Національний університет “Львівська політехніка”

На правах рукопису

Селюченко Мар’ян Олександрович

УДК 621.391

Моделі та алгоритми підвищення якості обслуговування у телекомунікаційних програмно-конфігурованих мережах

05.12.02 – *телекомунікаційні системи та мережі*

Дисертація на здобуття наукового ступеня
кандидата технічних наук

Науковий керівник -
доктор технічних наук,
професор **Климаш М.М.**

Ідентичність всіх примірників дисертації
ЗАСВІДЧУЮ:
Вчений секретар спеціалізованої
вченої ради **/І.В.Демидов/**

Львів – 2016

ЗМІСТ

стор.

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

API – Application Programming Interface, формалізований набір визначень для взаємодії програмного забезпечення.

CSP – Constrained Shortest Path Problem, розширення проблеми знаходження найкоротших шляхів, на які накладаються певні обмеження.

DPID – Datapath ID, ідентифікатор OpenFlow комутатора, що використовується контролером.

DSCP – Differentiated Services Code Point, поле коду диференційованої послуги.

EIGRP – Enhanced Interior Gateway Routing Protocol, закритий протокол маршрутизації.

IP – Internet Protocol, технологічна основа побудови транспортної мережі з комутацією пакетів.

ITU – International Telecommunication Union, Міжнародний союз телекомунікацій.

LLDP – Link Layer Discovery Protocol, протокол для обміну інформацією про конфігурацію між мережевими пристроями.

MAC – Media Access Control, адреса мережевої карти.

MCFP – Multi-Commodity Flow Problem, проблема прокладання декількох потоків у мережі між різними вхідними та вихідними вузлами.

NFV – Network Functions Virtualization, віртуалізація мережних функцій.

OSPF – Open Shortest Path First, протокол динамічної маршрутизації на основі стану каналу для знаходження найкоротшого шляху.

QoE – Quality of Experience, оцінка досвіду використання користувачем певної послуги.

QoS – Quality of Service, якість обслуговування, що включає в себе такі параметри: затримка, джиттер та втрати пакетів.

ToS – Рівень пріоритету IP, вид послуги.

TTL – Time to live, кількість вузлів, через які може пройти пакет, після чого він буде видалений.

VoD – Video on Demand, відео на замовлення.

VoIP – Voice over IP, телефонія на основі протоколу IP.

МОС – Мережева операційна система.

ПКМ – Програмно-конфігурована мережа.

ВСТУП

Актуальність теми. Телекомунікаційні мережі стають все більш динамічними, а системи управління – складнішими та інтелектуальнішими. Програмно-конфігурована мережа (ПКМ) – телекомунікаційна мережа, в якій рівень керування представлений програмним контролером і є відділеним від рівня передавання даних. Технологія ПКМ забезпечує високу гнучкість управління та суттєво спрощує віртуалізацію мережних ресурсів. Можливість динамічної конфігурації мережі з допомогою контролера без зміни апаратного чи програмного забезпечення мережних пристроїв зумовила те, що сьогодні більшість операторів телекомунікаційних мереж впроваджують цю технологію. Разом з тим, алгоритми управління трафіком та методи передавання даних залишаються незмінними, а отже, питання забезпечення якості обслуговування трафіку згідно з вимогами користувачів та ефективності використання мережних ресурсів не втрачають актуальності.

Задачі маршрутизації та управління інформаційними потоками для підвищення якості обслуговування в мультисервісних програмно-конфігурованих мережах досліджували провідні вітчизняні та зарубіжні вчені, зокрема такі: Лемешко О.В., Євсеєва О.Ю, Ложковський А.Г., Беркман Л.Н., Безрук В.М., Климаш М.М, Szu-Yuan Chen, Hongyan Qian, Wolfgang Kellerer, H. Hasan, Ken-Ichi Suzuki, Muhammad Aziz Muslim, Alexander Gelberger.

Незважаючи на значну кількість розроблених та впроваджених технічних рішень управління процесами передавання даних, невирішеними досі залишаються задачі ефективної маршрутизації потоків для забезпечення вимог якості обслуговування мультисервісного трафіку в умовах перевантажених каналів та обмежених мережних ресурсів. Основною причиною цього є відсутність засобів контролю за процесом передавання окремих потоків, внаслідок чого система управління не має змоги зафіксувати погіршення якості обслуговування для цих потоків, а тому не може гарантувати рівень

обслуговування, узгоджений у сервісному договорі SLA (Service Level Agreement).

Відсутність можливості здійснювати диференційоване управління окремими потоками певних користувачів та враховувати вимоги кожного потоку до параметрів якості обслуговування призводить до низької ефективності маршрутизації, неоптимального розподілу навантаження та погіршення якості обслуговування потоків реального часу. Таким чином, в умовах постійного зростання обсягів трафіку та кількості користувачів сервісів потокового контенту актуальним є наукове завдання розроблення методів та моделей управління процесами передавання даних у телекомунікаційних програмно-конфігурованих мережах для підвищення якості обслуговування користувачів та ефективного використання мережних ресурсів.

Зв'язок роботи з науковими програмами, планами, темами. Тематика дисертаційної роботи безпосередньо пов'язана з науковим напрямом кафедри телекомунікацій «Інфокомунікаційні системи та мережі», з положеннями «Концепції національної інформаційної політики», «Стратегії розвитку інформаційного суспільства в Україні», Закону України «Про основні засади розвитку інформаційного суспільства в Україні на 2007-2015 роки».

Дисертаційну роботу виконано в рамках держбюджетної науково-дослідної роботи "Моделі та структури конвергентних телекомунікаційних мереж на основі CLOUD – технологій" ("ДБ/CLOUD"), (2013-2014 рр.), номер держреєстрації 0113U003184, держбюджетної науково-дослідної теми «Методи побудови та моделі інформаційно – телекомунікаційної інфраструктури на основі SDN – технологій для систем електронного урядування" ("ДБ/SDN"), (2015-2016 рр.), номер держреєстрації 0115U000444.

Мета дисертаційної роботи полягає у підвищенні якості обслуговування в програмно-конфігурованих мережах шляхом зниження затримки і втрат пакетів та підвищення рівномірності завантаження мережних каналів.

Для досягнення визначеної мети поставлено такі **наукові завдання:**

1. Аналіз принципів побудови програмно-конфігурованих мереж та моделей управління процесом передавання даних.

2. Розроблення способу ідентифікації потоків з урахуванням класу трафіку та вимог користувача щодо параметрів якості обслуговування.

3. Підвищення точності вимірювання затримки пакетів для окремих потоків та шляхів передавання даних.

4. Підвищення точності оцінки стану мережевих ресурсів та характеризування процесів передавання даних.

5. Зменшення середньої затримки пакетів при обслуговуванні потоків, чутливих до флуктуацій часових параметрів передавання даних.

6. Розроблення рекомендацій щодо інтеграції запропонованих технічних рішень у систему управління програмно-конфігурованими мережами.

Об'єкт дослідження – процес обслуговування інформаційних потоків у програмно-конфігурованих мережах.

Предмет дослідження – моделі та алгоритми обслуговування інформаційних потоків у програмно-конфігурованих мережах.

Методи дослідження. Для розв'язання поставлених завдань у роботі використано методи лінійного програмування, теорії ймовірностей та математичної статистики, аналітичного та імітаційного моделювання, методи планування експерименту.

Наукова новизна роботи полягає у тому, що:

1. *Вперше запропоновано модель* адаптації системи моніторингу програмно-конфігурованої мережі, зокрема частоти опитування стану мережних ресурсів, до завантаження вузлів та каналів, що дало змогу підвищити адекватність оцінки характеристик процесу передавання даних.

2. *Набув подальшого розвитку метод* вимірювання затримки передавання пакетів окремого потоку, у якому, на відміну від існуючих, у площину передавання даних вводиться тестовий пакет із заголовком, ідентичним до пакетів вимірюваного потоку, що дає змогу оцінити рівень забезпеченої якості

обслуговування відповідно до вимог окремих користувачів та гарантій оператора мережі.

3. *Удосконалено модель* балансування навантаження, у якій, на відміну від існуючих, критерієм переспрямування пакетів вибрано допустимий рівень завантаження каналу, що дало змогу зменшити імовірність втрат, підвищити ефективність використання пропускної здатності каналу та пропорційно розподілити трафік в мережі.

4. *Удосконалено модель* маршрутизації потоків, яка, на відміну від відомих, використовує значення відносного пріоритету для ідентифікації потоку, що дало змогу розв'язати оптимізаційну задачу управління потоками за критеріями якості обслуговування та рівномірного використання мережних каналів.

Практичне значення одержаних результатів полягає в тому, що:

1. Розвинуто метод вимірювання затримки, який дає змогу підвищити точність вимірювання затримки передавання пакетів уздовж вибраного шляху до 2,5 разів, якщо завантаження цього шляху наближається до 100%.

2. Запропоновано модель адаптації системи моніторингу, яка дає змогу підвищити точність оцінки завантаження мережних каналів не менше, ніж на 10% залежно від статистичних характеристик мультисервісного трафіку.

3. Розроблено алгоритм перерозподілу потоків, який дає змогу уникнути перевантаження інтерфейсу комутатора та покращити параметри якості обслуговування пріоритетного трафіку.

4. Удосконалена модель маршрутизації дає змогу зменшити середню затримку потоків реального часу на 15% та підвищити рівномірність завантаження каналів на 30% порівняно з протоколом EIGRP.

5. Удосконалена модель балансування навантаження, за умов середнього завантаження каналів 90% та середньоквадратичного відхилення інтенсивності вхідного навантаження 10%, дає змогу знизити втрати пакетів у середньому в 6 разів.

Наукові та практичні результати виконаних досліджень використано в навчальному процесі, в лекційних курсах і лабораторних роботах, які

проводяться для студентів кафедри "Телекомунікації" Національного університету "Львівська політехніка" за напрямом "Телекомунікації" та спеціальністю "Інформаційні мережі зв'язку", зокрема в дисципліні "Розподілені сервісні системи та Cloud-технології".

Результати роботи використано для покращення якості обслуговування телекомунікаційних мереж у ПАТ «Укртелеком», ТОВ "Літех", ПП "Цифрові технології", що підтверджено актами впровадження.

Апробація результатів дисертації. Основні наукові результати і положення дисертації представлені, доповідались та всебічно обговорювалися на 18-ти міжнародних та всеукраїнських науково-технічних конференціях, наукових семінарах та симпозіумах: CADSM (Поляна-Свалява, 2013, 2015 рр.); Науково-технічній конференції "Проблеми телекомунікацій" (м. Київ, 2014, 2015, 2016 рр.); Problems of Infocommunications, Science and Technology (PICS&T) (м. Харків, ХНУРЕ, 2014, 2015 рр.); науково-практичній та науково-методичній конференції "Сучасні проблеми телекомунікацій і підготовка фахівців в галузі телекомунікацій" (м. Львів, 2012, 2013, 2014, 2015 рр.); 4-ій Міжнародній науково-практичній конференції, присвяченій 25-річчю заснування кафедри "Радіотехніки та інформаційної безпеки" Чернівецького національного університету ім. Юрія Федьковича, (м. Чернівці, 2014р.); VI-му відкритому науковому семінарі ПММ-ТКС 2015 (м. Полтава, 2015р.). Крім цього, дисертаційну роботу представлено на науковому семінарі кафедри телекомунікацій Національного університету "Львівська політехніка".

Публікації. За результатами досліджень, які викладено у дисертаційній роботі, опубліковано 35 наукових праць, з них: 3 статті в іноземних наукових фахових виданнях, що включені до міжнародних наукометричних баз даних [1, 2, 6]; 5 статей у наукових фахових виданнях України, які включені до міжнародних наукометричних баз даних [3-5, 7-8]; 3 статті у наукових фахових виданнях України згідно переліку МОН та 24 публікації у збірниках праць міжнародних і всеукраїнських конференцій [9-11].

Особистий внесок здобувача. Усі результати досліджень, викладені в дисертації, одержано автором особисто. У працях, опублікованих у співавторстві, дисертантові належать: [3, 5, 8, 11, 14, 15, 17, 23, 30, 33, 35] – удосконалена модель балансування навантаження для підвищення ефективності використання каналів та зниження рівня втрат пакетів; [4, 8, 16, 18, 29, 30] – тестова платформа програмно-конфігурованої телекомунікаційної мережі; [2, 4, 5, 13, 15] – удосконалена методика розрахунку відносного пріоритету потоку для ідентифікації трафіку у вузлах мультисервісної мережі; [1-6, 14-17, 23-27] – система генерації мультисервісного трафіку для оцінювання стану мережних ресурсів; [8, 12, 18] – система моніторингу для програмно-конфігурованих мереж та модель адаптації системи моніторингу до характеру процесів передавання даних.

Структура та обсяг роботи. Робота складається зі вступу, чотирьох розділів, висновків, списку використаних джерел та двох додатків. Загальний обсяг роботи складає 156 сторінок друкарського тексту, із них: 6 сторінок вступу, 120 сторінок основного тексту, 60 рисунків, 20 таблиць, список використаних джерел зі 107 найменувань, додаток на 4 сторінках.

РОЗДІЛ 1. АНАЛІЗ МОДЕЛЕЙ ТА АЛГОРИТМІВ ФУНКЦІОНУВАННЯ ПРОГРАМНО-КОНФІГУРОВАНИХ ТЕЛЕКОМУНІКАЦІЙНИХ МЕРЕЖ

У першому розділі розглянуто архітектуру та основні принципи побудови програмно-конфігурованих мереж та комутації потоків. Проведено детальний аналіз методів управління потоками в комутаторах з апаратною реалізацією OpenFlow функціональності та способів підвищення якості обслуговування мультисервісного трафіку. Виділено найбільш поширені методи та засоби моніторингу параметрів функціонування мережевих пристроїв та якості обслуговування мультисервісних потоків. Встановлено, що більшість існуючих моделей управління потоками використовують базові засоби OpenFlow для балансування навантаження та не здатні проводити оперативну адаптацію мережі до стрибків інтенсивності мультисервісного трафіку. Особливий акцент зроблено на методах маршрутизації потоків, які зазвичай враховують тільки один з критеріїв оптимальності, наприклад, один з параметрів якості обслуговування чи коефіцієнт використання мережевих ресурсів. Встановлено, що жоден з проаналізованих методів не дає змогу оптимізувати мережу, опираючись на параметри якості обслуговування індивідуального клієнта, а також не забезпечує оперативну та гнучку адаптацію мережі до характеристик трафіку. Аргументовано, що розробка методів та моделей управління потоками з оперативним урахуванням стану мережі для забезпечення необхідних параметрів якості обслуговування кожного клієнта є актуальною науковою задачею.

1.1. Архітектура програмно-керованих мереж

Програмно-керована мережа (ПКМ) – мережа передачі даних, в якій рівень керування мережею відділений від рівня передачі даних та реалізується

програмно [1]. Ця концепція є однією з форм віртуалізації обчислювальних ресурсів.

Ключовими принципами програмно-керованої мережі є: розділення процесів передачі та керування даними, централізація керування мережею за допомогою уніфікованих програмних засобів, віртуалізації фізичних мережевих ресурсів [2]. На рис.1.1 зображено архітектуру програмно-конфігурованої мережі.

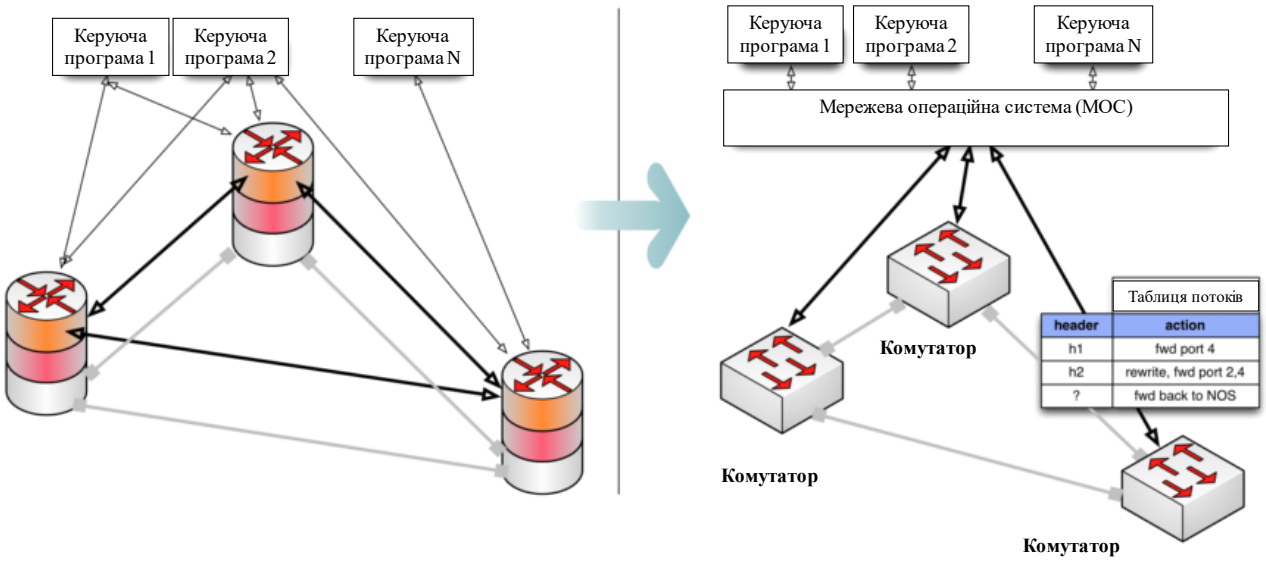


Рис. 1.1. Архітектура SDN

Основна ідея розвитку SDN-підходу полягає в тому, щоб:

- відділити керування мережевим обладнанням від управління передачею даних за рахунок розроблення спеціального програмного забезпечення, яке може працювати на окремому обладнанні під контролем адміністратора мережі;
- перейти від управління окремими екземплярами мережевого обладнання до керування мережею в цілому;
- створити інтелектуальний програмно-керований інтерфейс між мережним додатком та транспортним середовищем мережі.

У стандартній архітектурі ПКМ виділяють три основних рівні:

- інфраструктурний рівень (рівень передачі даних), на якому функціонують мережеві комутатори та канали передачі даних;

- рівень керування – набір програмних засобів, логічно відділених від рівня передачі даних, що забезпечують реалізацію механізмів керування пристроями інфраструктурного рівня;
- рівень мережевих додатків – неуніфікований та незалежний від виробника рівень для простішого керування мережею і внесення додаткових функцій та елементів, які потрібні саме власнику мережі.

Протокол OpenFlow [3] – протокол управління процесом обробки даних, що передаються по мережі маршрутизаторами та комутаторами. Використовується для керування мережевими елементами з центрального пристрою – контролера мережі, яким може слугувати звичайний ПК або сервер. Протокол OpenFlow, що реалізовує незалежний від виробника інтерфейс між логічним контролером мережі та мережевим транспортом, є однією з реалізацій концепції програмно-керованої мережі.

Ядром рівня керування програмно-керованої мережі є мережева операційна система – програмний засіб, що забезпечує, з одного боку, інтерфейс засобами рівня передачі даних (наприклад, динамічну зміну таблиць маршрутизації), а з іншого боку – прикладний програмний інтерфейс для рівня мережевих додатків, сформульований в термінах більш високого рівня абстракції (наприклад, «ім'я користувача», «назва оператора») порівняно з тим, який використовується в параметрах конфігурації мережевих пристроїв (IP-адреса, MAC-адреса, шлюз).

На відміну від традиційного тлумачення МОС (Мережевої операційної системи) [4] як операційної системи інтегрованої зі стеком мережевих протоколів, у дослідженні під МОС будемо розуміти програмну система, що забезпечує моніторинг, доступ, керування ресурсами всієї мережі, а не конкретного вузла. МОС формує дані про стан всіх ресурсів мережі й забезпечує доступ до них для додатків управління. МОС дає змогу створювати додатки як централізовані програми з високорівневими іменами на відміну від

розроблення розподілених алгоритмів, що використовують низькорівневі адреси.

Об’єктом керування мережевої операційної системи є один або кілька комутаторів. Контролер забезпечує набір інтерфейсів для створення, редагування, видалення, керування конфігурацією таблиць потоків у комутаторах. Комутатором керує програмний процес, який виконується на контролері.

Контролер повинен володіти інформацією про топологію мережі в будь-який момент часу. Інформація про топологію мережі також містить інформацію про розміщення користувачів та серверів, інших елементів та сервісів мережі, а крім того, прив’язку між іменами та адресами. Тому однією з найважливіших задач, що розв’язуються мережевою операційною системою, є постійний моніторинг мережі та побудова топології/карти мережі.

На сьогодні розроблені такі контролери з відкритим програмним кодом: NOX, POX, Beacon, Ryu, Thema [5; 6; 7; 8; 9]. Крім того, є компанії, що випускають свої власні, проте закриті рішення. Найвідомішою серед таких компаній є CISCO.

Використання стандартизованого відкритого інтерфейсу площини передачі даних дає можливість впроваджувати інновації набагато оперативніше, ніж це відбувається сьогодні. Власники мережі та оператори (також постачальники, дослідники і розробники) можуть додавати нові функціональні можливості й послуги в мережу. Нова функціональність та послуги впроваджуються за допомогою створення мережевих сервісів на мережевій операційній системі або контролері з використанням стандартизованого API. Це суттєво прискорює розвиток мережі, наприклад, впровадження нових методів контролю доступу.

Постійно актуальною проблемою в ПКМ є створення розподілених контролерів. Існуючі сегменти ПКМ підтримуються за допомогою єдиного контролера, що характеризується низкою проблем, а саме:

- проблема масштабованості контролера, тобто проблема збільшення тривалості встановлення нових потоків для великої мережі;

- проблема надійності, оскільки відмова контролера зумовлює відмову всієї мережі;
- Для вирішення зазначених проблем дослідники й розробники в галузі ПКМ дійшли висновку про необхідність впровадження фізично розподіленого рівня управління.

