмеження наскрізної затримки та ймовірності втрат даних у віртуальній мережі.

Таким чином, враховуючи обов’язкові накладені обмеження на якість обслуговування потоків мережі та значення матриць запитів, після вирішення

результуючих систем рівнянь маршрути трафіку всередині мережі можуть бути явно визначені.

2.6 Аналіз мережі на практиці

Проведемо аналіз мережі зображеної на рисунку 2.2. Для цього прикладу скористаємося ортогональним методом аналізу мереж, як цікавим варіантом, в якому перетворення над вихідною мережею не проводяться.

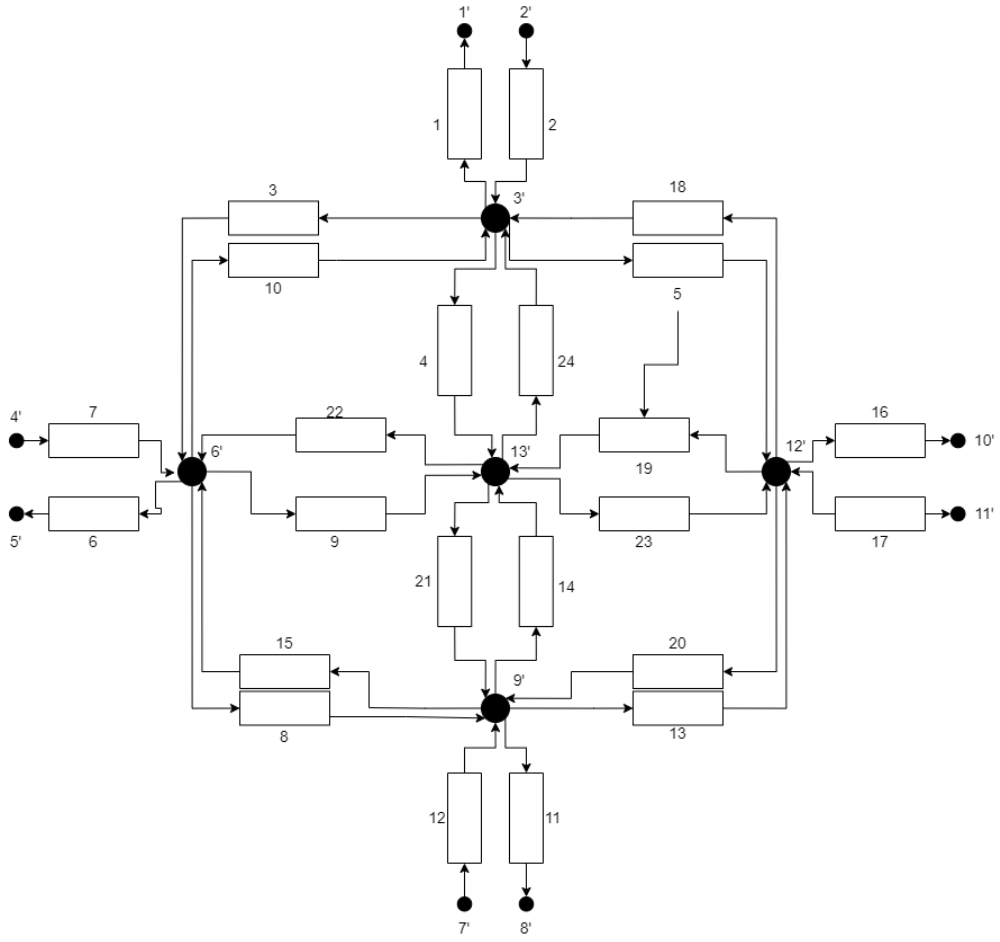

Рисунок 2.2 – Схема досліджуваної мережі

Відповідно до ортогонального методу аналізу визначимо тензор перетворення від структури досліджуваної мережі до структури примітивної мережі, який пов'язує вузлові та контурні інтенсивності досліджуваної

мережі з вузловими та контурними інтенсивностями примітивної мережі через формулу (2.16).