Комутатор є основною складовою мережі, а тому реалізація площини передавання даних у ПКМ має декілька підходів:

- використання спеціально розробленого комутатора OpenFlow. Таке рішення є найдорожчим, проте забезпечує найкращу реалізацію концепції ПКМ. Комутатори випускаються як з повною підтримкою функцій OpenFlow, так і гібридні, що здатні передавати потоки OpenFlow та здійснювати традиційну комутацію/маршрутизацію пакетів;
- використання вже встановлених комутаторів, які функціонують в існуючій мережі. Для цього необхідною є заміна програмної частини наявних комутаторів за умови, що ці комутатори здатні адаптуватися до потрібних змін. Крім того, зміна прошивки комутатора залежить від виробника обладнання, а тому використання в мережі обладнання різних виробників ускладнює перехід до ПКМ та подальше її функціонування;
- використання ПКМ разом з NFV (Network Functions Virtualization – віртуалізація мережевих ресурсів) [10]. Така концепція архітектури мережі пропонує здійснювати віртуалізацію цілих класів функцій мережевих вузлів у блоки, які можуть бути з’єднані в одну систему, щоб забезпечити послуги зв’язку. У локальних мережах віртуалізація ресурсів забезпечує можливість абстрагуватися від різнорідності обладнання за рахунок встановлення потрібної операційної системи на віртуальній машині [11]. У глобальних мережах віртуалізація здійснюється на основі Hypervisor [12], що дає змогу не просто створити єдину вертикальну площину, а й поділити мережеві ресурси між операторами, виділивши кожному необхідні ресурси для створення окремого домену.

Програмно-конфігуровані мережі разом з віртуалізацією ресурсів у центрах обробки даних забезпечують можливість поєднати переваги трьох моделей обслуговування, зокрема:

- Software-as-a-Service надає можливість клієнту використовувати прикладне програмне забезпечення з різних пристроїв чи за допомогою інтерфейсу програми;
- Platform-as-a-Service забезпечує користувачу можливість використання ресурсів центру обробки даних для встановлення базового програмного забезпечення з метою подальшого розміщення на ньому нових або існуючих додатків (власних, розроблених на замовлення);
- Infrastructure-as-a-Service надає ресурси для самостійного управління обробкою, зберіганням та процесом передавання даних, а також іншими фундаментальними обчислювальними процесами.

Поєднання описаних моделей за допомогою віртуалізації забезпечить операторам, а також власнику мережі можливість раціональніше використовувати ресурси центрів обробки даних. У свою чергу ПКМ дасть змогу користувачам отримувати доступ до потрібних їм Cloud ресурсів з необхідною якістю обслуговування.

1.2. Комутація пакетів та балансування навантаження на основі таблиць потоків

Комутатор OpenFlow зазвичай містить одну або більше таблиць потоків, а також одну групову таблицю. Використовуючи протокол OpenFlow, контролер може маніпулювати таблицями потоків, наприклад, додавати, оновлювати чи видаляти потоки, встановлювати обмеження на максимальну кількість потоків тощо.

Кожна таблиця потоків складається з набору потоків, а кожен потік містить правила, лічильники та набір інструкцій. Структурні елементи записів у таблиці потоків зображено на рис.1.2.

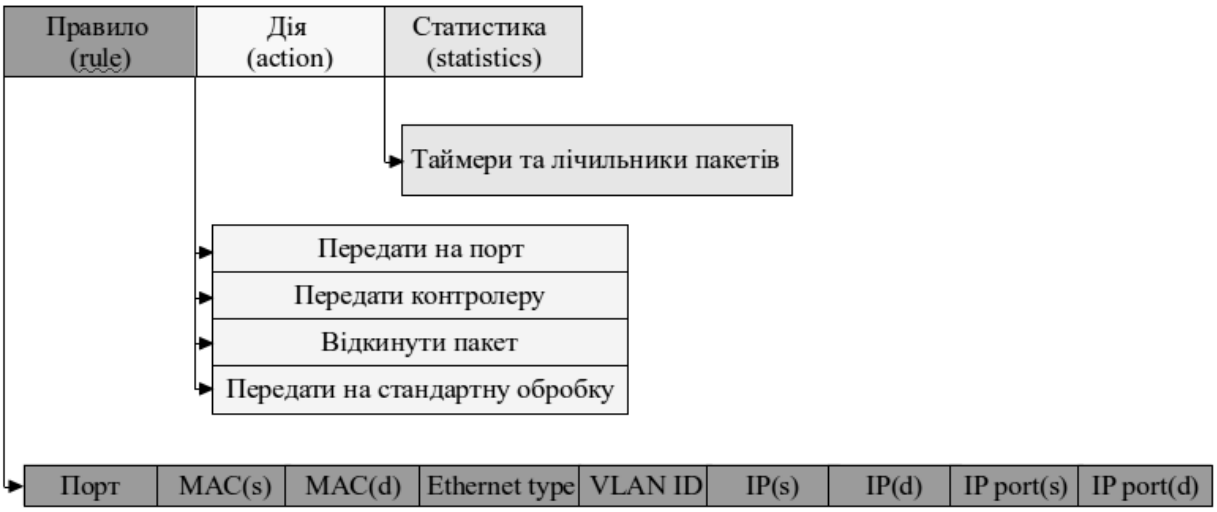


Рис. 1.2. Елементи таблиці потоків

У тому разі, коли на інтерфейс комутатора надходить новий пакет, комутатор відокремлює його заголовок, який утворений протоколами канального, мережевого та транспортного рівнів. Після цього комутатор порівнює ці заголовки з усіма потоками, що записані в таблиці потоків. Якщо відповідний потік знайдено, то комутатор виконує інструкції, присвоєні цьому потоку та обновлює лічильники статистики. До інструкцій належать: передача пакету на вихідний інтерфейс, модифікація полів заголовку, відкидання, широкомовна розсилка, відправлення на контролер.

Всі потоки в таблиці потоків характеризуються пріоритетом. Якщо для одного й того самого пакету знайдено два правила, то комутатор вибирає правило з вищим пріоритетом, і до пакету застосовуються відповідні інструкції. Якщо в таблиці потоків не знайдено потоку, якому належить цей пакет, то подальші дії залежать від конфігурації комутатора, зокрема: пакет може бути відправлений на контролер, відкинутий, переданий на пошук в іншій таблиці або ж відправлений в ту частину комутатора, яка відповідає за традиційну обробку пакетів.

Інструкції, пов’язані з кожним потоком, можуть вказувати комутатору на необхідність пошуку пакету в груповій таблиці або ж ще одній таблиці потоків. Передача пакетів між таблиця потоків від вхідного інтерфейсу до вихідного дає

змогу гнучкіше обробляти трафік, а сам принцип організації таблиць називають каскадуванням (рис.1.3).

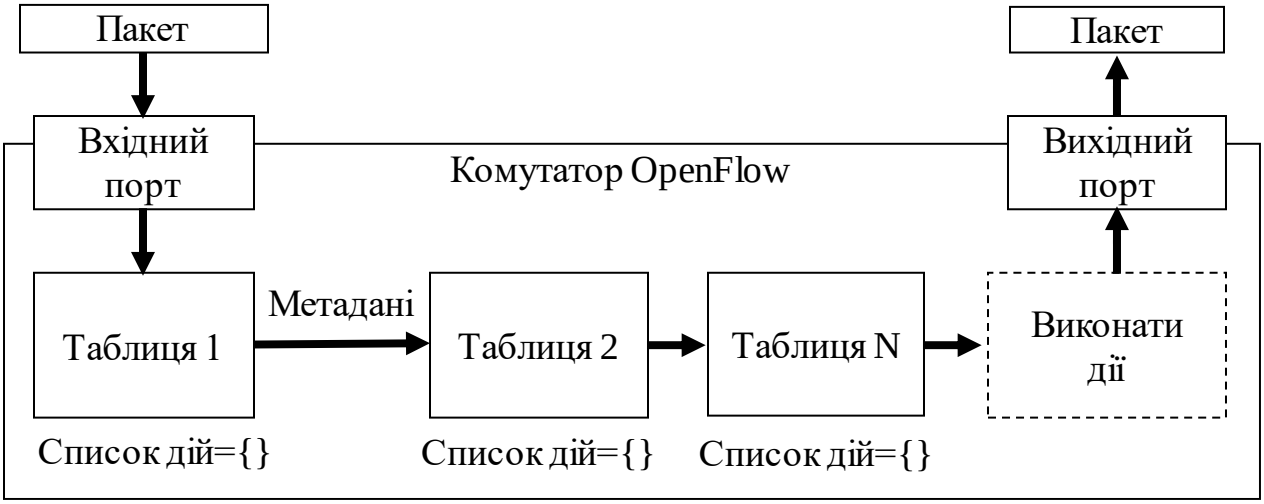


Рис. 1.3. Модель гнучкої обробки пакетів на основі каскадування таблиць потоків

Інструкції каскадної обробки забезпечують комутатору можливість пересилати пакети в наступні таблиці потоків для подальшої обробки. Разом з пакетом між таблицями може передаватися службова інформації у форматі метаданих. Каскадна обробка зупиняється тоді, коли набір інструкцій, пов'язаних з відповідним потоком, не вказує на наступну таблицю. Це означає, що з пакетом виконуються всі необхідні інструкції, і він передається на вихідний інтерфейс. Як правило, ним є фізичний порт, але він також може бути віртуальним портом, визначеним комутатором або зарезервованим віртуальним портом, визначеним специфікацією. Зарезервовані віртуальні порти можуть використовуватися для відправлення на контролер, на всі порти або ж для відправлення з використанням традиційних методів обробки. Порти, визначені комутатором як віртуальні, можуть створювати групи агрегації каналів. Потоки можуть також вказувати на групову таблицю, що означає додаткову обробку.

Відповідно до принципу каскадування комутатор починає виконувати пошук пакету в першій таблиці потоків. Поля, що використовуються для співставлення, залежать від типу пакета. Пакет відповідає запису таблиці потоків, якщо значення полів у пакеті співпадає з відповідними полями в

правилі потоку. Якщо поле в правилі містить значення ANY(«*»), тоді відповідне поле в заголовку пакета може містити довільне значення.

Така модель обробки пакета відкриває унікальні можливості. Комутатор може використовуватися в ролі як комутатора, так і маршрутизатора чи мережевого екрану. Вся функціональність закладена в таблиці потоків, приклад якої з правилами для реалізації різних ролей функціонування комутатора представлено в табл.Таблиця 1.1.

Таблиця 1.1

Приклади реалізації різноманітних спеціалізованих функцій комутатора за допомогою таблиці потоків

	Порт	Eth src	Eth dst	Eth type	VLAN ID	IP src	IP dst	TCP src	TCP dst	Дія
Ethrnet комутатор	*	00: 1F	*	*	*	*	*	*	*	Порт №6
Мережевий екран (TCP)	*	*	*	*	*	*	*	*	22	Відкину -ти
IP маршрутизатор	*	*	*	*	*	*	192. 168. 0.1	*	*	Порт №1
OpenFlow комутатор	#3	00: 20	00: 1F	0800	1	192. 168. 0.2	192. 168. 0.1	27	80	Порт №4

Групова таблиця містить записи груп, а кожен запис містить також список інструкцій зі специфічною семантикою, що залежить від типу групи. Дії в одній застосовуються для всіх пакетів, що посилаються до цієї групи. Модель комутації з використанням групової таблиці представлено на рис.1.4.

Збір статистики реалізується за допомогою лічильників. Лічильники можуть бути призначені для кожної таблиці, потоку, порту, черги або групи. Табл.1.2 містить набори лічильників, визначені спеціалізацією OpenFlow. OpenFlow-сумісні лічильники можуть бути реалізовані в програмному забезпеченні. Вони відображають інформацію на основі опитування апаратних лічильників, які мають більш обмежений діапазон значень.

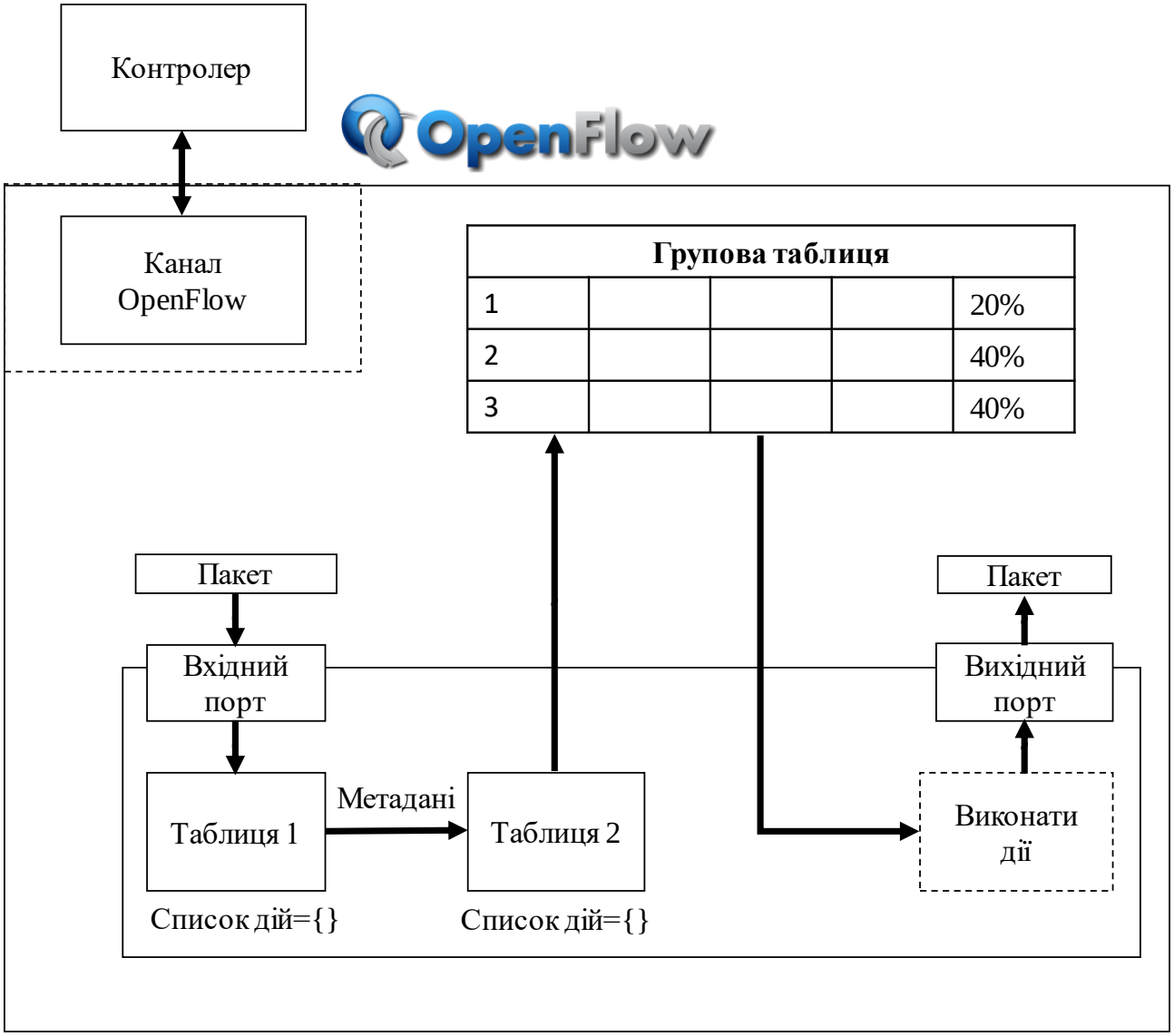


Рис. 1.4. Модель комутації з використанням групової таблиці потоків та реалізація балансування навантаження на основі вагових коефіцієнтів шляхів

Кожен потоковий запис містить набір інструкцій, які виконуються в тому разі, коли пакет відповідає правилу. Всі інструкції можна розділити на такі типи: для негайного виконання, для очищення списку інструкцій, для додавання нової інструкції, для запису метаданих, для переходу до наступної таблиці. Інструкції для негайного виконання дають вказівку застосувати певні дії негайно, без будь-яких змін у наборі дій. Така інструкція може бути використана для модифікації пакета при передачі його до іншої таблиці. Інструкції для очищення видаляють всі інструкції з набору інструкцій для окремого потоку. Інструкції для додавання нової інструкції доповнюють наявний набір інструкцій для окремого правила новими інструкціями.

Інструкція переходу до наступної таблиці містить номер таблиці, в яку пакет буде переданий після обробки в поточній таблиці. Номер наступної таблиці обов'язково повинен бути більшим від номера поточної таблиці.

Таблиця 1.2

Список лічильників

Підрахунок	Лічильник	Кількість біт
По таблицях	Підрахунок заголовків	32
	Пакетів-запитів	64
	Пакетів-відповідностей	64
По потоках	Отриманих пакетів	64
	Отриманих байт	64
	Тривалість (в с)	32
	Тривалість (в нс)	32
По портах	Отриманих пакетів	64
	Переданих пакетів	64
	Отриманих байт	64
	Переданих байт	64
	Отриманих частин	64
	Переданих частин	64
	Отриманих помилок	64
	Переданих помилок	64
	Отримання циклічної помилки	64
	Отримання помилки переповнення	64
	Отримання CRC помилки	64
	Колізії	64
По чергах	Переданих пакетів	64
	Переданих байт	64
	Переданих помилок перевантаження	64

Для кожного потоку визначено список інструкцій. Цей список є порожнім за замовчуванням. Надалі список інструкцій для окремого потоку може змінюватися. Набір операцій зберігається між потоком таблиць. Список інструкцій містить максимум одну інструкцію кожного типу. Інструкції в списку застосовуються в порядку, вказаному нижче, незалежно від порядку, в якому вони були додані до списку. Комутатор може підтримувати довільний порядок виконання інструкцій, який визначається реалізацією функцій застосування інструкцій. Стандартний порядок є таким: копіювання зовнішнього TTL, видалення мітки, додавання мітки, копіювання внутрішнього TTL, зниження TTL, застосування всіх дій, застосування всіх маніпуляцій з

QoS, застосування всіх групових дій, пересилання пакета у вказаному напрямку.

1.3. Методи та засоби моніторингу параметрів мережних елементів та мультисервісних потоків

Кожен день мільйони комп'ютерів під'єднуються до Інтернету, генеруючи великі обсяги трафіку. Процес є динамічним, оскільки обсяг трафіку постійно зростає за рахунок нових додатків та сервісів, які доповнюють глобальні потоки трафіку. У зв'язку з тим, що зростання трафіку відбувається постійно, для оператора мережі особливо важливим є отримувати актуальну інформацію про стан конкретної частини мережі. Вимірювання параметрів трафіку дає змогу аналізувати характеристики функціонування мережі в реальному режимі часу, що, в свою чергу, дозволяє динамічно гарантувати необхідний сервісний рівень у середовищі, яке постійно змінюється. Можливість отримувати інформацію про потоки даних у реальному часі дає змогу аналізувати проблеми, що виникають в мережі, генерувати матриці навантаження та оптимізувати мережу, використовуючи методи управління трафіком. Більше того, це дає змогу впровадити алгоритми маршрутизації та реалізувати складніші алгоритми прийняття рішень, збільшуючи ефективність використання ресурсів та зменшуючи перевантаження мережевих каналів чи вузлів.

У ПКМ переважна більшість функцій моніторингу є вбудованою та стандартизованою у протоколі OpenFlow, який здійснює моніторинг основних елементів мережі. Серед програм та додатків, що використовуються для моніторингу на основі OpenFlow є такі:

Ping [13] Додаток, що використовується для вимірювання затримки в мережі. Його перевага полягає в тому, що він використовує пакети обсягом 24 байти, а тому не сильно завантажує трафік;

Ndb [14] – мережевий прототип відлагоджувача (debugger). Завдяки цьому засобу комутатор підтримує відстеження обробки пакетів, зберігає записи таблиці, що відповідають певним заголовкам пакета. Ndb дає змогу виявити

помилки, що впливають на правильність вибору шляху, у тому числі логіку управління помилок, умови мережі, помилки конфігурації, несподівані формати пакетів і недоліки реалізації комутатора;

Oftrace [15] – бібліотека OpenFlow для аналізу та відстеження перевантажених елементів. Вхідними даними є вже відформатовані файли та вихідні корисні статистичні дані про сесію OpenFlow. Прикладами таких додатків є ofstats/pyofstats (виводить час затримки обробки контролера) та ofdump/pyofdump (створює списки керуючих повідомлень OpenFlow з часовими мітками). Oftrace працює на будь-якому контролері, допомагає відстежувати проблемні ділянки мережі.

Окрім охарактеризованих вище є ще засоби, що не просто вбудовані в концепцію SDN, а використовують за основу протокол OpenFlow. Серед них можна виділити:

Oflops [16] – автономний контролер для тестування різних аспектів комутаторів OpenFlow. Реалізує модульну структуру для додавання та виконання незалежних тестів для кількісної оцінки ефективності комутаторів;

Cbench [17] – програма для тестування контролерів. Емулює певну кількість комутаторів, які підключаються до контролера, відправляє пакети-повідомлення та стежить за модифікацією потоків.

Додатки для моніторингу ПКМ – є одним з видів програмного забезпечення, яке контролює роботу мережі та процес передавання даних в межах мережі з заданими параметрами [18]. Такі додатки є, перш за все, інструментом управління мережею та моніторингу, який співпрацює з мережевою операційною системою, використовуючи спеціалізований програмний інтерфейс API [19]. API інтегрований у всі мережні пристрої і допомагає отримати конкретні дані, які запитує додаток моніторингу. Після цього додаток зберігає отримані дані та використовує їх в подальшому для аналізу й прогнозування стану мережі.

Традиційно використовується велика кількість методів вимірювання. Варто також зазначити, що OpenFlow надає засоби для реалізації значної

кількості методів моніторингу, включаючи їхні комбінації, в той час, як без OpenFlow кожен тип вимірювання потребує встановлення окремих апаратних та програмних засобів.

Всі методи вимірювання мережевих параметрів поділяють на два типи: активні та пасивні. До пасивних методів можна віднести вимірювання мережевого трафіку без введення додаткових тестових пакетів у мережу. Переваги цих методів полягають у тому, що вони не створюють надлишкового навантаження. Проте недоліком їх є необхідність встановлення додаткового апаратного забезпечення (агентів моніторингу). З іншого боку, активні методи використовують додаткові тестові пакети для вимірювання того чи іншого параметру. Найбільш відомими засобами, що використовують активний метод вимірювання, є *Traceroute* та *Ping*. Обидва додатки методів є корисними для моніторингу мережі, проте необхідно брати до уваги деякі аспекти при створенні нових засобів моніторингу, зокрема: інформаційна надлишковість та точність.

Для вимірювання стану каналів зазвичай використовуються лічильники портів. Сьогодні оператори мереж переважно використовують протокол SNMP для вимірювання завантаження каналу [21]. Лічильники SNMP агентів на кожному інтерфейсі кожного комутатора використовуються для вимірювання кількості переданих байт та пакетів. Проте існують суперечки стосовно того, наскільки часто можна опитувати комутатор та як це впливає на завантаження його центрального процесора. Інформація, зібрана за допомогою SNMP лічильників не дає змогу отримати статистику стосовно кожного окремого потоку та поведінки кінцевих користувачів. Більш того, SNMP нездатний виміряти інші параметри, тому його не можна використовувати як єдине рішення для моніторингу мережі.

Для масштабованого моніторингу в реальному режимі часу існують спеціалізовані рішення, зокрема такі: NetFlow [22] та sFlow [23]. В їх основі лежить метод вибірки пакетів. З кожного потоку періодично вибирається один пакет, проте це не дає достатньо високої точності. Наприклад, вибірка

здійснюється для кожного двохсотого пакета, а кожну п’яту хвилину маршрутизатор відправляє статистику потоку на центральний компонент моніторингу для подальшого аналізу.

Сьогодні затримка та втрати пакетів переважно зумовлені вимірюванням на прикладному рівні. На жаль, оператор мережі не має доступу до кінцевих пристроїв користувачів та додатків, які вони використовують, а тому не може використовувати методи, які потребують залучення кінцевих терміналів.

Приклад моніторингу з використанням пасивного методу представлено в роботі [24]. Метод моніторингу полягає в захопленні заголовку кожного пакету і внесенні в нього поточного часу, після чого такий пакет вводиться в канал знову. В іншій точці мережі пакет знову вибирається з мережі та на основі записаної мітки визначає час передачі. Такий метод є достатньо точним і дає можливість виміряти затримку до декількох мікросекунд. Проте він потребує центрального елемента, який має обробляти зібрані вимірювання, а всі точки вимірювання вимагають точної синхронізації годинників. Інший подібний метод використовується для вимірювання втрат [25]. Цей метод позначає кожен пакет унікальною міткою під час його проходження через вузол відправник і встановлює, чи він був отриманий на вузлі приймачі.

OpenTM [26] формує матрицю трафіку на основі моніторингу статистики для кожного потоку, яка є результатом опитування всіх комутаторів у мережі. Додаток визначає, який комутатор опитувати вже в процесі функціонування мережі. У статті [27] проведено порівняння декількох алгоритмів опитування для статичного інтервалу моніторингу тривалістю 5 секунд. Інший метод активного вимірювання представлено в роботі [28]. Автори використовують факт того, що кожен новий потік повинен пройти через контролер. Це дає змогу направити трафік до однієї з доступних систем моніторингу, яка проаналізує його та обновить статистичні дані.

Повідомлення на зразок *PacketIn* та *FlowRemoved,* відправлені комутатором на контролер, несуть інформацію про те, коли потік встановлений та коли видалений. Додатково повідомлення *FlowRemoved* містить інформацію

про кількість переданих байт/пакетів протягом всього існування потоку. Ця особливість використовується в пасивних вимірюваннях та реалізована в системі моніторингу *FlowSence* [29]. Тим більше, існують пропозиції щодо розроблення нового протоколу для програмно-керованих мереж, який дасть змогу збирати статистику. Деякі напрацювання в цьому напрямку представлено в роботі [30], де запропоновано нову архітектуру системи моніторингу трафіку. Щоб продемонструвати можливості нової архітектури, автори проводять п’ять типів вимірювання. Метою вимірювання є визначення найбільш активних потоків, джерела широкомовного передавання, зміни обсягів трафіку, розподілу розмірів потоків.

Додатково в процесі дослідження проаналізовано функціональність таких рішень для моніторингу в ПКМ:

BISmark [31] – додаток для активного та пасивного вимірювання параметрів процесу передавання даних;

DCM [32] – розподілена система моніторингу трафіку;

FleXam [33] – забезпечує гнучкий спосіб аналізу пробних пакетів за допомогою OpenFlow;

OpenNetMon [34] – система для моніторингу параметрів якості обслуговування потоків з метою здійснення управління трафіком, яка дає змогу виміряти також часові параметри процесу передавання пакетів;

OpenSample [35] – вимірювання на основі високошвидкісної вибірки пробних пакетів із загального потоку;

PaFloMon [36] – реалізує метод пасивного моніторингу для використання кінцевими користувачами.

Результати аналізу підтверджують, що кожна система націлена на моніторинг обмеженого числа параметрів мережі а тому не дає змогу отримати необхідну інформацію для ефективної маршрутизації мультисервісного трафіку.

1.4. Моделі балансування навантаження та підвищення якості обслуговування пріоритетних потоків у програмно-конфігурованих мережах

Всі роботи, пов'язані з якістю обслуговування потоків у програмно-конфігурованих мережах, можна розділити на три категорії: у першій категорії робіт досліджують принципи динамічного керування якістю обслуговування трафіку [37; 38; 39]; у другій категорії – вплив апаратних та програмних характеристик комутаторів на параметри продуктивності мережі та забезпечення необхідної якості обслуговування; у третій категорії – параметри функціонування мережі, які можуть мати потенційний вплив на якість обслуговування користувачів, наприклад, доступність, адаптивність, відмовостійкість тощо.

У роботі [37] розроблено платформу для забезпечення високої якості надання сервісів потокового відео користувачам. Платформа гарантує користувачам параметри якості обслуговування на основі виділення стабільної смуги пропускання. Платформа також використовує TCP потоки з тривалим часом існування. Проте робота зорієнтована тільки на сервіси потокового відео і не враховує всі інші класи трафіку. У роботі [40] запропоновано прототип контролера, який дає змогу забезпечити гарантовану якість обслуговування з кінця в кінець за рахунок маршрутизації потоків з урахуванням необхідних параметрів функціонування мережі. У тому разі, коли надходять нові потоки, контролер визначає доступну пропускну здатність всіх каналів та вузлів, на основі чого встановлює потоки через найкоротші шляхи для забезпечення якості обслуговування. Хоч функціонування розробленого прототипу тестується з допомогою одного програмного та одного апаратного комутатора, основну увагу в роботі сконцентровано на функціонуванні самого контролера, і в ній не досліджено вплив гарантій, які надаються пріоритетним потокам, на всі інші потоки, що паралельно існують в мережі. Для мереж, що використовуються центрами обробки даних, також є свої рішення. Наприклад, у роботі [39] розроблено платформу для уникнення перевантаження каналів у

випадку міграції віртуальних машин. Основним результатом функціонування цієї платформи є зниження впливу міграції віртуальних машин у центрі обробки даних на всі ніші потоки в мережі за рахунок механізмів обмеження пропускної здатності для процесів міграції. Їхні результати без сумніву є корисними в процесі обслуговування мультисервісного навантаження, зокрема для дослідників, які займаються проблемами віртуалізації мережевих ресурсів та строгого розподілу пропускної здатності під кожну віртуальну мережу.

Робота [38] присвячена проблемі управління пропускною здатністю для сервісу потокового відео *Youtube*. Зокрема, досліджується можливість зміни пропускної здатності залежно від обсягу буферизованих даних на кінцевому пристрої. У цій роботі автори також вивчають та порівнюють вплив різних стратегій формування черг. Варто зауважити, що дослідники більше уваги звертають на управління ресурсами.

У роботі [41] досліджено функції забезпечення сталої швидкості потоків на апаратних комутаторах *Pica8 P3290*. Досліджено вплив UDP потоків з різною тривалістю існування на перевантаження каналів та пристроїв, через те, що інтенсивна модифікації таблиць потоків (видалення старих та встановлення нових) можуть раптово підвищити інтенсивність вхідного навантаження до критичного рівня для певного мережного елементу. Це є особливо важливим, оскільки в умовах мультисервісного трафіку стрибки інтенсивності навантаження навіть при відомій швидкості потоків є звичайним явищем, яке може призводити до локальних втрат та перевантаження інтерфейсів обслуговуючих пристроїв. Автори зазначеної роботи звертають увагу на статичне управління потоками, а також розглядають потоки TCP за умов динамічного управління якістю обслуговування.

Автори роботи [42] досліджують властивості лічильників OpenFlow. На основі лічильників комутатор може обмежувати пропускну здатність потоку до необхідного значення. У зазначеній роботі такий підхід використовується для обмеження TCP потоків з метою уникнення перевантаження каналів. Автори встановлюють, що відкидання пакетів за допомогою лічильників дає змогу

уникнути перевантаження каналу за рахунок того, що протокол TCP різко сповільнює швидкість передачі та переходить у режим повільного старту. У процесі дослідження всі експерименти проводяться з використанням емулятора *Mininet* без використання реального обладнання. При цьому методи управління потоками та параметрами якості обслуговування є статичними, тобто не адаптуються до змін інтенсивності трафіку в мережі.

У роботі [43] впроваджено механізми динамічної маршрутизації для мінімізації негативного впливу потоків з гарантованими параметрами якості обслуговування на потоки з негарантованою доставкою. У мережі може виникнути ситуація, коли немає більше доступних шляхів для маршрутизації потоків. Для цього на основі *OpenQoS* у роботі [44] автори оптимізували алгоритм маршрутизації і запропоновано адаптивний підхід до маршрутизації потокового відео, який змінює маршрути для окремих потоків реального часу з метою досягнення необхідної якості обслуговування. Проте запропоновані рішення не можуть гарантувати якість обслуговування в умовах обмежених ресурсів пропускної здатності. Авторами роботи [45] представлено підхід для виділення пропускної здатності, що задовольняє вимоги до параметрів QoS для всіх клієнтів, які користуються *Cloud* сервісами. Однак всі експерименти проведено на основі контролера, який реалізований як програний компонент, інтегрований у програмний комутатор *OpenVSwitch*.

Рішення щодо трансляції потокового відео через OpenFlow мережу представлено у роботі [46]. Автори пропонують розв'язувати дві оптимізаційні проблеми при розробці логіки керування мережею. Їхнє рішення дає змогу забезпечити передачу даних без втрат. Результати показують, що середня якість обслуговування потокового відео покращена на 14% за рахунок зміни маршруту потоків основного рівня та на 6,5% завдяки зміні маршруту додаткового рівня. *OpenQoS* [47] пропонує рішення щодо забезпечення якості обслуговування з кінця в кінець для доставки мультимедійного контенту. Маршрути потоків реального часу перенаправляються в режимі реального часу, щоб забезпечити параметри якості обслуговування, зокрема такі: затримка та

втрати. Автори статті пропонують здійснювати динамічну маршрутизацію потоків, чутливих до погіршення параметрів QoS, в той час як менш чутливі потоки залишаються на тих самих коротких маршрутах. Результати показують, що запропоноване рішення дає змогу забезпечити передавання потокового відео без втрат. Більш того, вони підтверджують, що така маршрутизація не має впливу на інші потоки, які проходять у мережі. Недоліком рішення є те, що *OpenQoS* не підтримує диференціації потоків окремих користувачів.

У статті [48] розглянуто архітектуру системи забезпечення параметрів якості обслуговування на основі протоколу OpenFlow. Система використовується для розширення можливостей забезпечення якості обслуговування потокам реального часу з двома рівнями QoS: базовим та розширеним. Маршрутизація здійснюється на основі розв'язання задачі знаходження найкоротшого шляху з обмеженнями CSP (*Constrained Shortest Path Problem*). Результати показують, що система дає змогу суттєво підвищити якість обслуговування для потокових сервісів. Маршрутизація потоків відео основного рівня забезпечує необхідні параметри якості обслуговування потоків реального часу. Маршрутизація розширеного рівня використовується у випадках перевантаження мережі, а також у випадку наявності великої кількості потоків основного рівня з низькою швидкістю.

Ще одну проблему оптимізації маршрутизації за критеріями параметрів якості обслуговування розглянуто в роботі [49]. Автори статті пропонують розширювати архітектуру рівня керування ПКМ для підтримки маршрутизації на основі параметрів якості обслуговування. Формулювання проблеми забезпечує шляхи для потоків з різними вимогами до QoS, які конвертуються в таблицю потоків різного класу трафіку. Крім того, автори розробляють тестове середовище на основі контролера, де встановлюються сервісні договори, конфігуруються маршрути для потоків та здійснюється моніторинг мережевих параметрів, щоб у випадку перевантаження мережі провести динамічну маршрутизацію.

Питання *QoE* (Quality of Experience) розглядається в багатьох роботах. У роботі [50] запропоновано платформу для підвищення якості сприйняття клієнтів у мережах OpenFlow. Використовуючи особливості ПКМ та OpenFlow, автори пропонують підвищити *QoE* всім клієнтам, що користуються сервісом потокового відео, з врахування їхніх вимог до параметрів якості обслуговування. Основною характеристикою системи є динамічна адаптація відео потоків для гарантування рівномірності якості сприйняття для всіх користувачів.

У статті [51] запропоновано метод оптимізації якості відеоконтенту, що передається протоколом HTTP в ПКМ, який полягає в пошуку найближчого на конкретний момент сховища контенту до користувача. У роботі [52] запропоновано систему *QoSFlow*, яка дає змогу зробити контроль за якістю обслуговування більш гнучким. Метою системи є здійснювати контроль над алгоритмами управління чергами в комутаторах. Автори перевіряють ефективність рішень на основі оцінки тривалості відгуку алгоритмів планування в площині передачі даних та тривалість відгуку площини передавання даних на зміни її параметрів системою управління. Основними параметрами, що оцінюються, є максимальна пропускна здатність, інтенсивність використання апаратних ресурсів та QoE.

У статті [53] пропонується глобальна мережа з програмним керуванням, яка забезпечує ефективну передачу великих обсягів трафіку між центрами обробки даних. Єдина система керування мережею здійснює глобальну координацію швидкостей передачі даних всіх потоків і для кожного з них виділяє необхідну пропускну здатність. Система керування динамічно вибирає пропускну здатність, яку слід надати тому чи іншому сервісу. Основною проблемою можуть стати переповнення, що виникають внаслідок підвищеної частоти оновлення площини передачі даних. Однак автори пропонують залишати резервну пропускну здатність у кожному каналі з метою уникнення переповнення каналів. Результати експериментів показують, що мережа здатна передати до 60% більше трафіку, ніж мережі з існуючим керуванням, без

надлишкового сигналізаційного керування та оперативним внесенням змін у площину передачі даних.

1.5. Недоліки існуючих методів моніторингу та забезпечення якості обслуговування мультисервісних потоків

У результаті аналізу наукової літератури та впроваджених рішень можна зробити висновок, що більшість з них орієнтована лише на забезпечення вимог щодо якості обслуговування певній групі потоків. У той час, як деякі роботи концентруються на потоках реального часу, в інших роботах досліджують методи підтримки якості обслуговування для потоків нереального часу в мережі. Загалом ці роботи доповнюють одна одну та вирішують спільну проблему. Проте рішення, запропоновані в них, залишаються окремими рішеннями і не можуть гарантувати якісного обслуговування всіх потоків у програмно-конфігурованій мережі. Крім того, практично жодна з праць не розглядає потоки з точки зору клієнта. Класифікація потоків для забезпечення якості обслуговування залишається, в кращому випадку, на рівні класів трафіку.

У той час, як розробка платформ для забезпечення якості обслуговування базується на алгоритмах маршрутизації та багатошляхової передачі потоків, в них не враховується вплив багатошляхового поширення на якість обслуговування. У більшості досліджень підвищення якості обслуговування досягається зміною маршруту. При чому, маршрут змінюється саме для того потоку, якість обслуговування якого є незадовільною. Це може мати негативний вплив на перенаправлений потік у зв’язку з тим, що відбувається частковий розрив з’єднання в момент, коли потік змінює напрям. Це пов’язано з різницею затримки пакетів. Крім того, новий шлях не завжди означає кращу якість обслуговування. Останнє залежить від методу маршрутизації та критерію вибору шляху. Більшість моделей маршрутизації потоків вирішують проблему знаходження найкоротшого шляху. У переважній більшості випадків це можна здійснити за допомогою алгоритму Дейкстри та його модифікацій. Існують також модифіковані алгоритми Дейкстри, для яких метрика каналу є

комплексним значенням, що враховує не тільки пропускну здатність, але й інші параметри каналу, наприклад, затримку, втрати чи більш складні параметри, зокрема середню ймовірність блокування.

Для вирішення проблеми знаходження найкоротшого шляху використовують також аналітичні методи оптимізації, що базуються на різних моделях, наприклад, транспортних чи потокових. Клас потокових моделей дає змогу адекватно описати процеси передачі даних у програмно-конфігурованих мережах. Потокова модель забезпечує можливість оптимально розподілити навантаження з урахуванням вартості каналів. Проте використання лише методу оптимізації не дає змогу забезпечити необхідні параметри якості обслуговування для всіх потоків, оскільки неможливо диференціювати потоки за класом сервісу чи іншою ознакою для врахування індивідуальних потреб потоку щодо параметрів якості обслуговування. Отже, маршрутизація проводиться однаковим чином для всіх потоків. Це означає, що при недостатній пропускній здатності в каналі з найкращими умовами передачі потік, чутливий до розриву з'єднання або зміни порядку пакетів, може бути розділений на два шляхи, а це ставить під сумнів забезпечення необхідного значення джиттеру та уникнення втрат пакетів. У той же ж час потоки, для яких можна забезпечити низьку якість обслуговування, маршрутизуються каналами з найкращою якістю обслуговування. Зрозуміло, що у такій ситуації можна здійснити перемаршрутизацію всіх потоків для перегрупування потоків з низьким пріоритетом у канали з нижчою якістю обслуговування. Проте виконання таких операцій пов'язане з серйозним ризиком спричинити перевантаження мережі службовою інформацією, спровокувати розрив з'єднання для багатьох користувачів та суттєво погіршити якість обслуговування.

Найбільшим недоліком існуючих методів підвищення якості обслуговування є відсутність інформації про стан мережі в режимі реального часу. Більшість рішень про перемаршрутизацію приймаються на основі інформації, яка є відносно застарілою в умовах динамічного мультисервісного середовища. Згідно проаналізованих систем та методів моніторингу, збір

статистики в існуючих системах моніторингу відбувається за допомогою агентів, що встановлені на мережних вузлах та агрегують статистичну інформацію, яку передають з певним інтервалом у центр моніторингу. Зазвичай інтервал між двома опитуваннями є досить великим (в середньому досягає однієї хвилини), оскільки в сучасних маршрутизаторах частота опитування суттєво впливає на завантаження центрального процесора, а, отже, на пропускну здатність обслуговуючого пристрою. Зрозуміло, що збільшення частоти опитування призводить до зростання обсягу службової інформації, яка буде генеруватися системою моніторингу в процесі обміну повідомленнями між центром моніторингу та агентами. Існуючі системи моніторингу дають змогу збирати обмежену інформацію про трафік, яка зводиться максимум до завантаження портів чи окремих черг у буфері. У той же ж час протокол OpenFlow надає можливість проводити моніторинг більшої кількості параметрів. На основі централізованої архітектури ПКМ розроблено методи вимірювання параметрів якості обслуговування, зокрема таких: затримка та джиттер. Проте вони вимірюють затримку в каналі для агрегованого потоку, що не може адекватно відобразити якість обслуговування для потоку окремого класу, оскільки для їхнього обслуговування зазвичай на кожному маршрутизаторі використовуються алгоритми пріоритетного обслуговування. У такому випадку пакет обслуговується однією з черг, яка відповідає за певний клас трафіку, тоді як особливо важливі потоки можуть зазнавати погіршення якості обслуговування. Однією з основних проблем існуючих систем моніторингу є їхня статичність. При високому завантаженні каналу мережа повинна оперативно реагувати на критичні ситуації для уникнення перевантаження. Проте інтервал моніторингу існуючих мереж є незмінним і зазичай перевищує 5 секунд. Це означає, що контролер отримає усереднене значення інтенсивності завантаження каналу за 5 секунд і не зможе встановити факт виникнення моментального перевантаження внаслідок стрибка інтенсивності навантаження. Таким чином, піки інтенсивності навантаження згладжуються, і система управління спостерігає рівномірне завантаження

каналу без перевантаження. З іншого боку, моніторинг незавантажених елементів мережі також має сталий інтервал, а це, у свою чергу, означає, що генерується надлишкове службове навантаження.

Ураховуючи проведений аналіз, можна стверджувати, що існуючі моделі управління трафіком у програмно-конфігурованих мережах:

- не враховують вимоги окремого клієнта, а диференціюють потоки лише за класами трафіку;
- не використовують актуальні параметри якості обслуговування як окремих каналів, так й індивідуальних потоків окремого клієнта;
- не можуть справитися з перевантаження елементів мережі, вузлів чи каналів;
- проводять маршрутизацію потоків, не диференціюючи їх за чутливістю до перемішування порядку пакетів та розриву з'єднання.