Потім за формулами (2.19) та (2.20) визначимо фазові навантаження та фазові інтенсивності обслуговування. Після чого за формулою (2.21) знайдемо навантаження у кожній гілці. Далі за формулою (2.22) обчислимо навантаження у кожній з гілок загалом.

Вхідними даними для аналізу будуть виступати матриці запитів:

$$X_\alpha = \begin{bmatrix} 0 & 5 & 6 \\ 7 & 0 & 3 \\ 1 & 2 & 0 \end{bmatrix}, \quad (2.23)$$

$$X_\beta = \begin{bmatrix} 0 & 8 & 6 \\ 10 & 0 & 3 \\ 15 & 9 & 0 \end{bmatrix}, \quad (2.24)$$

$$X_\gamma = \begin{bmatrix} 0 & 9 \\ 10 & 0 \end{bmatrix}, \quad (2.25)$$

Час обслуговування однієї одиниці інформації прийнято вважати за 0,01 с або 100 пакетів на секунду. Одним із рішень системи рівнянь (2.22) з урахуванням обмежень, що значення елементів векторів і повинні знаходитися в діапазоні [0;1), відповідає наступний розподіл навантажень по гілках:

$$\rho_{1\alpha} = [0,06 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0,05 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0,11 \;\; 0,06 \;\; 0 \;\; 0,05 \;\; 0 \;\; 0 \;\; 0 \;\; 0]$$
$$\rho_{2\alpha} = [0,03 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0,1 \;\; 0,07 \;\; 0,03 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0,03]$$
$$\rho_{3\alpha} = [0 \;\; 0,03 \;\; 0 \;\; 0,02 \;\; 0,01 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0,02 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0,01 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0,02 \;\; 0 \;\; 0 \;\; 0], \quad (2.25)$$

$$\rho_{1\beta} = [0,06 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0,08 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0,14 \;\; 0,06 \;\; 0,08 \;\; 0 \;\; 0 \;\; 0,08 \;\; 0 \;\; 0]$$
$$\rho_{2\beta} = [0,03 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0,13 \;\; 0 \;\; 0,1 \;\; 0,03 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0,1 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0,1 \;\; 0]$$
$$\rho_{3\beta} = [0 \;\; 0,24 \;\; 0,15 \;\; 0 \;\; 0,09 \;\; 0,15 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0,09 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0], \quad (2.26)$$

$$\rho_{1\gamma} = [0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0,1 \;\; 0,1 \;\; 0 \;\; 0 \;\; 0,1 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0]$$
$$\rho_{2\gamma} = [0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0,09 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0,09 \;\; 0 \;\; 0 \;\; 0,09 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0], \quad (2.27)$$

Кожен вектор відображає навантаження на гілках, створене р-м сайтом q-ої мережі і при цьому, загальний вектор навантажень для кожної гілки буде таким:

$$\rho_{\text{гілок–загальне}} = [0,18 \;\; 0,27 \;\; 0,15 \;\; 0,02 \;\; 0,1 \;\; 0,32 \;\; 0,23 \;\; 0,1 \;\; 0,1 \;\; 0,03 \;\; 0,17 \;\; 0,19$$
$$0,07 \;\; 0,03 \;\; 0,09 \;\; 0,27 \;\; 0,25 \;\; 0,12 \;\; 0,08 \;\; 0,05 \;\; 0,02 \;\; 0,8 \;\; 0,1 \;\; 0,03], \quad (2.28)$$

Таким чином, було отримано розподіл трафіку каналами зв'язку за допомогою ортогонального методу, що задовольняє поставлену задачу. Також слід відзначити, що це лише одне рішення із безлічі можливих.