Недолік існуючих систем моніторингу полягає в тому, що вони не дають змогу проводити аналіз якості обслуговування окремо вибраних шляхів у мережі. Лише даних про завантаження певного мережевого інтерфейсу не достатньо для того, щоб робити висновки стосовно того, чи він перевантажений, чи ні, оскільки у випадку, коли швидкість потоку рівна швидкості передачі інтерфейсу, якість обслуговування буде стало задовільною. У разі збільшення швидкості потоку інтерфейс буде завантажений не більше, ніж на 100%, проте втрати та затримка суттєво зростуть за рахунок зберігання пакетів у буфері. Зазвичай завантаження інтерфейсів у мережах обчислюють як відношення кількості переданої інформації до максимально можливого обсягу передачі інформації для конкретного інтерфейсу. Як наслідок, система керування мережею не здатна провести оптимізацію передачі потоків без зовнішнього втручання, наприклад, адміністратора.

Ще одним недоліком існуючих систем моніторингу є відсутність можливості відрізнити потоки окремих користувачів в агрегованому потоці пакетів одного класу. Якщо розглянути класичну таблицю маршрутизації IP

пакетів, то стає зрозумілим, що маршрутизатор ідентифікує пакет лише за IP адресою мережі призначення, яку вираховує за допомогою маски. Другим ідентифікатором є мітка якості обслуговування, на основі якої маршрутизатор відносить пакет до того чи іншого типу трафіку та записує його у відповідну чергу вхідного буферу. У результаті маніпуляція трафіком суттєво обмежена, оскільки, у разі виникнення вузького місця в мережі, перенаправлення здійснюється для цілого потоку одразу (IP адреса призначення, мітка якості обслуговування). Це може негативно вплинути на якість передачі потоків з низькою швидкістю, а саме вони й формують зазначений агрегований потік. Особливо гостро постає проблема виникнення джиттеру для потоків реального часу, пакети яких передаються по декількох шляхах внаслідок балансування навантаження. Деякі методи спрямовані на знаходження шляхів, затримка яких забезпечить мінімальний джиттер.

Висновки до 1-го розділу

Проведено порівняльний аналіз принципів побудови, архітектури та процесів функціонування традиційних (з децентралізованим керуванням) та програмно-керованих мереж. В архітектурі програмно-керованих мереж функції управління та передачі даних розділені на два окремих рівні, які реалізуються на окремих фізичних пристроях. Рівень керування представлений контролером, що використовує протокол OpenFlow для управління комутаторами, які представляють рівень передачі даних. Завдяки північному та південному інтерфейсу, а також можливості програмувати функції площини передачі даних контролер утворює мережеву операційну систему, на яку є можливість встановлювати різноманітні додатки для інтелектуального управління мережею та забезпечення якісних параметрів мережі, зокрема таких: якість обслуговування потоків, адаптивність, надійність, стійкість, доступність та ефективність. Визначено, що рівень забезпечення зазначених характеристик залежить від реалізації програмних додатків, встановлених на контролер, а також архітектури самого контролера.

Аналіз розроблених методів та моделей управління програмно-керованими мережами показав, що проблема забезпечення якості обслуговування займає важливе місце в наукових працях закордонних та вітчизняних дослідників. Зокрема, увага науковців прикута до адаптивності програмно-керованої мережі в умовах обслуговування мультисервісного трафіку, яка реалізується завдяки можливостям протоколу OpenFlow здійснювати оперативний моніторинг структурно-функціональних параметрів пристроїв передачі даних та динамічно їх програмувати.

В рамках аналізу складових елементів системи управління, що забезпечують якість обслуговування потоків в програмно-конфігурованих мережах, встановлено наступні недоліки.

Незважаючи на суттєво вищий рівень оперативності обчислення маршрутів та аналіз топології мережі, маршрутизація здійснюється за допомогою традиційних алгоритмів, що використовуються існуючими протоколами, зокрема: OSPF та EIGRP.

Різнорідність апаратної реалізації пристроїв передачі даних призводить до того, що різні комутатори можуть не підтримувати певні функції, або ж підтримувати їх з обмеженою продуктивністю. У процесі роботи мережі це може суттєво вплинути на пропускну здатність окремих потоків або ж цілих доменів мережі.

Маршрутизація потоків здійснюється за критерієм якості обслуговування або за критерієм рівномірного завантаження мережевих ресурсів. Підвищення якості обслуговування зазвичай здійснюється з врахуванням класифікації трафіку згідно ITU-T, що суттєво обмежує можливість управління трафіком. Разом з тим, більшість моделей не враховують характеристик мультисервісного трафіку та призводять до суттєвого погіршення якості обслуговування і підвищення ймовірності блокування каналів.

Друга категорія методів базується на потокових аналітичних моделях для оптимізації мережі. Проте ці методи використовують моделі балансування навантаження потоків, які не враховують чутливості трафіку до перемішування

порядку пакетів, погіршення джиттеру чи затримки й тимчасового розриву з’єднання. Відсутність можливості здійснювати диференційоване управління трафіком для окремих потоків окремих клієнтів та врахувати їхні вимоги щодо якості призводить до низької ефективності маршрутизації, неоптимального розподілу навантаження, погіршення якості обслуговування високо пріоритетних потоків.

Засоби контролю за процесом передачі окремих потоків відсутні, внаслідок чого система управління не має змогу визначити погіршення якості обслуговування для цих потоків, а тому не може гарантувати рівень якості, узгоджений у сервісному договорі SLA.

Недоліками існуючих методів моніторингу є:

- низька точність відтворення характеристики процесів функціонування комутаторів та передачі даних;
- неможливість проведення моніторингу тих параметрів, які є прихованими від інтерфейсу OpenFlow розробниками пристроїв (завантаження центрального процесора, розміщення правил всередині комутатора);
- відсутність методів моніторингу параметрів якості обслуговування між вхідним та вихідним вузлом довільного потоку;
- відсутність єдиної моделі системи моніторингу параметрів функціонування ПКМ.

РОЗДІЛ 2. МОДЕЛІ ТА АЛГОРИТМИ ОБСЛУГОВУВАННЯ ІНФОРМАЦІЙНИХ ПОТОКІВ У ПРОГРАМНО-КОНФІГУРОВАНИХ МЕРЕЖАХ

У розділі запропоновано моделі управління процесами передачі даних та метод моніторингу в програмно-конфігурованій мультисервісній мережі для підвищення якості обслуговування, уникнення перевантажень та підвищення рівномірності використання мережевих ресурсів. Зокрема, розроблено систему моніторингу параметрів функціонування мережевих пристроїв та метод моніторингу з адаптивними параметрами опитування. Розроблено метод вимірювання затримки окремого потоку в агрегованому потоці окремого класу для індивідуального моніторингу параметрів якості обслуговування потоків та подальшого їх підвищення з використанням удосконаленої потокової моделі маршрутизації. Для пропорційного розподілу трафіку в умовах перевантаження каналів удосконалено модель балансування навантаження. Наведені в розділі результати опубліковано у працях [74-77; 80; 83; 86; 87; 89; 90; 94; 95; 97; 102; 105; 107].

2.1. Розроблення системи моніторингу параметрів функціонування програмно-керованої мережі

Найважливішим завданням для оперативного управління мережею та перспективного планування її розвитку є облік використання ресурсів мережі на різних часових інтервалах (година, доба, тиждень, місяць) різними групами користувачів. Інше важливе завдання полягає в оперативному відстеженні стану мережі та її компонентів з метою виявлення аномальної поведінки, яка може бути наслідком атак на мережу або порушенням порядку використання мережевих ресурсів абонентами. Беручи до уваги високу цінність мережевих ресурсів (в першу чергу - пропускної здатності каналів), слід, перш за все, забезпечити можливість оперативного аналізу нерегулярностей та аномалій. У

зв’язку з цим систему моніторингу слід ще розглядати як важливий компонент системи забезпечення безпеки корпоративної мережі.

Системи моніторингу мережі повинні розроблятися також з урахуванням можливих перешкод різного виду (складність вимірювання, неточність результатів) та безпосередніх атак на ці системи. Через очевидну недостатність традиційної класифікації портів за протоколами/номерами необхідно залучати додаткову інформацію вищих семантичних рівнів. У сучасних мережах практично повсюдно використовується протокол SNMP, який дійсно є простим за своєю суттю, але потребує додаткових ресурсів для забезпечення постійного збору, зберігання, аналізу даних і візуалізації результатів цього аналізу. У мережах досить великого масштабу висока складність та поверхневий підхід до побудови системи збору та аналізу статистики призводить до неефективних, немасштабованих рішень.

Завдання дослідження характеристик інтенсивності потоків даних у сучасних корпоративних мережах при завантаженні в сотні і більше Мбіт/с є нетривіальним і вимагає ретельного вивчення.

Спрощені підходи до аналізу трафіку і спроби отримання інформації про мережеві додатки на підставі легкодоступних атрибутів та характеристик потоків даних найчастіше виявляються непродуктивними. Так, в більшості програмах обміну файлами активно використовуються прийоми, що ускладнюють достовірну ідентифікацію таких додатків (динамічне призначення портів, децентралізовані сховища, використання поширених прикладних протоколів як транспортних, а також криптографічні методи приховування та маскування трафіку). Тому для виявлення мережевих аномалій необхідний ретельний аналіз. Ці обставини вимагають використання додаткових технічних, програмних та алгоритмічних засобів і математичних моделей досліджуваних процесів.

Характеристики функціонування програмно-керованої мережі, що складається з різнорідних OpenFlow комутаторів, залежать від апаратних характеристик кожного комутатора. З метою зменшення складності та вартості

комутатора постачальники можуть обмежити функціональні можливості OpenFlow, наприклад, розмір апаратної таблиці потоків або можливостей цієї таблиці щодо пошуку за певними полями чи виконання певних дій над пакетами. Додатки, що встановлюються на контролері, повинні явно враховувати ці обмеження, щоб уникнути можливих проблем, пов'язаних з погіршенням продуктивності мережі в процесі її роботи. У цій роботі ми розробляємо систему моніторингу для тестування продуктивності та якості обслуговування потоків у програмно-керованих мережах на етапі їхнього функціонування. Варто також зазначити, що ця система може бути використана як платформа для розробки автоматизованих тестів, специфічних для апаратної реалізації того чи іншого пристрою або топології мережі.

Основна мета полягає в забезпеченні централізованого моніторингу параметрів мережевих пристроїв і формуванні характеристик функціонування мережі на виході. Результати моніторингу можуть бути додатково використані при плануванні мережі або для динамічної оптимізації мережі, наприклад, встановлення правил з урахуванням апаратних характеристик комутатора й передачі трафіку через різні шляхи для уникнення вузьких місць у мережі.

Специфікація OpenFlow визначає необхідну функціональність, яка повинна бути реалізована в кожному комутаторі OpenFlow. Метою цієї специфікації є забезпечення уніфікованих процесів контролю та управління мережею, що складається з пристроїв різних моделей від різних виробників. Проте реалізація необхідної функціональності є індивідуальною та змінюється від виробника до виробника [16] і, таким чином, може привести до змін продуктивності в реальній мережі. Як правило, проблеми виникають уже тоді, коли мережа налаштована й окремі комутатори починають створювати вузькі місця, погіршуючи продуктивність та якість обслуговування. Без сумніву, що такі недоліки можна врахувати на стадії проектування мережі та розробки програмного забезпечення для контролера. Для цього необхідно, щоб розробники додатків для ПКМ змогли отримати інформацію про обмеження продуктивності конкретного комутатора з метою забезпечення стабільної та

надійної роботи мережі. Завдяки запропонованій системі розробники зможуть отримати необхідні дані та можливість охарактеризувати продуктивність конкретного комутатора.

Комутатор, залежно від апаратної реалізації, може надавати доступ до окремих параметрів (повідомлення типу ofp_capabilities). Специфікація OpenFlow містить загальний список параметрів доступних для моніторингу:

- статистика потоків (тривалість існування, пріоритет, кількість оброблених пакетів та байт);
- статистика таблиць потоків (кількість правил, кількість оброблених пакетів та байт);
- статистика портів (кількість переданих/відкинутих пакетів/байт, помилок передачі та колізій);
- статистика черг (довжина черги, кількість переданих пакетів/байт, кількість відкинутих пакетів через переповнення);
- інші (STP, збірка сегментів IP пакетів).

Основна мета полягає в тому, щоб створити простий, легкий у використанні інструмент порівняльного аналізу, який може бути основою для написання тестових сценаріїв користувачем. Тому в роботі розроблено систему моніторингу параметрів функціонування комутатора OpenFlow з використанням мови програмування *Python* [60].

Розроблена система дозволяє здійснювати моніторинг як OpenFlow параметрів, так і параметрів QoS, пов'язаних з роботою комутатора. До параметрів QoS належать: пропускна здатність, затримка, джиттер і втрати. До параметрів OpenFlow відносять: максимальну кількість правил з конкретною структурою, тривалість встановлення правила, тривалість опитування статистики таблиці потоків, тривалість обробки пакетів з використанням програмних таблиць потоків, тривалість зчитування інформації з лічильників, а також завантаження центрального процесора.

На рис.2.1 відображено архітектуру програмно-керованої мережі з інтегрованою системою моніторингу.

Рис. 2.1. Архітектура програмно-керованої мережі з інтегрованою системою моніторингу

Основним компонентом розробленої системи моніторингу є додаток моніторингу, встановлений на фізичному сервері. Цей додаток виконує ключову роль збору, форматування та представлення інформації у форматі, зручному як для людини, так і для подальшої обробки цієї інформації іншими додатками, наприклад, контролером. База даних зберігає інформацію, необхідну для роботи центрального додатка, наприклад, налаштування комутаторів, моделі та технічні особливості того чи іншого комутатора. Ці дані заносить в базу даних адміністратор мережі. У базі даних також зберігаються зібрані дані моніторингу та оброблені статистичні характеристики поведінки мережі в певні періоди.

Для збору даних можуть використовуватися декілька каналів зв'язку як з комутаторами, так і з контролером. Зв'язок з комутаторами може відбуватися через віддалене з'єднання з операційною системою комутатора на основі протоколу SSH або через вторинний OpenFlow канал за допомогою додатку OVS-OFCTL, що є частиною пакету OpenVSwitch. Комунікація з контролером може відбуватися безпосередньо за рахунок інтеграції його в контролер або ж за допомогою REST API, що на сьогодні надається більшістю існуючих контролерів.

База даних містить дві категорії даних. До першої категорії належать дані для конфігурації системи моніторингу, до другої – дані, зібрані системою моніторингу з мережі. Модель даних для конфігурації системи моніторингу відображено на рис.2.2.

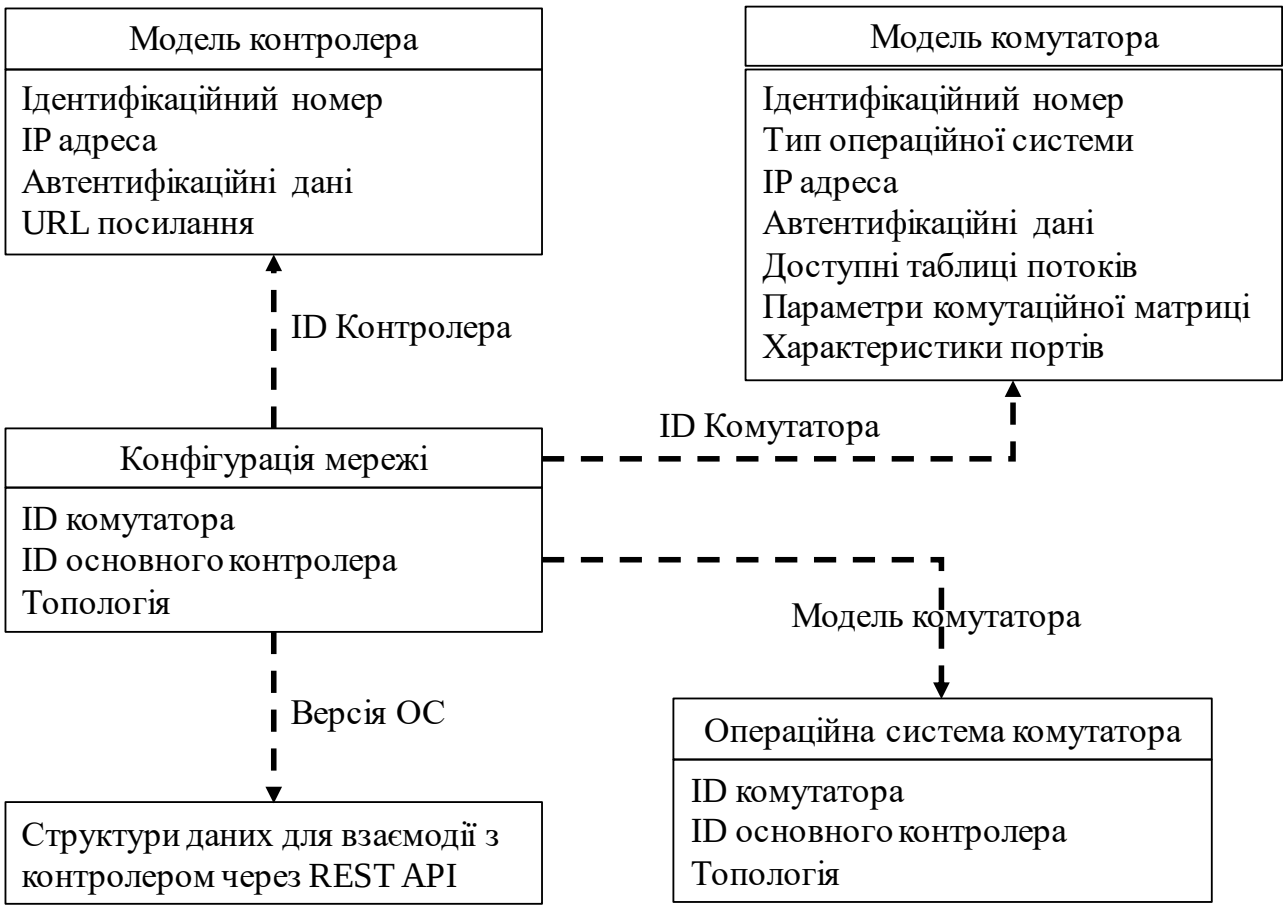


Рис. 2.2. Модель даних для конфігурації системи моніторингу

Наведена модель відображає структуру інформації про комутатори та контролери, які є елементами окремої мережевої топології. Ця інформація заноситься адміністратором мережі. У ПКМ кожен комутатор повинен мати власний ідентифікаційний номер довжиною 48 біт. У результаті аналізу структури даних «Конфігурація мережі» система моніторингу отримує на основі зазначених ідентифікаторів всю необхідну інформацію як про комутатори, так і про контролери.

У процесі роботи система моніторингу зберігає всю зібрану інформацію у базі даних. Варто зауважити, що, у разі невеликих обсягів даних, частину даних про попередній стан мережі система моніторингу зберігає в своїй кеш пам’яті.

Це потрібно для того, щоб пришвидшити аналіз динаміки зміни стану мережі протягом невеликих інтервалів часу.

Приклад структури даних у форматі JSON для контролера та комутатора відображено в табл.2.1.

Таблиця 2.1

Приклад заповненої структури даних для конфігурації комутатора та контролера

datapath2 = {
"ip":"10.1.1.15",
"user":"manager",
"password":"",
"openflow_instance_name":"test01",
"cli_name":"sw1#",
"dpid":"00:64:74:46:a0:5f:1e:80",
"model":"3500",
"software_tables":["200"],
"hardware_tables":["100"]}
	controller2 = {
	"id":"00:00:00:00:01",
	"ip":"10.0.1.17",
	"user":"sdn",
	"password":"skyline",
	"URL":"ip:5544/sdn/login",
	"OS":"2.5.1",
	}

Ця інформація також може надаватися контролеру на вимогу, а то й повністю керуватися контролером у випадку інтеграції системи моніторингу та контролера. Модель даних для зберігання поточного стану мережі відображено на рис.2.3.

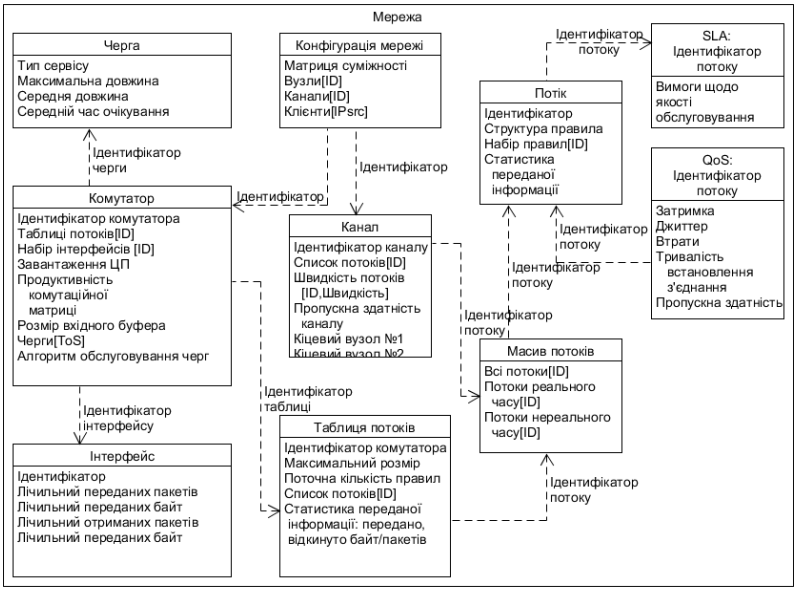


Рис. 2.3. Модель даних для зберігання поточного стану елементів мережі

Розглянемо архітектуру центрального додатку моніторингу, представлену на рис.2.4. Інформація про комутатори (IP-адреса, SSH ім'я користувача й

пароль, ідентифікаційний номер комутатора, модель, ідентифікатори таблиць потоків тощо) зберігаються в компоненті «Конфігурація мережі». Цей компонент також містить набори моделей потоків, які сумісні з програмно-керованими комутатори конкретної моделі. Для контролю за параметри комутатора використовується один з компонентів «Монітор» залежно від типу параметра. У випадку моніторингу через SSH «Екземпляр мережі» використовує монітор SSH, який на основі ідентифікатора комутатора отримує дані для формування об'єкта «Віддалений доступ». Цей об'єкт базується на бібліотеці для роботи з терміналом та містить методи для управління й моніторингу комутатора з використанням інструкцій операційної системи конкретного комутатора. Таким чином, система моніторингу може бути розширена та доповнена іншими моделями комутаторів.

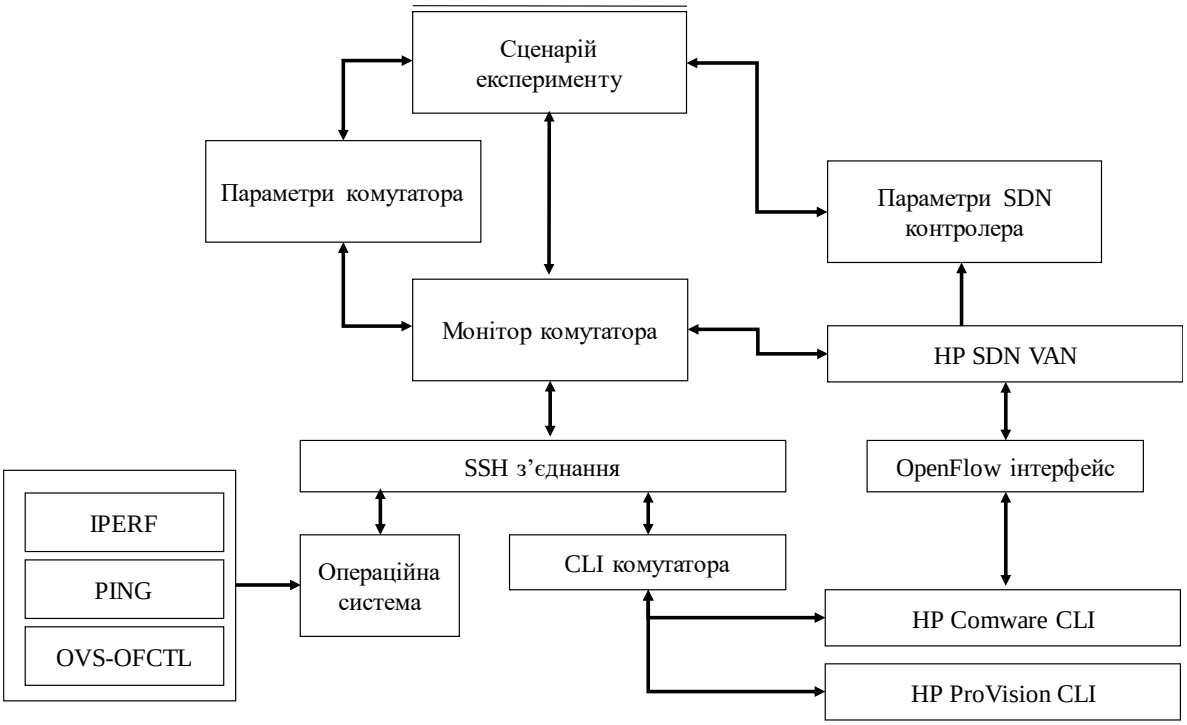


Рис. 2.4. Архітектура центрального додатка моніторингу програмно-керованої мережі

Для маніпуляцій OpenFlow параметрами комутатора додаток може використовувати два способи. Перший забезпечується шляхом інтеграції системи з контролером або ж може використовувати інтерфейс REST API для комунікації з контролером. Перевагою використання цього каналу є можливість

отримати як необроблену інформацію про потоки даних та лічильники, так і більш складну інформацію, зокрема: зв'язки між комутаторами чи найкоротший шлях між двома вузлами в мережі. Недоліком використання цього способу є необхідність створення моделей запитів та відповідей у форматі JSON. Відповідно до другого способу, таблицями потоків можна керувати безпосередньо за допомогою додатку *ovs-ofctl*. У разі, коли система моніторингу вбудована в контролер, він може безпосередньо використовувати методи запущеного додатку SDN для отримання інформації про конкретний комутатор чи маніпулювання таблицею потоків.

2.2. Модель адаптації системи моніторингу до властивостей процесу передавання даних

Для забезпечення процесів управління мережею, побудованою за основними принципами SDN, необхідно здійснити модернізацію механізмів моніторингу стану її ресурсів, оскільки при використанні стандартного механізму спостерігається генерація надлишкової службової інформації для елементів, що простоюють. Цей факт може мати негативний вплив на ефективність роботи мережі загалом через завантаженість каналів.

У зв’язку з недоліками існуючих систем моніторингу, визначеними в першому розділі роботи, пропонуємо використовувати метод динамічної адаптації параметрів системи моніторингу. Наприклад, зміну інтенсивності моніторингу стану мережевого елемента залежно від його поточної завантаженості. Інтелектуальний моніторинг на мережі реалізується шляхом встановлення головного додатку моніторингу на контролер та відповідних підконтрольних йому агентів на кожен комутатор.

Процес моніторингу ґрунтується на двох основних підходах:

- перший – зміна інтенсивності моніторингу залежно від попередньо накопиченої статистичної інформації про стан завантаженості в пам’яті керуючого пристрою. На основі аналізу зібраної інформації можна визначати характеристики трафіку в певний період часу та змінювати інтенсивність

відповідно до визначених потреб. Наприклад, у години пікового навантаження (як правило, під час ранкового робочого періоду) слід збільшувати інтенсивність моніторингу, а вночі можна її зменшити;

- другий – динамічна зміна інтенсивності моніторингу залежно від завантаженості елемента мережі на основі зібраних у режимі реального часу даних стану елемента мережі. У випадку наближення завантаженості до критичного значення збільшується частота опитування мережевих вузлів, таким чином здійснюється перехід у стан пильного моніторингу відповідного елемента мережі.

Аналіз моніторингу виконує додаток, встановлений на контролері, а підконтрольні йому агенти виконують функції збору інформації, її модифікацію та відправлення на контролер. Диференціація інтенсивності моніторингу окремих сегментів мережі дає можливість усунути надлишкові процеси моніторингу та обробки даних, виконуючи необхідну інтенсивність лише на тих вузлах, де це необхідно. На основі диференціації інтенсивності моніторингу можна збільшити гнучкість управління елементами мережі, та, як наслідок, ефективніше використовувати обчислювальні ресурси пристроїв рівня управління. Також, володіючи актуальною службовою інформацією, можна здійснювати оптимальний розподіл навантаження на мережі, попереджуючи таким чином негативні явища перевантаження мережевих вузлів. Окрім того, можна переводити елементи, що простоюють, у режим очікування, зберігаючи при цьому можливість надійного і швидкого їх відновлення до нормального режиму роботи. Це значно зменшує витрати на електроенергію та збільшує час «життя» мережевих пристроїв. Повністю відключати обладнання не завжди доцільно, оскільки його ввімкнення потребує певного часу для відновлення таблиць комутації, запуску всіх апаратних та програмних процесів комутатора, що загалом впливає на час перебудови топології мережі та спричиняє стрибок службової інформації між контролером та комутатором.

Особливістю розглянутого механізму є зміна інтенсивності моніторингу лише конкретного мережевого вузла, який цього потребує, однак це не впливає

на інтенсивність моніторингу інших вузлів. Пропонований механізм є адаптованим до сучасного динамічного трафіку та забезпечує хорошу базу для подальшого процесу балансування навантаження. Для прикладу, чим вище миттєве значення завантаження каналу, тим швидше відбудеться наступне опитування. Тобто, чим більше завантаження каналу, тим частіше система моніторингу здійснює опитування щодо кількості переданої інформації. Такий метод можна використовувати стосовно усіх параметрів комутатора у випадку, коли підвищення інтенсивності їхнього опитування не впливає негативно на характеристики продуктивності та якості функціонування пристрою.

Ще одним важливим параметром процесу моніторингу є кількість інформації, що генерується цим процесом. У випадку підвищення інтенсивності моніторингу кількість службової інформації і, як наслідок, навантаження на інформаційний канал зростатиме, що загалом призведе до часткового зниження ефективності використання інформаційного каналу. Функція зміни інтенсивності моніторингу буде різною для кожного окремого параметру, а її максимальне значення залежатиме від рівня навантаження процесу моніторингу на мережеві елементи. Обмеження щодо частоти моніторингу деяких параметрів часто встановлюють виробники комутаторів, вказуючи критичне значення в документації до пристроїв. Емпірична модель адаптації системи моніторингу, зокрема частоти опитування комутатора, до завантаження мережних каналів (1):

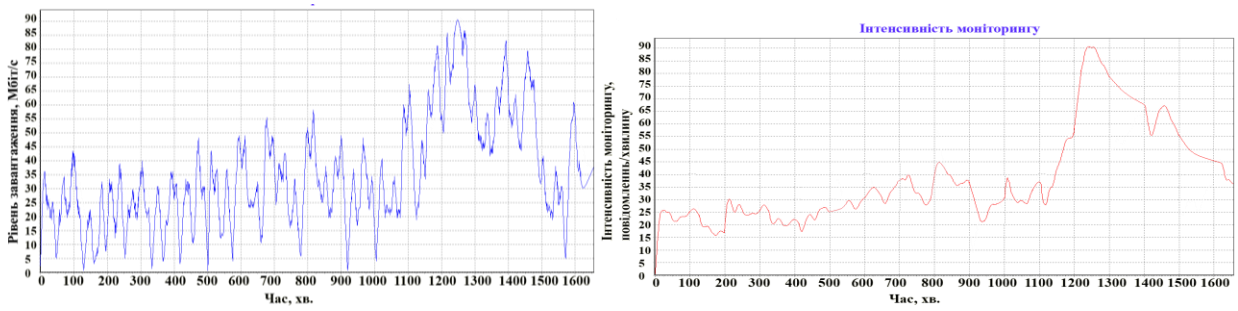
$$f(\rho_i) = \begin{cases} 1 - \dfrac{\rho_i}{2}, & 0 < \rho_i \le 0.5; \\ \dfrac{0.1}{(\rho_i - 0.367)}, & 0.5 < \rho_i \le 1, \end{cases} \tag{2.1}$$

де f – функція завантаженості, що використовується для визначення інтервалу опитування комутатора; ρ – завантаження інтерфейсу комутатора, безрозмірний коефіцієнт; i – порядковий номер опитування.

Всі параметри поділяються на декілька категорій. До першої категорії належать параметри, що фіксуються постійно: завантаження каналу,

завантаження комутатора. Ці параметри є критично важливими, оскільки перевантаження вказаних елементів призведе до відмови в обслуговуванні, суттєвих втрат інформації та погіршення якості. До другої категорії відносяться параметри, які необхідно додатково спостерігати у випадку, якщо один з параметрів першої категорії досягнув критичного рівня. Наприклад, у разі перевантаження одного з каналів слід додатково отримати інформацію щодо кожного потоку, який проходить через цей канал (швидкість потоку, втрати). Після цього на основі пріоритетів потоку здійснити моніторинг параметрів якості обслуговування для потоку реального часу. Якщо поточне значення параметрів якості обслуговування не відповідає вимогам, встановленим для цього потоку, тоді інформація про потік передається в частину системи керування, що відповідає за перерозподіл потоків у мережі.

На рис.2.5 показано адаптацію частоти моніторингу до завантаженості мережевого елемента. У разі збільшення величини *x* та падіння навантаження на елемент мережі до 70% інтенсивність процесу моніторингу стабілізується до нормального режиму. Цей механізм збільшуватиме інтенсивність моніторингу не мережі загалом, а лише проблемних ділянок мережі, що не спричинятиме стрімкого росту службової інформації, а також не потребуватиме великої обчислювальної потужності контролера, отже, не перевантажуватиме його.



а б

Рис. 2.5. Завантаженість мережевого елемента (а) та зміна інтенсивності моніторингу згідно даної завантаженості (б)

Із наближенням інтенсивності навантаження до пікового значення для даного елементу мережі, здійснюється ряд запобіжних заходів, зокрема: прогнозування, а також приведення в нормальний режим роботи резервних

ресурсів мережі. Такі рішення дають можливість швидко та адекватно реагувати на певні критичні випадки, що виникатимуть на транспортній мережі, максимально збільшуючи при цьому стійкість системи до відмов.

2.3. Метод вимірювання затримки передавання пакетів для потоків окремого користувача

Для сервісів потокового характеру втрати пакетів або зміна порядку надходження пакетів спричиняє погіршення якості обслуговування, що проявляється у формі завмирання та спотворення кадру відео чи фрагменту аудіо даних. Для того, щоб забезпечити належний рівень якості обслуговування, а відповідно і сприйняття, моніторинг параметрів якості обслуговування необхідно проводити не окремо на кожному комутаторі, а загалом з моменту входження трафіку в мережу до моменту його виходу з неї.

Методи моніторингу трафіку завжди були актуальними серед науковців у сфері телекомунікацій. Зазвичай, він зводиться до вимірювання обсягів і характеристик трафіку для окремої мережі. На сьогодні велика кількість методів використовується для вимірювання завантаження каналу, затримки передачі пакету з кінця в кінець, а також втрат пакетів. Архітектура ПКМ, завдяки централізованій площині керування, дає змогу управляти багатьма потоками та приймати складні рішення в єдиній логічній точці системи.

Запропонована система моніторингу дає змогу оцінити параметри якості обслуговування для довільних потоків у програмно керованих мережах. Основна ідея вимірювання – використовувати додаток моніторингу як опорну точку. Для цього в мережу вводиться тестовий пакет, заголовок якого відповідає заголовку пакетів, що належать цьому потоку. Пакет вводиться в мережу на комутаторі, де потік входить у мережу, і виводиться з площини даних на комутаторі, де потік покидає мережу. При цьому корисне навантаження тестового потоку містить дві важливі змінні: перша змінна включає ідентифікатор потоку; друга – момент введення потоку в мережу. Тестовий пакет періодично вводиться у мережу з наперед визначеним

інтервалом, що залежить від рівня завантаження каналу та швидкості потоку, для якого вимірюється затримка. Таким чином, пакет передається шляхом проходження всіх пакетів цього потоку і зазнає такої самої затримки. Схему вимірювання затримки потоку відображено на рис.2.6.

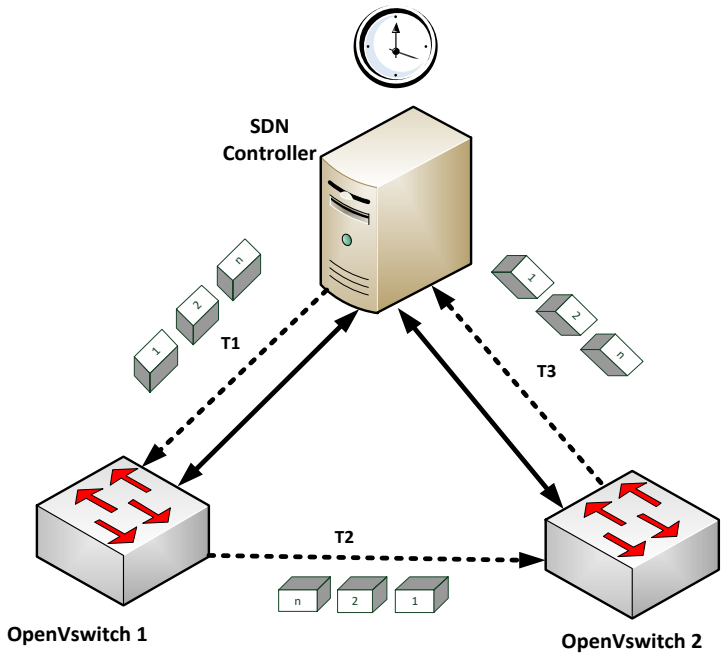


Рис. 2.6. Схема вимірювання затримки окремого потоку в програмно-керованій мережі

На останньому комутаторі тестовий пакет виводиться із площини передачі даних і відправляється в систему моніторингу. Система моніторингу визначає затримку цього пакету як різницю між часом його відправлення, отриманим з корисного навантаження цього пакету, та часом надходження, який фіксується в момент його отримання додатком моніторингу.

Алгоритми моніторингу параметрів якості обслуговування представлено на рис.2.7. Вся інформація про затримки потоків зберігається в спеціалізованій таблиці. Використовуючи ці дані, інші процеси на контролері мають змогу здійснювати управління трафіком та оптимізацію якості обслуговування для окремих потоків.

Перший алгоритм слідкує за розкладом моніторингу потоків, які були внесені контролером у список таких, що потребують оцінки параметрів якості обслуговування. У випадку, якщо такий список не порожній, алгоритм

відправляє тестові пакети в мережу та записує їхні копії в спеціальну чергу. Ці копії використовуються другим алгоритмом для співставлення з отриманими пакетами з мережі.

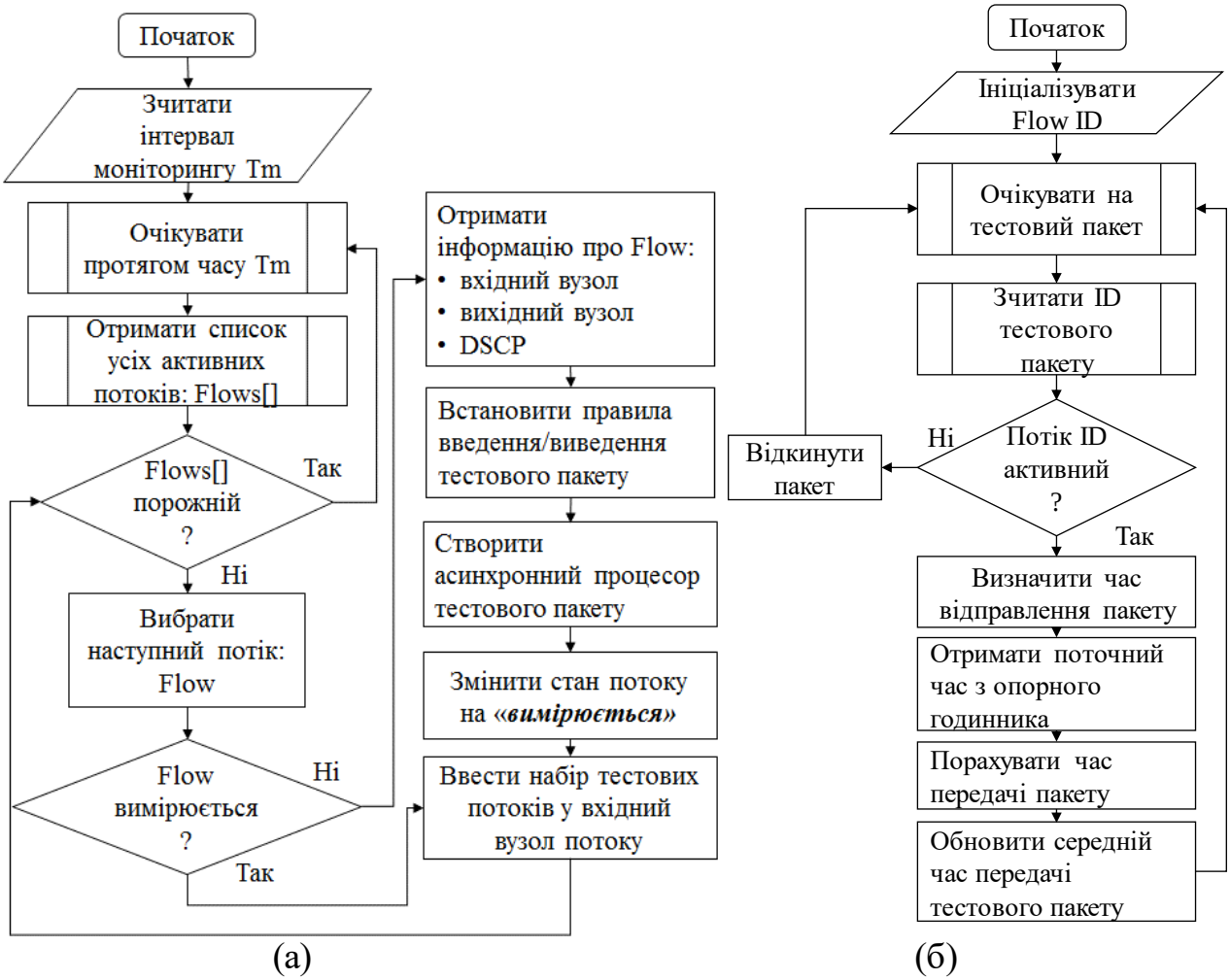


Рис. 2.7. Алгоритми відправки (а) та очікування (б) пакету для вимірювання затримки потоку

Особливістю запропонованого алгоритму є те, яким саме чином тестовий пакет передається по мережі. Для цього можуть використовуватися додаткові правила, що встановлюються для комутації такого пакету вздовж шляху, для якого вимірюється затримка. Додаткове правило встановлюється на кожному комутаторі. У запропонованому методі додаткове правило встановлюється тільки на комутаторі, де потік покидає мережу. На всіх інших вузлах пакет порівнюється з тим же ж правилом, з яким порівнюються всі інші пакети потоку. Тому, щоб відокремити тестовий пакет на вихідному комутаторі з потоку, пропонується структура правила, подана в табл. 2.2.

Таблиця 2.2

Приклад правила в таблиці потоків для вимірювання затримки потоку

Порт	DST MAC	SRC MAC	ETH TYPE	IP TOS	DST IP	SRC IP	IP PROTO	TCP/ UDP SRC	TCP/ UDP DST
*	*	*	*	*	*	*	*	*	40000/4

Відомо, що програмно-керовані комутатори мають змогу порівнювати поля заголовків пакетів з полями правил, використовуючи маску. Це, в першу чергу, застосовується для реалізації функцій маршрутизації, коли комутатор порівнює не всю IP адресу, а лише її мережну частину, ігноруючи при цьому біти, що відповідають за адресу станції. Для цього, встановлюючи правило, контролер явно задає комутатору кількість біт адреси, які повинні ігноруватися. Саме ця властивість використовується для вилучення тестових пакетів з потоку. Порівняння на основі TCP чи UDP портів застосовується в мережах вкрай рідко, переважно на крайових вузлах. Такий базовий аналіз протоколів транспортного рівня дає змогу створити найпростіший Firewall на межі мережі та заблокувати проходження окремих потоків, наприклад, заблокувати пакети, які приходять з певної IP адреси та мають встановлені певні порти.