Отже, у даному підрозділі було проаналізовано вузловий, контурний та ортогональний методи дослідження деградуючих полінгових мереж. Оскільки мережева структура багатьох інфраструктур побудована на основі полінгових мереж, провайдери таких послуг повинні надавати гарантії якості обслуговування для трафіку кожної мережі. Розгалужена мережева інфраструктура провайдера робить завдання керування трафіком досить рутинним, а аналітичні методи аналізу, у свою чергу, добре описують розподіл трафіку у великих системах завдяки добре сформульованому методу матричної математики, що залишається незмінним, незалежно від розміру досліджуваної топології.

Також слід зазначити, що використовувані арифметичні операції легко піддаються розпаралелюванню і це значно скорочує час аналізу мереж в багатопроцесорних системах. Серед переваг такого підходу для аналізу

можливості організації маршрутизації в конкретній області мережі можно визначити простоту визначення математичної моделі мережі та лінійне зростання складності обчислення маршрутів, що залежить від збільшення числа каналів в мережі. В той же час, складність знаходження конкретного маршруту зростає, через накладання обмежень на відсутність маршрутних петель для кожного маршруту в мережі. Ще однією перевагою є те, що завдяки цим методам можно зменшити витрати часу на обчислення, сформулювавши одразу декілька вимог до матриці. Цьому сприяє властивість лінійно незалежних циклів визначати вид тензора перетворень контурних інтенсивностей.

3 ВИКОРИСТАННЯ МОДИФІКОВАНОГО КВЕСТ-МЕТОДУ ДЛЯ АНАЛІЗУ ДЕГРАДУЮЧИХ ПОЛІНГОВИХ МЕРЕЖ

Для реалізації полінгових мереж зазвичай використовується платформа MANET (Mobile Ad hoc Networks). Основним призначенням подібних мереж – є організація безпечного зв'язку між рухомими об'єктами: автомобілями, людьми, водним чи залізничним транспортом.

Неможливість передачі трафіку у реальному часі на великій швидкості в таких мережах пов’язана з особливістю роботи мережевого і канального рівнів, що зумовлені сильним впливом формального уявлення топологічних змін MANET-систем, які використовуються в програмних комплексах, необхідних для дослідження деградуючих полінгових мереж. Теорія випадкових графів стала головним математичним інструментом для представлення змін у топології полінгових мереж. Згідно з нею просування вузлів мережі виражається у формі броунівського руху, в той час як детермінованість їх траєкторії опускається, що являється некоректним, якщо розглядається в ролі мережевих вузлів об'єктів транспорту [6].

Вважаючи вищевикладене, мета дослідження – скорегувати формальну інтерпретацію топологічних змін, розробити математичну модель процесу оцінки якості роботи таких мереж та розробити модифікований квест-метод аналізу подібних полінгових мереж.

Як основу для формального представлення топологічних змін мережі можно використати модель «мерехтливого графа». Суть цієї моделі полягає в тому, щоб подати динамічну топологію мережі у вигляді еволюціонуючого графу, що змінює свій стан дискретно:

$$\{G(\Delta t)\} = G(\Delta t_0) \to G(\Delta t_1) \to ...G(\Delta t_k), \qquad (3.1)$$

де $G(\Delta t)$ – граф, що описує топологію на протязі часу .

Різниця між графом мережі в стані *(k+1)* і графа в стані k відображається у прирості *ΔG*, який показує кількість вилучених або доданих ребер до топології при переході зі стану k в стан *(k+1)*:

$$G(\Delta t_{k+1}) \to G(\Delta t_k) \cup \Delta G., \tag{3.2}$$

У розглянутому представленні, на відміну від відомих моделей з дискретним часом, передбачається, що Δt≠const. Завдяки цьому його можно поширити на деградуючі полінгові мережі, в яких період повторення топологічних станів відсутній. Узагальнення виразів (3.1), (3.2) дозволяє на концептуальному рівні представити процес трансформації топологічних змін у просторі (коли зміна взаємного розташування вузлів призводить до зміни топології мережі) у топологічні зміни в часі (коли зміна топології мережі визвана відкриттям чи закриттям каналу зв'язку між вузлами).