У роботі пропонуємо використовувати діапазон портів від *40000* до *50000*, маскуючи останні чотири цифри порту. Коли проводиться вимірювання затримки для конкретного потоку, система моніторингу вибирає для нього порт із зазначеного діапазону та створює відповідне правило, яке повністю дублює правило для передачі цього потоку і порівнює тестовий пакет з вибраним портом.

Схему вимірювання затримки потоку в мережі відображено на рис.2.8. Усі тестові пакети, введені в мережу з цим портом, будуть виведені з мережі на кінцевому комутаторі шляху і передані на контролер або ж на систему моніторингу, якщо вона фізично відокремлена.

Зрозуміло, що виміряна затримка враховує тривалість передачі пакету між контролером та вхідним комутатором, а також тривалість передачі між контролером і вихідним комутатором. Тому з кожним десятим тестовим

пакетом контролер здійснює оцінку сигнального каналу, а саме – відправляє ICMP запит до обох комутаторів та отримує відповідь.

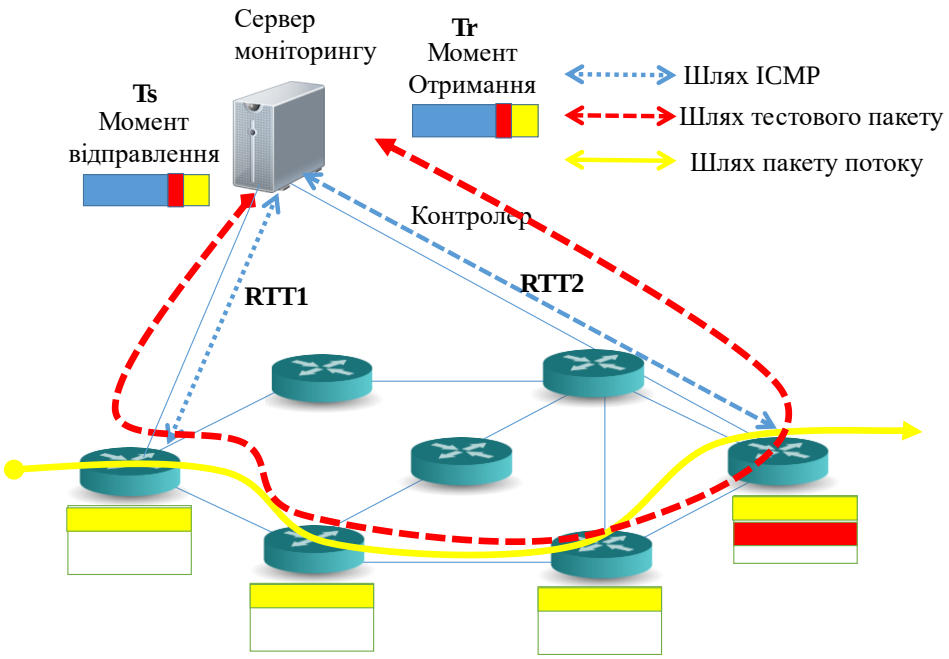


Рис. 2.8. Схема вимірювання затримки передачі окремого потоку в мережі

У результаті отримані значення враховуються при обчисленні затримки шляху таким чином:

$$Delay = t_s - t_r - (\frac{RTT_1}{2} + \frac{RTT_2}{2}) \quad (2.2)$$

де *Delay* – тривалість передачі тестового пакету між першим та останнім комутатором шляху потоку;

t_s – момент відправлення тестового пакету з контролера;

t_r – момент отримання тестового пакету контролером;

RTT_1 та RTT_2 – тривалості відгуку першого й останнього комутатора шляху на *ICMP* запит від контролера.

Джиттер потоку вимірюється як різниця зазначених затримок.

2.4. Алгоритм перерозподілу трафіку на основі відносного пріоритету потоку

Технологія програмно-керованих мереж дає змогу ідентифікувати потік за більшою кількістю ознак, а саме: номером вхідного порту, адресою канального

рівня, міткою віртуальної локальної мережі, пріоритетом у віртуальній локальній мережі, типом мережевого протоколу (*ARP*, *ICMP*, *IP*), адресою мережевого рівня (*ToS* – типом сервісу у випадку, якщо протокол мережевого рівня IP), типом транспортного протоколу (*UDP*, *TCP*), значеннями портів транспортного протоколу. Специфікація *OpenFlow* також передбачає ідентифікацію за міткою *MPLS*. Аналіз пакету за перерахованими ознаками дає змогу керувати окремими потоками, що належать окремим користувачам, не впливаючи при цьому на всі інші потоки. У такому разі швидкість потоку не має значення, а важливі тільки параметри, що його ідентифікують у мережі.

Для гнучкого управління трафіком важливо врахувати взаємозв'язок між типом трафіку (наприклад, реального часу чи передачі даних з файлового сервера) та мережевих параметрів, зокрема таких: доступна пропускна здатність, втрати пакетів, джиттер та затримка. Аналізуючи типи трафіку, варто зауважити такі моменти:

- VoIP трафік є дуже чутливим до втрат; в ідеалі втрати повинні бути відсутніми, особливо, коли використовується кодек з компресією;
- максимально допустима затримка потокового відео залежить від розміру буфера кінцевих клієнтів, у зв'язку з чим вона може суттєво варіюватися;
- допустимий джиттер не є фундаментальним параметром для передачі потокового відео, а тому до нього немає строгих вимог;
- вимога потокового відео до пропускної здатності залежить від формату кодування та швидкості відеопотоку, а, отже, не є фіксованим значенням.

Більш того, як у мультимедіа, так і в додатках для телефонії, параметри якості обслуговування можуть залежати від типу кодека, механізму виправлення помилок та швидкості складових потоків. Зважаючи на ці зауваження, досить складно точно виміряти якість обслуговування з точки зору користувача, оскільки необхідно врахувати значну кількість різних змінних.

Вимоги щодо якості обслуговування певних класів трафіку, згідно з рекомендаціями ITU-T Y.1541, зведено в табл.2.3.

Таблиця 2.3

Класифікація вимог до параметрів якості обслуговування

Параметри функціонування мережі	Класи трафіку					
	Клас 1	Клас 2	Клас 3	Клас 4	Клас 5	Клас 6 (невизначений)
Затримка, мс	100	400	100	400	1000	0
Джиттер, мс	50	50	0	0	0	0
Коефіцієнт втрат, %	0,001	0,001	0,001	0,001	0,001	0
Коефіцієнт помилок, %	0,0001					0

Нульове значення параметра означає, що рекомендація не встановлює ніяких обмежень щодо цього параметра. Його значення може бути довільно вибраним самим оператором мережі. Рекомендований інтервал оцінки параметрів у мережі становить одну хвилину, проте оператори самі можуть вибирати цей інтервал.

З урахуванням рекомендації щодо важливості тих чи інших параметрів для окремих класів трафіку в роботі сформовано таблицю вимог щодо якості обслуговування існуючих сервісів, що функціонують у мережі Інтернет (табл.2.4).

Таблиця 2.4

Розподіл найпопулярніших сервісів за класами якості обслуговування

Клас	Характеристики	Приклад
1	Реального часу, чутливі до джиттеру, інтенсивна взаємодія	VoIP, Skype (аудіо, відео)
2	Реального часу, чутливі до джиттеру, інтерактивні	VoIP, Інтернет, телебачення
3	Транзакції, високоінтерактивні	HTTP, Web-Services
4	Транзакції, інтерактивні	Сигналізація (OpenFlow)
5	Тільки з низькою чутливістю до втрат	FTP, Cloud, потокове відео (Youtube)
6	Традиційні додатки IP мереж	Будь які інші дані

У результаті аналізу двох таблиць представлено зведену табл.2.5 вимог щодо параметрів якості обслуговування для конкретних додатків. Введемо два

додаткових критерії для класифікації потоків: перший критерій – чутливість до неправильного порядку отриманих пакетів; другий критерій – пріоритет окремого користувача.

Для розрахунку відносного пріоритету потоку для конкретного шляху спочатку знайдемо відносні коефіцієнти для кожного потоку. Відносні коефіцієнти розраховують за формулами (2.3-2.9), тобто, як відношення мінімальних значень параметрів якості обслуговування до поточних значень, отриманих у результаті моніторингу мережі.

Відносний коефіцієнт втрати пакетів:

$$P = \frac{P_{\min}}{P}. \tag{2.3}$$

Відносний коефіцієнт затримки пакетів:

$$t = \frac{T_{\min}}{T}. \tag{2.4}$$

Відносний коефіцієнт джитера:

$$j = \frac{J_{\min}}{J}. \tag{2.5}$$

Відносний коефіцієнт смуги пропускання:

$$c = \frac{C}{C_{\max}}. \tag{2.6}$$

Відносний коефіцієнт тривалості встановлення з’єднання:

$$t_c = \frac{Tc_{\min}}{T_c}. \tag{2.7}$$

Відносний коефіцієнт чутливості до перемішування пакетів:

$$p_m = \frac{Pm_{\min}}{P_m}. \tag{2.8}$$

Відносний коефіцієнт пріоритетності користувача:

$$c_p = \frac{C_p}{Cp_{\max}}. \tag{2.9}$$

Таблиця 2.5

Вимоги до QoS окремих додатків

Тип потоку	Параметри QoS						
	P, %	T, мс	J, мс	C, кбіт/с	Tc, мс	Pm	Cp
Голосові дані (VoIP, Skype)	<0,25	150	10	64	900	10^{-2}	2–256
Відеоконференція (Skype)	<2	100	20	2048	800	10^{-2}	256–512
IPTV	<1	100	50	4096	1000	10^{-2}	512–768
Інтернет дані (HTTP, Online services)	<0,1	100	100	2048	1000	10^{-6}	1280–1536
Інтерактивні дані (FTP, Cloud)	<1	400	500	256–10000	950	10^{-4}	768–1024
Медіа за запитом (Youtube)	<0,1	400	30	2048	700	10^{-4}	1024–1280
Сигналізація (OpenFlow)	0,01	50	50	64	500	10^{-5}	1

P – втрати пакетів;
T – затримка;00
J – джитер;
C – пропускна здатність;
Tc – тривалість встановлення з'єднання;
Pm – чутливість перемішування пакетів;
Cp – пріоритет клієнта

Результати розрахунків заносимо в табл.2.6.

Таблиця 2.6

Формування матриці відносних коефіцієнтів

Тип потоку	Параметри QoS					
	P	T, мс	J, мс	C, кбіт/с	Tc, мс	Pm
Голосові дані	0,1	0,667	1	0,0063	0,556	10^{-5}
Відеоконференція	0,0125	1	0,5	0,2	0,625	10^{-5}
IPTV	0,067	0,1	0,2	0,4	0,5	10^{-2}
Інтернет дані	0,1	0,1	0,01	0,2	0,5	10^{-2}
Інтерактивні дані	0,1	0,25	0,02	0,025	0,526	10^{-5}
Медіа за запитом	0,2	0,2	0,333	1	0,714	10^{-5}
Сигналізація	1	1	0,01	0,0063	1	1

Табл.2.7 заповнюється числами 1, 2 та 3, які відображають відповідно низьку, середню та високу значимість вимог до показників QoS [6].

Таблиця 2.7

Формування матриці значимості показників QoS

Тип потоку	Параметри QoS					
	P	T, мс	J, мс	C, кбіт/с	Tc, мс	Pm
Голосові дані	2	3	3	1	3	3
Відеоконференція	2	3	3	2	3	3
IPTV	3	2	2	3	3	3
Інтернет-дані	3	1	1	1	1	2
Інтерактивні дані	2	2	1	1	3	2
Медіа за запитом	2	2	2	3	3	2
Сигналізація	3	2	1	1	3	3

Ці параметри може визначати оператор. Більш того, кожному клієнту визначено пріоритет для кожного типу трафіку. Якщо пріоритет не заданий явно в договорі щодо надання сервісу, тоді за замовчуванням клієнтові присвоюють найнижчий пріоритет з усіх можливих.

Відповідно, відносний пріоритет для кожної категорії послуг визначається за формулою (2.9):

$$\mathrm{Pr}_i = \frac{\sum_{j=1}^{4} X_{ij} Y_{ij}}{\sum_{i=1}^{7} \sum_{j=1}^{4} X_{ij} Y_{ij}} \tag{2.9}$$

де Pr – відносний пріоритет i-ої послуги;

i – номер типу послуги;

j – номер параметра якості обслуговування;

X_{ij} – відносний пріоритет параметру j для послуги i;

Y_{ij} – важливість параметру j для послуги i.

Кінцевим результатом наведеної формули є дробове число, діапазон значень якого лежить в межах від нуля до одиниці. Чим більше значення, тим вищий пріоритет потоку. Формулу можна застосовувати для будь-якої кількості інформаційних потоків та різних вимог щодо якості обслуговування.

З урахуванням наведеної методики розрахунку відносного пріоритету потоків у роботі запропоновано спосіб підвищення ефективності використання ресурсів таблиці потоків, розмір якої обмежується апаратною реалізацією

комутатора. Апаратні комутатори можуть використовувати як апаратну, так і програмну таблиці потоків. Перша характеризується високою продуктивністю та стабільністю, проте має дуже обмежений розмір порівняно з програмною. Зазвичай розмір апаратної таблиці потоків для апаратних комутаторів рівня доступу та агрегації становить від 300 до 3000 потоків. При цьому ресурс таблиці може змінюватися залежно від типів правил, які використовуються. Якщо ж таблиця потоків переповнена, тоді неможливо встановити нове правило, а потік отримує відмову в обслуговуванні.

Зважаючи на зазначене, у роботі запропоновано розділити потоки мультисервісної мережі на три категорії (табл.2.8).

Без сумніву, що у великих мережах з обмеженим розміром таблиці потоків може використовуватися такий самий підхід, як і в IP мережах, де маршрутизація відбувається на основі IP адреси мережі призначення. Для цього всі інші поля в правилах таблиці потоків повинні мати значення «*». Приклад таблиці потоків та принцип співставлення ідентифікаторів потоку з правилами в таблиці відображено на рис.2.9.

Таблиця 2.8

Класифікація типів трафіку мультисервісної мережі

Тип трафіку	Додатки	Вимоги	Протоколи транспортного рівня
Реального часу	IP-телефонія та відеоконференцзв'язок	чутливість до затримок чутливість до джитеру затримки - мала чутливість до втрат	RTCP, RSVP, UDP, RTP
	Процеси управління Ігри on-line	чутливість до затримок чутливість до джитеру затримки - чутливість до втрат	UDP, TCP
Потоковий	Аудіо на вимогу Відео на вимогу Інтернет-мовлення	мала чутливість до затримок чутливість до джитеру - чутливість до втрат	RSVP, UDP, TCP, SCTP
Еластичний	Конференція документів	мала чутливість до затримок мала чутливість до джитеру затримок - велика чутливість до втрат	TCP
	Анімація Передача файлів E-mail	дуже мала чутливість до затримок мала чутливість до джитеру затримок - висока чутливість до втрат	

У роботі пропонуємо поєднати дві схеми обробки пакетів, взявши переваги традиційної таблиці маршрутизації та таблиці потоків OpenFlow, на основі чого розробити свій спосіб формування таблиці потоків у програмно-керованих мережах.

Для потоків третьої категорії використовуються правила, які ідентифікують пакет за принципом класичної таблиці маршрутизації, а саме: за адресою мережі призначення та класом трафіку. Для потоків другої та третьої категорії створюються окремі правила для кожного окремого потоку. Як наслідок, балансування навантаження потоків даних може здійснюватися досить гнучко в довільно визначених пропорціях, при чому досить швидко. У той самий час перерозподіл потоків реального часу є більш тривалим та потребує більше уваги до параметрів якості обслуговування цих потоків.

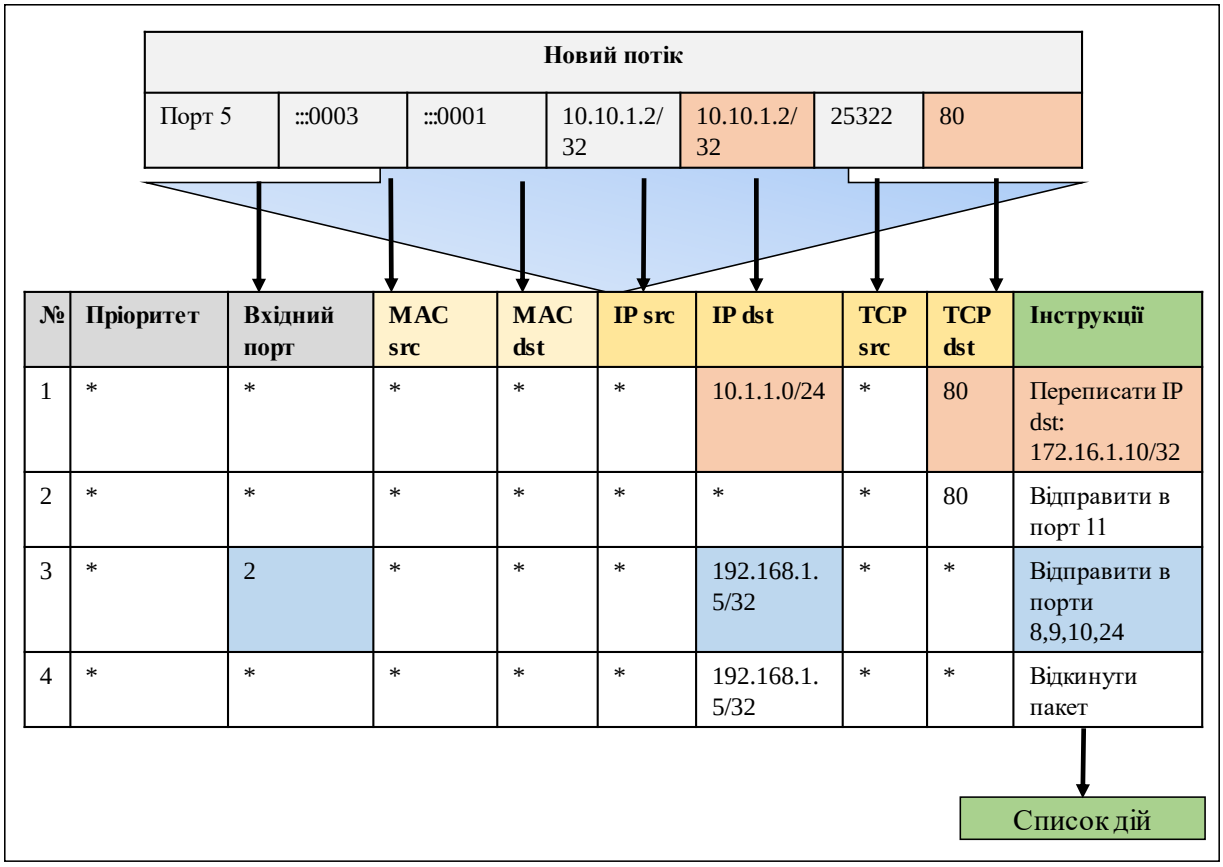


Рис. 2.9. Таблиця потоків у програмно-керованих мережах

Кожній сформованій категорії відповідає певна політика керування, згідно якої з потоками конкретної категорії дозволено чи заборонено виконувати певні

операції, а також зазначено умови, за яких ці дії можуть бути виконані. Наприклад, у випадку виникнення перевантаження певного шляху в певному комутаторі необхідно здійснити перерозподіл потоків. Ураховуючи те, що потоки першої категорії є чутливими до неправильного порядку пакетів та джиттеру, перенаправлення цих потоків може погіршити якість їхнього обслуговування. Тому розвантаження шляху починається з перерозподілу потоків третьої категорії. При чому, після кожної ітерації перерозподілу чи балансування відбувається вимірювання часових параметрів якості обслуговування спочатку першої категорії, а потім другої. Якщо параметри відповідають затребуваним, і шлях не є перевантаженим, тоді алгоритм перерозподілу завершує свою роботу.