Значення, що визначають час «відкриття» і «закриття» каналів зв'язку між вузлами мережі розраховуються відповідно до руху вузла і параметрів пристрою. Аналізуючи час «відкриття» і «закриття» каналів зв'язку між вузлами, можна судити про топологічний зв'язок в комп'ютерній системі під час руху вузла мережі, щоб прийняти рішення щодо базової продуктивності мережі. Це рішення дає змогу відсіяти навмисно нездійсненні топології під час процесу автоматизованого проектування.

Загалом, процес оцінки продуктивності спроектованої мережі має здійснюватися у два етапи: на рівні імітаційного моделювання та на рівні аналітичного моделювання. На основі результатів дослідження робиться висновок про непридатність чи придатність заданої топології для надання заданої послуги.

Функція (3.3) представляє процес такої оцінки:

$$A(t) = \Psi[f(\{x_i\},\{h\},t),\xi(\{v_i\},t)], \quad (3.3)$$

де функція $f(\{x_i\},\{h\},t)$ – це аналітичний компонент моделі, за допомогою якого можна визначити час взаємодії вузлів мережі;

$\{x_i\}$ – параметри рухомого і (траєкторія і швидкість руху, параметри обладнання, які відповідають за можливу дальність встановлення з'єднання);

$\{h\}$ – параметри статичних вузлів;

$\xi(\{v_i\},t)$ – імітаційна складова моделі;

$\{v_i\}$ – імовірнісні характеристики мережі(такі як трафікове навантаження або алгоритми роботи мережевих протоколів).

Результат аналітичної складової моделі – масив значень часу «відкриття» і «закриття» каналів між вузлами-учасниками полінгової мережі:

$$f(\{x_i\},\{h\},t) = \begin{bmatrix} \left\{T_{1,1}^{відк}\right\},\left\{T_{1,1}^{закр}\right\} & \dots & \left\{T_{1,m}^{відк}\right\},\left\{T_{1,m}^{закр}\right\} \\ \dots & \dots & \dots \\ \left\{T_{n,1}^{відк}\right\},\left\{T_{n,1}^{закр}\right\} & \dots & \left\{T_{n,m}^{відк}\right\},\left\{T_{n,m}^{закр}\right\} \end{bmatrix}, \quad (3.4)$$

де $\left\{T_{n,m}^{відк}\right\}$ – масив значень часу «відкриття» каналу між заданими вузлами;

$\left\{T_{n,m}^{закр}\right\}$ – масив значень часу «закриття» каналу між заданими вузлами.

Елементи множини $\left\{T_{n,m}^{відк}\right\}$, $\left\{T_{n,m}^{закр}\right\}$ визначаються за допомогою математичної моделі руху вузлів мережі та розміру зони покриття кожного вузла. Суть цього прийому зводиться до того, що на основі даних про розміри зони покриття і про умови руху вузлів, формулюючи рівняння, розв’язками яких є комплексні числа у вигляді $t = a + ib$, можна судити про тип взаємодій вузлів за їх зовнішнім виглядом. Основні випадки можно сформулювати так:

- при $a \leq 0$ і $b = 0$ взаємодія відбулась на момент початку експерименту;

- при $b \neq 0$ взаємодія між вузлами не відбулася(з точки зору теорії графів, це означає, що в топології немає шляхів між заданими вузлами);

- при $a \geq 0$, $b = 0$ та $a_n < a_{n+1}$ маємо a_n – час початку взаємодії і a_{n+1} – час закінчення взаємодії двох вузлів.