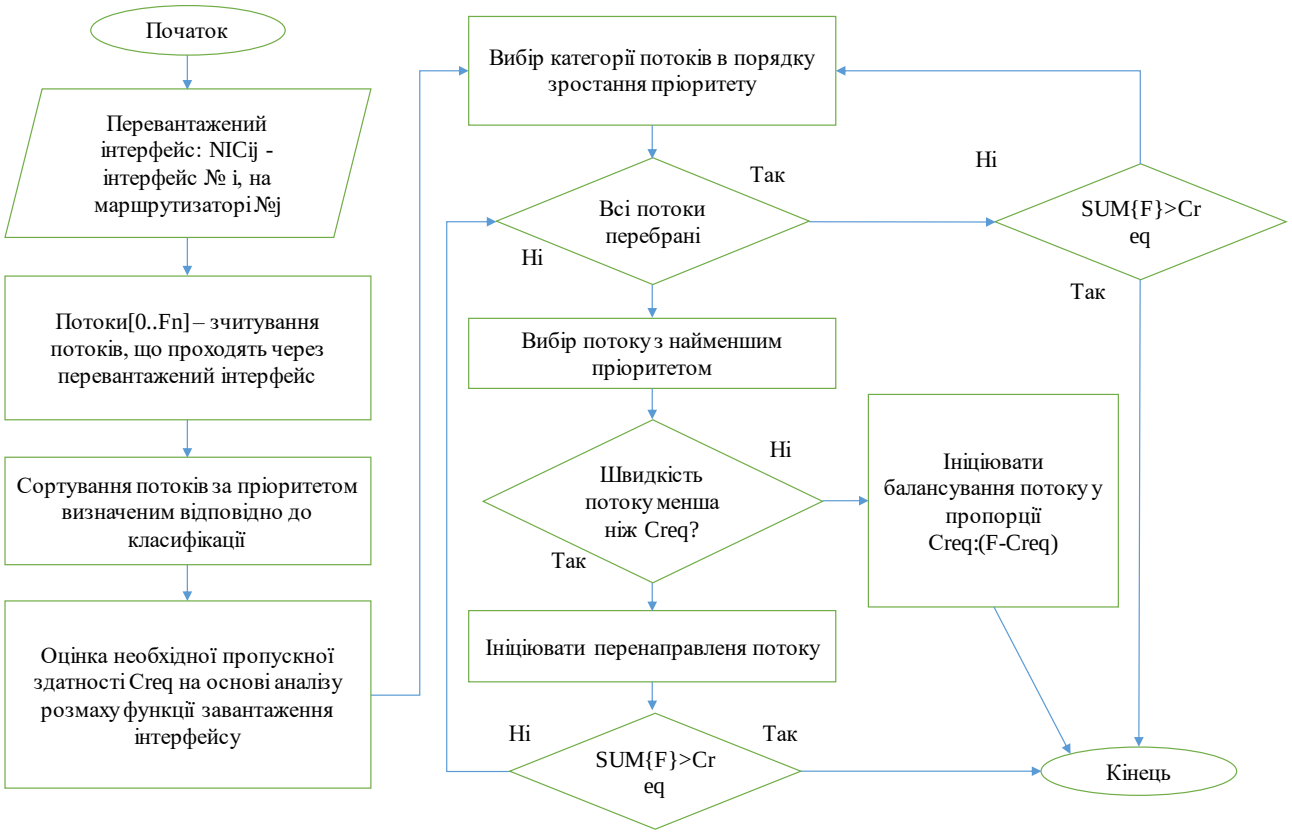


Рис. 2.10. Алгоритм перерозподілу потоків у мережі для уникнення перевантаження інтерфейсу

Важливим є те, яким саме чином визначити перевантаження конкретного шляху чи конкретного вузла. Враховуючи максимальну чутливість потоків

першої категорії до часових параметрів якості обслуговування, оцінювання якості шляху слід здійснювати саме на їх основі.

У випадку, коли мережа функціонує в нормальному режимі, без перевантаження чи вузьких місць, система моніторингу опитує мережеві пристрої зі стабільно великим інтервалом часу. Зокрема, система моніторингу опитує завантаження мережевих інтерфейсів, завантаження таблиці потоків комутатора та завантаження центрального вхідного буферу маршрутизатора, на якому ідентифіковано перевантаження інтерфейсів.

Система моніторингу опитує вузли з певним інтервалом часу та здійснює моніторинг часових параметрів передачі потоків першої категорії. У випадку середньостатистичного зростання завантаження інтерфейсу, затримки передачі та джиттеру система переходить у стан пильного моніторингу зазначеного інтерфейсу та підвищує інтенсивність вимірювання затримки. Крім того, коли рівень завантаження інтерфейсу сягає 0,9, система починає опитувати завантаження буферів пристрою.

У тому разі, коли середньостатистичне завантаження буферів потоку реального часу більше від 1, запускається алгоритм розвантаження інтерфейсу, описаний вище.

2.5. Удосконалення моделі маршрутизації потоків у програмно-конфігурованих мережах

Істотна кількість проблем у реальному світі може бути відтворена за допомогою мережевих моделей. Мережева модель може бути використана для відтворення потоків води, енергії, харчових ланцюгів, а також стану процесів виробництва. Складність мережевої моделі залежить від природи проблеми, що вирішується, та рівня деталізації моделі.

У роботі пропонується удосконалена модель маршрутизації, яка дає змогу балансувати навантаження за критеріями мінімального/максимального завантаження каналів мережі та якістю обслуговування для кожного потоку. Для цього використано класифікацію потоків, розглянуту в другому розділі,

задача маршрутизації яких звучить наступним чином. Потоки першої категорії повинні бути доставлені оптимальним шляхом за критерієм вартості, яка враховує параметри якості обслуговування: затримку та джиттер. Такі потоки є дуже чутливими до джиттеру, тому багатошляхова маршрутизація для них заборонена, тобто цілий потік може передаватися тільки одним шляхом. Потоки другої категорії є менш чутливими до часових параметрів якості обслуговування, а тому дозволяється здійснювати балансування таких потоків. Проте при балансуванні існує обмеження на мінімальний розмір підпотоку. Це обмеження необхідне у зв'язку з тим, що потоки другої категорії зазвичай є TCP потоками, які чутливі до втрат та зміни порядку надходження пакетів, а зі збільшенням кількості підпотоків та зменшенням їх розмірів імовірність втрат та перемішування внаслідок багатошляхової маршрутизації зростає. У такому разі потокам другої категорії гарантуються параметри якості обслуговування в межах допустимих. Потоки третьої категорії можуть бути доставлені будь яким шляхом без жодних гарантій якості обслуговування. Тому такі потоки використовуються для завантаження малозавантажених шляхів та вирівнювання розподілу навантаження між каналами у всій мережі.

Розроблена система моніторингу сприяє реалізації такого підходу за рахунок адаптивного моніторингу використання ресурсів каналів та пристроїв. Пошук оптимального шляху передачі здійснюється на основі математичної проблеми, яка використовується для вирішення двох загальновідомих проблем: Multi-Commodity Flow Problem and the Constrained Shortest Path Problem. Основною метою є знайти оптимальний набір маршрутів через мережу для всіх потоків з мінімальною сумарною вартістю. Основним обмеженням є те, що сумарна швидкість потоку через канал не може перевищувати пропускну здатність цього каналу.

Запропонована нами модель поєднує в собі обидві описані вище проблеми і дає змогу знайти для кожного потоку найкоротший шлях з урахуванням заданих обмежень. Нехай маємо мережу вузлів, де кожен канал характеризується затримкою, втратами та пропускною здатністю. Крім того,

кожен шлях характеризується вартістю, що розраховується як зважена сума затримки та втрат пакетів. У моделі робиться припущення, що для кожного типу сервісу/трафіку існує набір потоків, які можуть мати абсолютно різні як вхідні, так і вихідні вузли в мережі. Основною метою є знайти набір оптимальних шляхів у мережі для кожного потоку з мінімальною вартістю та з врахуванням деяких обмежень, зокрема таких: максимально допустима затримка, втрати пакетів та доступна пропускна здатність у каналах.

Нехай мережа представлена графом $G=(V, E)$, де V – набір вузлів, E – набір дуг між кожною парою вузлів. Дуги, тобто канали, характеризуються доступною пропускною здатністю b_{ij}, затримкою d_{ij}, втратами пакетів p_{ij} та вартістю передачі одиниці потоку c_{ij}. У результаті вартість можна обчислити за такою формулою:

$$c_{ij} = \alpha \cdot d_{ij} + \beta \cdot p_{ij}, \quad \forall (i, j) \in E, \tag{2.10}$$

де α та β – вагові коефіцієнти для затримки та втрат відповідно.

Ця формула дає змогу скоригувати вартість шляху з урахуванням того, наскільки затримка чи втрати пакетів важливі для цього потоку. Таким чином можна змінювати ці параметри відповідно до вимог щодо якості обслуговування кожного потоку. Наприклад, трафік мультимедіа має строго визначені обмеження щодо затримки з кінця в кінець, тому можна встановити $\alpha = 1$ та $\beta = 0$, щоб врахувати тільки затримку. Кожен окремий потік ідентифікується за допомогою відносного пріоритету k, значення якого знаходиться в межах від нуля до одиниці. Набір потоків, які необхідно передати по мережі, позначений K. Кожен потік характеризується п’ятьма параметрами:

- вузол, де потік k входить в мережу s_k;
- вузол, де потік k виходить з мережі t_k;
- f_k швидкісь потоку k, який необхідно доставити з вхідного до вихідного вузла;

- $P_{\max}^{k} \geq 0$ – максимально допустиме значення втрат пакетів для *k*-го потоку;
- $D_{\max}^{k} \geq 0$ – максимально допустиме значення затримки для *k*-го потоку.

Метою оптимізації є маршрутизація всіх потоків у мережі вздовж найкоротшого шляху з мінімальною вартістю.

Набори:

- вузли: $n \in V$;
- ребра: $(i, j) \in E$;
- канали: $(i, j) \in E \cup (j, i) \in E$;
- Змінні:
- $0 \leq x_{ij}^{k} \leq f_k$ – обсяг *k*-го потоку, що проходить по каналу (i, j).
- Параметри:
- $b_{ij} \geq 0$ – доступна пропускна здатність у каналі (i, j);
- $0 \leq \eta \leq 1$ – максимальне завантаження каналу по всій мережі;
- $0 \leq r \leq 1$ – відносний пріоритет вартості шляху над максимальним завантаженням каналу η;
- $C_{\max} \geq 0$ – максимальна пропускна здатність каналу;
- $c_{ij} \geq 0$ – вартість каналу (i, j), що розраховується як $\alpha \cdot d_{ij} + \beta \cdot p_{ij}$;
- α – ваговий коефіцієнт для затримки;
- β – ваговий коефіцієнт для втрат;
- $s_k \in V$ – вхідний вузол *k*-го потоку;
- $t_k \in V$ – вихідний вузол *k*-го потоку;
- f_k – швидкість *k*-го потоку;
- $P_{\max}^{k} \geq 0$ – максимально допустиме значення втрат пакетів *k*-го потоку;
- $p_{ij} \geq 0$ – значення втрат пакетів у каналі (i, j);
- $D_{\max}^{k} \geq 0$ – максимально допустиме значення затримки *k*-го потоку;

- $d_{ij} \geq 0$ – затримка в каналі (i, j);
- $B^k \geq 0$ – пропускна здатність, необхідна для k-го потоку.

Маніпуляція параметрами маршрутизації відбувається на основі відносного пріоритету потоку, що обчислюється згідно методики, представленої в другому розділі роботи. Всі значення відносного пріоритету потоку TC лежать у межах від 0 до 1. Введемо параметри $TC_{High\,Priority} \in TC$ та $TC_{BestEffort} \in TC$, при чому завжди $TC_{High\,Priority} < TC_{BestEffort}$. Параметр $TC_{High\,Priority}$ містить значення відносного пріоритету, стосовно якого порівнюються відносні пріоритети всіх інших потоків. Якщо відносний пріоритет потоку більший, ніж $TC_{High\,Priority}$, тоді потік належить до першої категорії і має високий пріоритет. Це означає, що його не можна розбивати на підпотоки для балансування навантаження. Якщо відносний пріоритет потоку менший, ніж $TC_{High\,Priority}$, проте більший від $TC_{BestEffort}$, тоді потік належить до другої категорії і має середній пріоритет. Це означає, що потік можна розбивати на підпотоки та вибирати для нього шляхи з якістю обслуговування, не нижчою, ніж може забезпечити наявний доступний для передачі найкоротший шлях. Такі потоки мають обмеження, яке визначає мінімально можливий розмір підпотоку. Це дає змогу керувати рівнями перерозподілу потоку в мережі з метою уникнення сильного перемішування пакетів для потоків типу TCP. Всім потокам, пріоритет яких нижчий, ніж $TC_{BestEffort}$, не гарантується якість обслуговування. Вони можуть бути розбиті на підпотоки довільно малих розмірів. Для таких потоків обираються шляхи з мінімальним завантаженням каналів.

Цільова функція (2.11) мінімізує вартість залежно від параметрів якості обслуговування в каналах, а також від коефіцієнта мінімального/максимального завантаження каналів, який залежить від типу відносного пріоритету потоку:

$$F(x) = \eta + r \sum_{(i,j)\in E} \sum_{k\in K} c_{ij} x_{ij}^k \to \min. \qquad (2.11)$$

Обмеження (2.12) відповідає умові збереження потоку:

$$\sum_{(i,j)\in E} x_{ij}^k - \sum_{(i,j)\in E} x_{ji}^k = \begin{cases} f_k, & i = s_k, \\ -f_k, & i = t_k, \\ 0, & i \neq s_k, t_k \end{cases} \quad \forall i \in V, \forall k \in K \tag{2.12}$$

Обмеження (2.13) відповідає за допустимий рівень розподілу потоку залежно від його відносного пріоритету:

$$x_{ij}^k \geq \begin{cases} B_H, & k > TC_{High\,Priority}, \forall (i,j) \in E, \\ B_L, & TC_{BestEffort} < k < TC_{High\,Priority}, \forall (i,j) \in E, \\ 0, & k < TC_{BestEffort}, \forall (i,j) \in E, \end{cases} \tag{2.13}$$

де B_L та B_H - мінімальна пропускна здатність підпотоків при балансування основного потоку другого та першого класу відповідно.

Наступні два обмеження (2.14 та 2.15) відображають максимально допустимі значення втрат пакетів та затримки, які не повинні перевищувати критичних значень $D_{\max}^k$, $P_{\max}^k$ для потоку з відносним пріоритетом *k*. У випадку маршрутизації третьої категорії цим параметрам надається максимально можливе значення:

$$\sum_{(i,j)\in E} p_{ij} \leq \begin{cases} P_{\max}^k, & x_{ij}^k > 0, \forall k \in K; \\ 0, & x_{ij}^k = 0, \forall k \in K; \end{cases} \tag{2.14}$$

$$\sum_{(i,j)\in E} d_{ij} \leq \begin{cases} D_{\max}^k, & x_{ij}^k > 0, \forall k \in K; \\ 0, & x_{ij}^k = 0, \forall k \in K; \end{cases} \tag{2.15}$$

Нерівність (2.16) задає обмеження на доступну пропускну здатність кожного каналу з врахуванням всіх потоків k, що проходять цими каналами. При чому, це обмеження є нижчим у випадку маршрутизації потоків другої та третьої категорії, що спричиняє їх передачу шляхами з низькою ефективністю завантаження та з погіршенням якості обслуговування:

$$\sum_{k\in K} x_{ij}^k \leq \begin{cases} \eta \cdot C_{\max}, & \forall (i,j) \in E, \quad k < TC_{High\,Priority} \\ b_{ij}, & \forall (i,j) \in E \end{cases} \tag{2.16}$$

Остання умова (2.17) визначає діапазон значень змінної та гарантує, що змінна набуває значення в межах швидкості потоку:

$$0 \leq x_{ij}^k \leq f_k \quad \forall (i,j) \in E, \quad \forall k \in K \tag{2.17}$$

Запропонована математична модель дає змогу встановити максимально допустимі значення втрат та затримки для потоків першої категорії. Ці значення є основними для вибору оптимального шляху для потоку за критерієм якості обслуговування. Завдяки використанню вагових коефіцієнтів можна підібрати вартість для кожного потоку залежно від його вимог. З одного боку, можемо встановити максимально допустимі значення втрат та затримки і в результаті розв'язку задачі лінійного програмування отримати шлях з мінімальною вартістю передачі, який задовольняє вимоги до параметрів якості обслуговування. З іншого боку, можемо змінювати максимально допустимі значення втрат і затримки та проводити перерахунок оптимальних шляхів з метою знаходження шляху, який буде гарантувати необхідну якість.

2.6. Модель балансування навантаження на основі критерію максимально допустимого завантаження каналу

Зазвичай використання фізичних каналів протягом дня становить від 30% до 40% від загальної пропускної здатності. Проте існують періоди, коли навантаження зростає і починає стрибкоподібно коливатися в значних межах, що дуже часто призводить до втрати пакетів та погіршення якості обслуговування користувачів.

Оптимальний розподіл трафіку та завантаження каналів здійснюється на основі методів динамічного балансування навантаження [61-66], які застосовуються під безпосереднім керівництвом контролера. Це означає, що контролер постійно здійснює моніторинг стану каналів та комутаторів та у випадку необхідності відсилає інструкції на комутатор щодо зміни таблиць та процесів обробки потоків.

Ідея методу полягає в розподіленні навантаження одного потоку між декількома незавантаженими каналами, які є елементами шляху, що веде до вузла призначення. Оскільки в груповій таблиці існує множина записів, що відображають різні маршрути передачі даних одного потоку, то можна легко здійснювати управління навантаженням, беручи за основу вагові коефіцієнти

кожної гілки. Ваговим коефіцієнтом гілки є числове значення пріоритету відповідної гілки, що вказує на відсоткове співвідношення кількості пакетів потоку, які необхідно передати по даній гілці. Проте такий розподіл не є ефективним, оскільки, у разі високого завантаження всіх каналів у мережі, статичний розподіл потоку не зможе уникнути перевантаження внаслідок непередбачуваних стрибків. Тому в роботі запропоновано вибрати як критерій балансування пакетів потоку в площині передачі даних не відносний пріоритет шляху, а максимально допустиме завантаження наступного каналу, через який проходить шлях. Відповідно до такого критерію, комутатор направляє всі пакети потоку в один канал, якщо завантаження каналу менше, ніж максимально допустиме завантаження. Наприклад, якщо в певний момент часу, сумарна кількість байт, переданих в канал за одну секунду, перевищує 90% пропускної здатності, тоді для передачі наступного пакету вибирається наступний можливий шлях. Перебір шляхів відбувається до тих пір, поки не буде знайдено шляху, доступного для передачі цього пакету. Якщо шлях не знайдено, тобто всі канали перевантажені, тоді пакет відкидається.

Контролер постійно здійснює моніторинг завантаження інтерфейсів і, коли він виявляє, що ймовірність блокування певного інтерфейсу наближається до одиниці, тоді розраховує набір можливих шляхів для перерозподілу потоків третьої категорії (табл..2.11). Тут же ж він визначає кількість необхідних гілок.

Групова таблиця не підтримує динамічну зміну пріоритетів гілок. Більш того, така зміна не може здійснюватися в дрібному масштабі часу (менше однієї секунди). Навіть, якщо така зміна була б можливою, це б сильно впливало на продуктивність таблиці внаслідок постійної її модифікації та завантаження центрального процесора. Тому в наступному розділі роботи дослідження цього методу проводиться на основі імітаційного моделювання. Відповідно до запропонованого методу, групова таблиця повинна оперувати обсягом переданих кожним шляхом байт та перенаправляти пакети на основі описаного вище механізму.

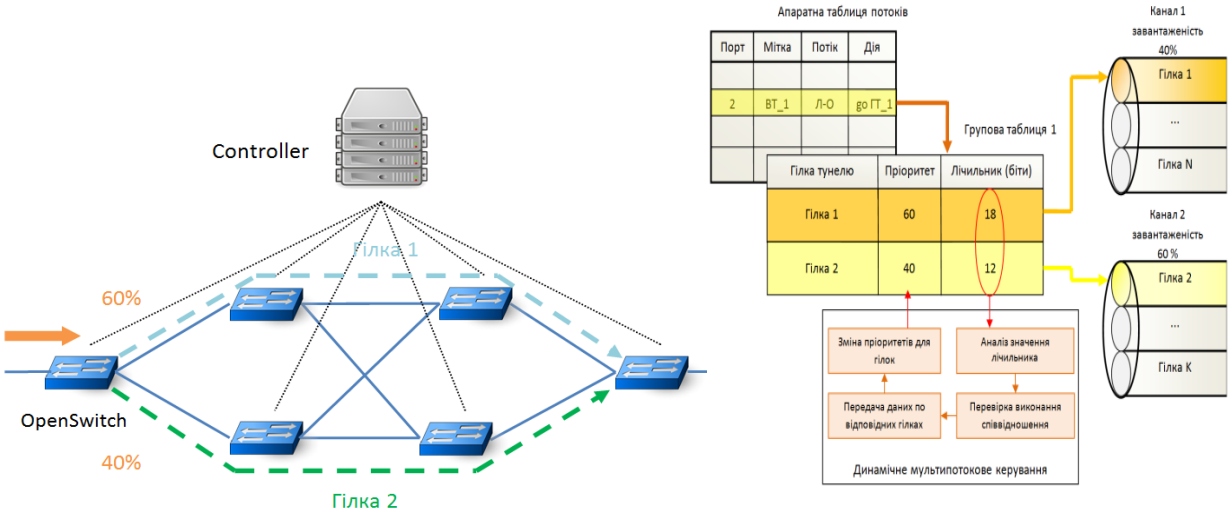


Рис. 2.11. Принцип балансування трафіку між гілками одного віртуального тунелю на основі зміни пріоритету гілок

Розподіл смуги пропускання мережевих каналів між існуючими потоками з урахуванням прогнозів зміни швидкості кожного потоку дозволяє ефективно використовувати ресурси мережі, розподіляючи їх між різними шляхами, що суттєво збільшує адаптивність мережі стосовно типу трафіку. Динамічна зміна смуги пропускання дасть можливість завжди забезпечувати ефективність використання ресурсів каналу на високому рівні без втручання контролера, оскільки ресурси оптимально перерозподіляються або надаються для можливого використання іншими тунелями. Цей метод особливо важливий у разі виникнення ситуації перевантаження всіх каналів, оскільки дає змогу комутатору локально обрати вільний шлях для передачі пакету, що знижує ймовірність втрати пакетів.

Висновки до 2-го розділу

Централізована архітектура програмно-керованих мереж дає змогу здійснювати гнучке керування процесами передачі даних для забезпечення необхідного рівня окремих характеристик мережі. Серед них можна виділити, зокрема, якість обслуговування мультисервісних потоків та ефективність використання мережевих ресурсів.