Аналізуючи значення матриці у виразі (3.4) можна судити про зв'язки топології мережі (графа) при перенесенні вузлів із зони впливу один одного. Умова, за якою можно оцінити пов'язаність топології мережі, у спрощеному вигляді така:

$$t_{закр_n} > t_{откр_{n+1}}, \tag{3.5}$$

Згідно з цією умовою час припинення взаємодії з поточним вузлом має наступити пізніше, ніж почнеться взаємодія з наступним вузлом. Вона забезпечує зв'язок мережі з динамічною топологією. Визначення різниці $t_{закр_n} - t_{откр_{n+1}}$ (достойність) залежить від швидкості роботи керуючих систем комп'ютерного обладнання, що знаходиться на вузлі, яке визначається із співвідношення:

$$t_{кер} = t_{дост} + t_{пку}, \tag{3.6}$$

де $t_{дост}$ – час, потрібний для достовірної оцінки рівня сигналу, необхідного для передачі трафіку по каналу між рухомими вузлами, враховуючи параметрами якості;

$t_{пку}\, t_{пку}$ – час, потрібний обладнанню щоб ухвалити рішення про «відкриття» нового каналу.

На фізичному рівні процес передачі мобільного вузла між сусідніми

мобільними або фіксованими вузлами загалом подібний до процесу естафетної передачі в існуючих системах мобільного зв’язку [7].

Рішення про передачу вузлів приймається або в центрі управління(при централізованому управлінні), або на самому вузлі(при децентралізованому управлінні), в залежності від способу управління мережею. Рівень потужності($P_{пор}$) найбільш доречно використовувати в якості основного контрольованого параметра, на підставі значення якого буде прийматися рішення про «відкриття» або «закриття» каналу з іншими вузлами. Досягнення певного порогового значення $P_{пор}$ буде включати процедуру попередження, а при досягненні певного допустимого значення $P_{пор}$ буде прийматися рішення про зупинку передачі даних(що призведе до втрати пов’язаності топології) або про перехід на роботу з іншим вузлом мережі.

В цілому виразами (3.1)-(3.3) допускається відсутність детермінованості топологічних змін, тоу процес зміни каналу доречно представити у вигляді ланцюга Маркова (рисунок 3.1).

На рисунку 3.1 використовуються такі позначення:

U_1 – оптимальний рівень прийому на (– парне число,);

U_2 – сигнал про кількості даних, які передаються між вузлами і наближення до значення ;

U_3 – «закриття» каналу зв'язку між парою вузлів;

U_4 – процедура зміни сусідньої мережі;

U_5 – наближення до порогового значення $P_{пор}$ на V_{j+1} ($V_{j+3},...V_{j+n}, n$ – парне число, $n \neq \infty$).

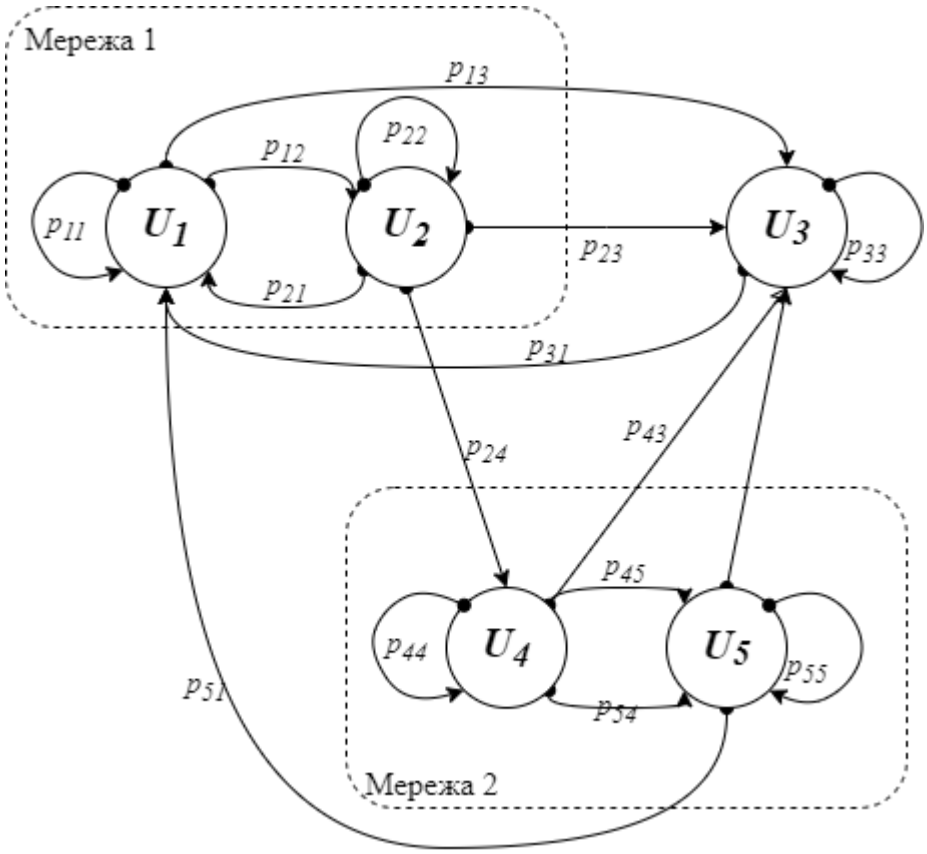

Рисунок 3.1 – Алгоритм подій зміни каналу між мережами

Виходячи з того факту, що програми, які змінюють стан системи подій, є несумісними, а також на теоретичних положеннях, умови переходу станів системи можно описати наступним масивом формул (3.7)...(3.13).

Масив формул (3.7)...(3.13) описує процес прийняття рішення про «закриття» або «відкриття» каналів зв'язку між рухомими вузлами мережі, що потрібно при написанні алгоритмів роботи програмних комплексів по моделюванню полінгових мереж (для підвищення достовірності результатів моделювання, завдяки обліку умов розповсюдження даних) і при створенні систем автоматизованого управління деградуючими полінговими мережами.

$$p_{11} = \begin{cases} 1, P_{\text{тек}}^{i+m} > P_{\text{пор}} \\ 0 \end{cases}, \; p_{12} = \begin{cases} 1, P_{\text{пор}} > P_{\text{тек}}^{i+m} > P_{\text{доп}} \\ 0 \end{cases}, \; p_{13} = \begin{cases} 1, P_{\text{тек}}^{i+m} < P_{\text{доп}} \\ 0 \end{cases}; \quad , \text{(3.7)}$$

$$p_{22}=\begin{cases}1, P_{\text{тек}}^{i+m}=P_{\text{доп}}\\ 0\end{cases},\ p_{21}=\begin{cases}1, P_{\text{тек}}^{i+m}\geq P_{\text{пор}}\\ 0\end{cases},\ p_{23}=\begin{cases}1, P_{\text{тек}}^{i+m}<P_{\text{пор}}\\ 0\end{cases}, \quad (3.8)$$

$$p_{24}=\begin{cases}1, P_{\text{тек}}^{i+n}>P_{\text{пор}}\\ 0\end{cases}, \quad (3.9)$$

$$p_{33}=\begin{cases}1, P_{\text{тек}}^{\forall i}<P_{\text{пор}}\\ 0\end{cases}, \quad (3.10)$$

$$p_{43}=\begin{cases}1, P_{\text{тек}}^{i+n}<P_{\text{доп}}\\ 0\end{cases},\ p_{44}=\begin{cases}1, P_{\text{тек}}^{i+n}\approx P_{\text{тек}}^{i+m}\approx P_{\text{доп}}\\ 0\end{cases},\ p_{45}=\begin{cases}1, P_{\text{пор}}>P_{\text{тек}}^{i+m}\geq P_{\text{доп}}\\ 0\end{cases}, \quad (3.11)$$

$$p_{51}=\begin{cases}1, P_{\text{тек}}^{i+m}>P_{\text{доп}}\\ 0\end{cases},\ p_{53}=\begin{cases}1, P_{\text{тек}}^{i+m}<P_{\text{доп}}\\ 0\end{cases},\ p_{54}=\begin{cases}1, P_{\text{тек}}^{i+m}>P_{\text{пор}}\\ 0\end{cases}, \quad (3.12)$$

$$p_{55}=\begin{cases}1, P_{\text{тек}}^{i+m}<P_{\text{пор}}\\ 0\end{cases}, \quad (3.13)$$

Якщо результати перевірки топології мережі на зв’язність, згідно формул (3.5) і (3.6), виявилися позитивними, то проводиться дослідження роботи полінгових мереж для надання необхідного спектру сервісів. Результатом дослідження являється масив параметрів оцінки якості мультисервісного трафіку, які далі порівнюються з нормованими

значеннями, що можуть надаватися такими концепціями, яу QoS (Quality of Service). Цей процес можно представити у вигляді:

$$A(t) = \begin{cases} \{\chi\} \subset \{X\}, 1, \\ \{\chi\} \supset \{X\}, 0, \end{cases}, \tag{3.14}$$

де $\{\chi\}$ – масив показників роботи полінгових мереж, які були отримані в ході імітаційного моделювання;

$\{X\}$ – масив нормованих параметрів якості роботи системи.

Топологія системи є придатною для надання номенклатури сервісів, якщо *A(t)=1*. В іншому випадку – *A(t)=0*.

Алгоритмічно процес роботи програмного комплексу, описаний виразами (3.3)...(3.14), представлений на рисунку 3.2.

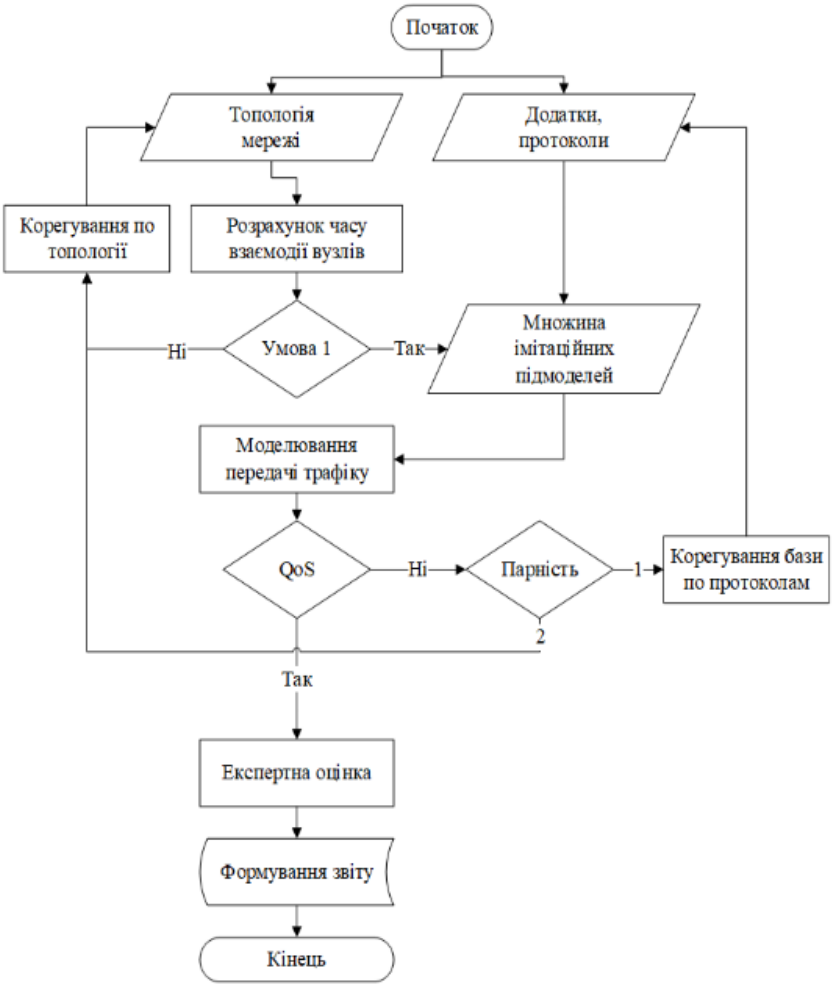

Рисунок 3.2 – Алгоритм роботи програмного комплексу за модифікованим квест-методом аналізу полінгових систем

Внесення правок у формальну інтерпретацію топологічних змін дозволило перевести процес трансформації топологічних змін у просторі (коли зміна взаємного розташування вузлів призводить до зміни топології мережі) у топологічні зміни в часі (коли зміна топології мережі визвана відкриттям чи закриттям каналу зв'язку між вузлами), що дозволило розглянути статичні і динамічні компоненти загальної топології мережі разом.

Завдяки розробленій математичній моделі процесу оцінки якості роботи полінгових мереж з урахуванням проведеного внесення правок можно проводити дослідження топології мережі в такій послідовності: спочатку

перевіряється зв'язаність топології мережі, після чого досліджується можливість передачі мультисервісного трафіку. На основі результатів дослідження робиться висновок про непридатність чи придатність мережі для надання тієї чи іншої послуги.

Алгоритмічний процес програмного комплексу на основі розробленої моделі показує, як вибирається полінгова мережа, реалізація якої є найбільш прийнятною в заданих умовах.

Отже, модифікований квест-метод аналізу деградуючих полінгових мереж надає можливість розробляти комплекси програм для моделювання деградуючих полінгових мереж.

ВИСНОВКИ

В результаті виконання кваліфікаційної роботи були досліджені різні методи аналізу деградуючих полінгових мереж, а також проаналізовані їх області використання, разом перевагами та недоліками використання того чи іншого методу. Тензорний метод дозволяє легко формалізувати процедури, необхідні для проектування полінгових мереж. А контурний, вузловий та ортогональний методи аналізу мереж дозволяють легко проаналізувати можливість організації маршрутів трафіку в конкретній області деградуючої полінгової мережі. В результаті теоретичної роботи було успішно проаналізовано мережу за допомогою ортогонального методу і отримано розподіл трафіку каналами зв'язку, що дозволяє визначити чи підходить мережа до поставленої задачі.

У той же час модифікований квест-метод аналізу мереж об'єднує в собі практично всі існуючі послуги мереж, щоб забезпечити заданий рівень якості передачі трафіку, а програмна реалізація квест-методу полегшує оцінку необхідних показників якості при вказаних обчислювальних витратах.

Дослідження в рамках виконання кваліфікаційної роботи проводилися на базі системи віртуалізації навчально-наукової лабораторії мікроконтролерних систем кафедри електронно-обчислювальних машин Харківського національного університету радіоелектроніки.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. V. Tkachov, M. Hunko, M. Bondarenko, S. Artyomov Technology of Load Balancing in Anonymous Network Based on Proxy Nodes Cascade Platform / Міжнародна науково-технічна конференція "Комп'ютерні та інформаційні системи і технології". – Харків. – С. 82.

2. "An OSI Model for Cloud". Cisco Blogs. 24 February 2017. Retrieved 16 May 2020.

3. T.A. Tarray and M.R. Bhat. A nonlinear programming problem using branch and bound method, Investigación Operacional, vol. 38, No. 3, pp. 291-298, January 2018.

4. Andrew L. Russell (30 July 2013). "OSI: The Internet That Wasn't". IEEE Spectrum. Vol. 50, no. 8.

5. Ткачов В.М. Метод передачі даних в комп’ютерній мережі проміжного зберігання даних складної інформаційної системи / В.М. Ткачов // Системи управління, навігації та зв’язку. – Полтава: Полтавський національний технічний університет ім. Ю. Кондратюка, 2017. – № 3 (43). – С. 117-119.

6. S. Zhang, C. Zhang and Q.Yang. Data preparation for data mining, Applied artificial intelligence, Vol. 17, No. 5-6, pp. 375-381, November 2003.

7. Abbate, Janet (2000). Inventing the Internet. MIT Press. pp. 123–4.