Запропоновано спосіб ідентифікації потоку конкретного окремого клієнта на основі відносного пріоритету, розрахованого з допомогою удосконаленого методу призначення пріоритетів трафіку. Розроблено систему моніторингу параметрів функціонування програмно-керованих мереж, яка використовує різні типи каналів для доступу до параметрів програмної та апаратної частини комутатора, зокрема підтримує декілька інтерфейсів для управління функціями OpenFlow, у тому числі й безпосередню комунікацію з комутатором без участі контролера. Система надає готові структури для зберігання та обробки даних моніторингу, які можуть бути розширені для індивідуальних потреб адміністратора мережі. Розроблена система дає змогу обійти складність доступу до параметрів апаратної частини комутатора за рахунок використання віддаленого протоколу доступу до терміналу та прямої маніпуляції комутатором на основі бази даних інструкцій встановленої операційної системи. За рахунок цього система суттєво спрощує отримання комплексної інформації про мережеві процеси, і може бути розширена новими методами моніторингу. У роботі розроблено метод моніторингу з адаптивними параметрами опитування, який підлаштовує параметри системи моніторингу відповідно до характеру/інтенсивності мережевих процесів, наприклад, завантаження каналу чи вузла. Розроблено метод вимірювання затримки для окремого потоку окремого класу, який дає змогу оцінити якість обслуговування потоків окремих клієнтів та відповідно отримати необхідну інформацію для оптимізації мережі та процесів передачі даних. Удосконалено потокову модель маршрутизації в програмно-керованих мережах для підвищення ефективності використання мережевих ресурсів та забезпечення необхідної якості обслуговування всім складовим мультисервісного потоку. Удосконалено модель балансування навантаження в програмно-керованих мережах для керування параметрами потоку та передачі даних в умовах перевантажених каналів.

РОЗДІЛ 3. ДОСЛІДЖЕННЯ ПРОЦЕСІВ ОБСЛУГОВУВАННЯ НАВАНТАЖЕННЯ ТА МОНІТОРИНГУ МЕРЕЖЕВИХ РЕСУРСІВ

У розділі розроблено систему для генерації мультисервісних потоків, в основі якої лежить комунікаційна платформа, використана для розробки системи моніторингу. Проведено дослідження параметрів функціонування апаратного OpenFlow комутатора *HP3500yl*. З урахуванням отриманих характеристик розроблено імітаційну модель OpenFlow комутатора, яка може використовуватися в системах моделювання на основі дискретних подій. На основі апаратного комутатора досліджено ефективність методу адаптації системи моніторингу та на реальному прикладі показано виграш, що може бути отриманий унаслідок його використання. Досліджено ефективність удосконаленої моделі балансування навантаження. Наведені у розділі результати опубліковано в працях [76; 80; 80; 88; 89; 90; 94; 95-99; 101; 102].

3.1. Розробка системи для генерації мультисервісних потоків

В імітаційних моделях для дослідження мультисервісних мереж переважно використовують генератори трафіку на основі статистичних законів розподілу випадкової величини. Найпростіший генератор трафіку оперує двома параметрами: розміром пакету та тривалістю інтервалу між двома пакетами. Для моделювання мультисервісного трафіку використовується третій параметр, значення якого генерується випадковим чином, і визначає тип трафіку з набору доступних класів. Зміна інтенсивності трафіку досягається зміною інтервалу між пакетами. Переважно для таких генераторів використовують закони розподілу, функція густини розподілу яких стрімко зростає чи спадає на незначному інтервалі значень параметру (експоненціальний, гіперекспоненціальний, лонгнормальний та інші). Потік, згенерований таким методом, зазвичай має чітко виражену характеристику самоподібності, яка притаманна мультисервісному трафіку.

У тому випадку, коли дослідження проводиться не на моделі, а на реальній мережі, необхідно використовувати програмне забезпечення, яке буде виступати в ролі генератора та приймача трафіку. Таке програмне забезпечення може імітувати поведінку як окремого користувача, так і групи користувачів. В останньому випадку зі збільшенням кількості користувачів погіршуються характеристики трафіку, оскільки програма опрацьовує тільки одного користувача в один момент часу, а стек протоколів операційної системи вносить свою затримку в передачу пакетів. Ці фактори впливатимуть на часові параметри передачі пакетів, що погіршуватиме якість та точність отриманих результатів дослідження параметрів якості обслуговування мультисервісних потоків загалом. Більш того, такий метод не забезпечує генерацію навантаження з різних IP адрес, а лише дає можливість моделювати навантаження від багатьох клієнтів як віртуальний пакет у корисному навантаженні реального пакету. Програма, яка моделює роботу мережевого вузла чи кінцевого пристрою користувача, отримує корисне навантаження з реального пакета та аналізує віртуальний пакет. Таким чином моделюється віртуальна IP мережа поверх реальної IP мережі. При цьому такі програми мають змогу змінювати деякі параметри заголовку IP пакету, наприклад, поле DSCP, та встановлювати пріоритет пакету. Такий метод можна використовувати для дослідження традиційних IP мереж, оскільки в них важливою є адреса вузла призначення та тип сервісу при маршрутизації пакетів. Саме ця інформація використовується протоколами маршрутизації при прокладанні маршрутів та більшістю методів оптимізації процесу передачі даних і розподілу навантаження в мережі. Основна перевага такого методу полягає в можливості генерації великих обсягів трафіку різних класів (максимальна швидкість потоку залежить від максимальної швидкості мережевої карти сервера, на якому встановлено генератор).

Для забезпечення адекватних характеристик функціонування мережі слід використовувати більшу кількість кінцевих фізичних пристроїв та якомога більше диференціювати їх за типами послуг. У таких умовах трафік може

генеруватися реальними додатками, що зазвичай використовують користувачі (браузер, мережевий програвач, засоби віддаленого перегляду медіа контенту та спілкування між користувачами в режимі реального часу). Замість реальних клієнтів дуже часто можуть використовуватися програми типу *Iperf*, які дають змогу тестувати пропускну здатність мережі. При цьому користувач має змогу змінювати тип транспортного протоколу, тип трафіку та генерувати трафік від декількох клієнтів. Також цей додаток дає змогу відтворити такий самий трафік на основі зібраних даних з реальної мережі, генеруючи відповідні пакети в певні моменти часу. Такі дані можуть бути збережені у файлі, де пакети та інформація про них відсортована в часі відповідно до моменту надходження на маршрутизатор. Цей спосіб дає змогу адекватно відтворити мультисервісний трафік, що передається в реальній мережі. Проте його недолік полягає у складності налаштування такого середовища та великій кількості необхідних пристроїв, а також у керуванні такою системою з метою створення умов, необхідних для дослідження.

У роботі розроблено систему для генерації мультисервісного трафіку, яка поєднує в собі переваги всіх вище перерахованих методів і може використовуватися для генерації мультисервісного навантаження залежно від обраного способу та сценарію дослідження.

Структурну схему моделі такої системи представлено на рис.3.1.

В основі цієї моделі лежить елемент «користувач», який здійснює генерацію пакетів. Контейнером для набору «користувачів» є «локальна мережа», яка містить елемент, що відповідає за розподіл мережевих адрес у разі використання системи в імітаційних моделях (при дослідженні в реальних системах елемент «користувач» отримує IP адресу, що належить операційній системі, яка виступає в ролі клієнтського терміналу). Кожна локальна мережа містить структуру даних, що описує всі компоненти «локальної мережі». Ця структура використовується для автоматичного налаштування та конфігурації системи генерації трафіку. У ній містяться такі дані: кількість користувачів, їхні адреси, класи трафіку та випадкові параметри, необхідні для генерації

відповідного типу трафіку для кожного користувача, адреси партнерів по комунікації для кожного користувача. У разі запуску системи локальна мережа використовує файл конфігурації для створення «користувачів» і налаштування їхніх параметрів. Для кожного «користувача» «локальна мережа» створює елемент, що відповідає за моніторинг параметрів користувача та їхню статистичну обробку (кількість переданих біт/пакетів, середня тривалість очікування відповіді від партнера по комунікації, міжпакетний джиттер у випадку потоків реального часу). «Монітор» користувача дає змогу впровадити математичний апарат для оцінки як параметрів якості обслуговування, так і параметрів якості сприйняття саме з точки зору «користувача». Одному «користувачу» відповідає один «монітор». Це дає змогу проводити середньостатистичну оцінку якості обслуговування не просто для агрегованого потоку, а для окремого потоку окремого клієнта.

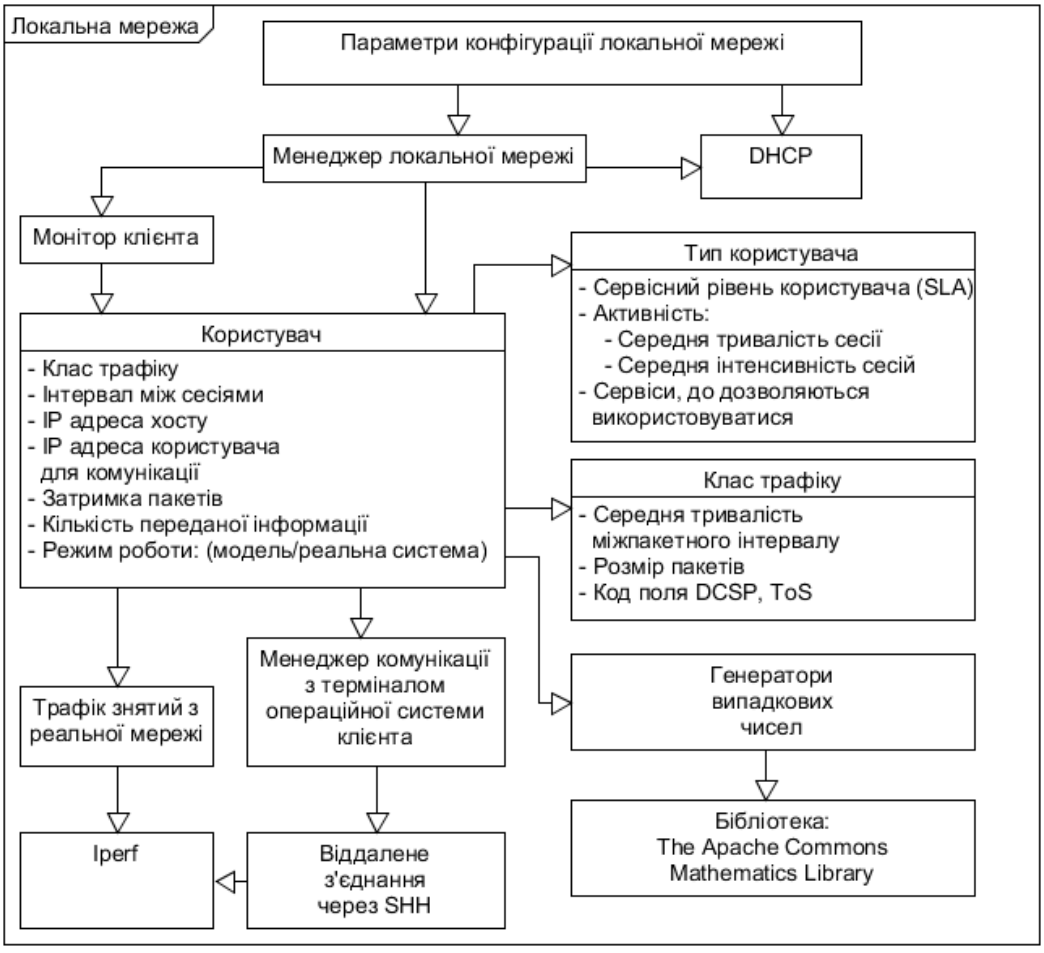


Рис. 3.1. Структурна схема моделі системи генерації мультисервісного трафіку

Кожен «користувач» характеризується такими параметрами: тип користувача, клас трафіку, інтенсивність відкриття комунікаційної сесії, власна мережева адреса, набір адрес партнерів по комунікації, режим роботи, поточна кількість переданої інформації в бітах і пакетах, а також масив затримок параметрів якості обслуговування, який постійно оновлюється з кожним новим отриманим пакетом.

Параметр «тип користувача» містить посилання на структуру даних, яка описує активність користувача та його вимоги щодо QoS за допомогою таких параметрів: середня тривалість сесії, середня інтенсивність сесій, а також сервісний рівень користувача SLA. Опираючись на останній параметр, «монітор» кожного «користувача» визначає відповідність наданого рівня сервісу рівню, прописаному в SLA. У цій структурі також містяться посилання на сервіси, які може отримувати «користувач», та їхні мережеві адреси.

Поле «клас трафіку» містить параметри, що характеризують потік пакетів, які генеруються «користувачем» упродовж однієї сесії одного типу. До них належать: середня тривалість міжпакетного інтервалу, середній розмір пакету, коди полів *ToS* та *DSCP*. Ця структура даних може бути розширеною для врахування додаткових полів заголовків протоколів. На основі перерахованих параметрів «тип користувача» і «клас трафіку» «користувач» здійснює генерацію пакетів, використовуючи генератори випадкових чисел. Останні запрограмовані та зібрані в одній бібліотеці, яка називається *The Apache Commons Mathematics Library* [67].

Параметр «режим роботи» визначає режим, в якому працює користувач. Розроблена система підтримує два режими: модель та реальна система. У режимі моделі користувач підключається до віртуального маршрутизатора, який є компонентом локальної мережі та виступає в ролі інтерфейсу між користувачами й іншими (OmNET++ [68], OpNet [69], Mininet [70]) імітаційними моделями телекомунікаційних систем. У режимі реальної мережі користувач під’єднується до віддаленого комп’ютера, використовуючи менеджера комунікації з терміналом операційної системи. Такий менеджер

містить наперед збережений протокол та алгоритм комунікації з відповідною операційною системою. Застосовуючи уніфікований інтерфейс менеджера, користувач може виконувати конкретні дії на віддаленому терміналі з допомогою одних і тих самих методів. Робота з віддаленим терміналом реалізується за допомогою компоненту *SSH*, який містить методи для роботи з протоколом *SSH*. Менеджер комунікації з віддаленим терміналом та компонент *SSH* разом утворюють канал для роботи з віддаленими пристроями. За допомогою цього каналу система генерації мультисервісного трафіку керує додатком *Iperf* (чи будь яким іншим генератором трафіку, що підтримує керування через термінал). Необхідний трафік може створюватися як на основі заданих у системі характеристик, так і зчитуватися з попередньо записаного файлу.

Запропонована базова модель системи генерації трафіку може бути використана для створення тестових середовищ з метою дослідження реальної мережі з використанням декількох окремих фізичних комп'ютерів для імітації клієнтів. При дослідженні характеристик передачі даних у реальній мережі основний та сигналізаційний трафік, що необхідний для роботи тестового середовища, повинні бути відокремлені. У найкращому випадку сигналізаційний трафік повинен передаватися через окрему фізичну мережу. В іншому випадку необхідно створити додаткову віртуальну локальну мережу в тестовій мережі. Така схема дасть змогу проводити експерименти віддалено, використовуючи єдиний пристрій для управління усіма іншими пристроями, що суттєво розширює можливості дослідника, спрощує проведення нових експериментів та економить час на їх постановку. У процесі дослідження для тестування ефективності функціонування системи проведено експеримент, схему якого відображено на рис.3.2.

Для генерації використано 5 фізичних серверів. На п'ятому сервері було встановлено розроблену систему моніторингу, яка керувала всіма іншими комп'ютерами. Всі комп'ютери під'єднані до OpenFlow комутатора *HP3500yl*.

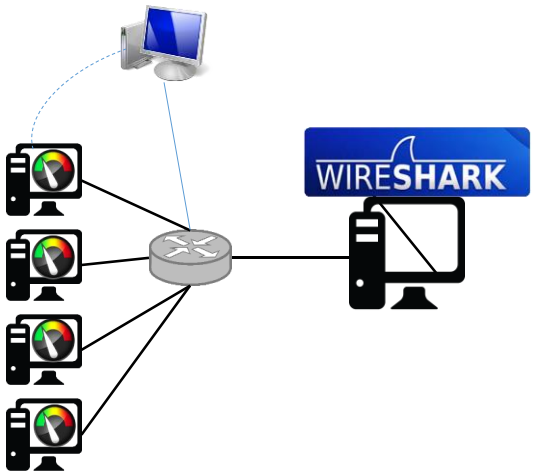


Рис. 3.2. Схема тестового середовища для дослідження характеристик мультисервісного трафіку та оцінки якості розробленої системи генерації мультисервісного трафіку

Для моніторингу створеного потоку трафіку всі потоки перенаправлялися в один вихідний порт комутатора. До цього порту був під'єднаний шостий сервер, на якому встановлений додаток *WireShark*. Саме з допомогою *WireShark* отримано графіки, відображені на рис.3.3. В описаному експерименті було створено 7 класів трафіку, визначених у другому розділі. На рис.3.3 представлено мультисервісний агрегований потік, що генерується за допомогою розробленої системи.

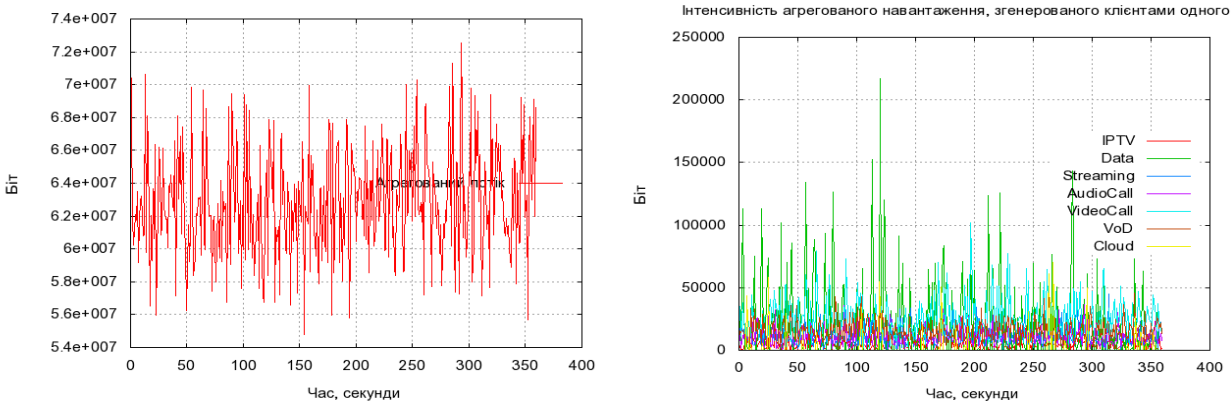


Рис. 3.3. Мультисервісний агрегований трафік, що генерується десятьма клієнтами з різних фізичних машин за допомогою розробленої системи

На графіку справа (див. рис.3.3) відображено агрегований мультисервісний потік, утворений меншими потоками від чотирьох користувачів, а зліва – інтенсивності потоків для семи класів послуг.

3.2. Розроблення імітаційної моделі апаратного програмно-конфігурованого комутатора

У роботі проведено аналіз факторів, що впливають на характеристики функціонування та продуктивність програмно-керованої мережі, яка складається з апаратних комутаторів *HP3500yl* та контролера *HP VAN SDN 2.5*. У процесі дослідження проводиться маніпуляція параметрами комутатора та встановлюються правила в різні таблиці потоків і ведеться спостереження щодо того, яким чином комутатори змінюють свою поведінку та наскільки швидко можуть виконувати ті чи інші інструкції. Мета дослідження – визначити основні параметри, що впливають на продуктивність комутатора, а саме дослідити:

- функціонування комутатора, коли таблиця потоків переповнена;
- зміну затримки комутації у разі наближення завантаження мережевого інтерфейсу до 100%;
- зміну затримки комутації, коли завантаження цілого комутатора наближається до 100%;
- зміну продуктивності комутації в процесі встановлення нових правил;
- залежність продуктивності комутації від інтенсивності моніторингу параметрів комутатора;
- залежність тривалості встановлення правил та отримання статистичної інформації від поточного завантаження інтерфейсу та цілого пристрою;
- залежність завантаження центрального процесора від інтенсивності моніторингу та встановлення правил;
- залежність тривалості встановлення правил та тривалості отримання статистики від розміру таблиць потоків.

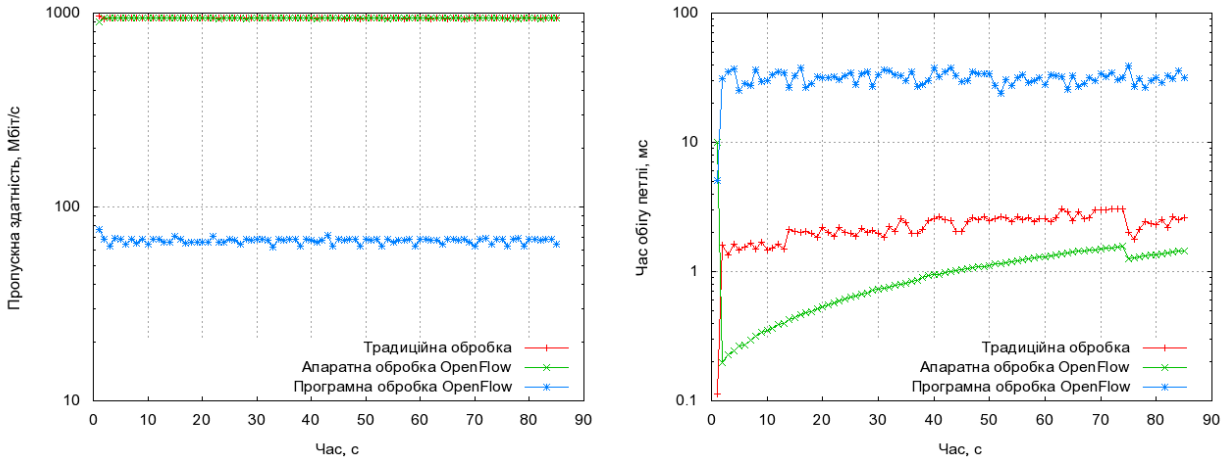
У всіх експериментах використовується проактивний метод конфігурації комутатора, коли всі необхідні правила встановлюються ще до того, як перший пакет надійде на комутатор.

Мотивація. Основною причиною проведення цих досліджень є те, що продуктивність та поведінка апаратних програмно-керованих комутаторів може суттєво відрізнятися поміж пристроїв різних виробників, а то й поміж моделей пристроїв одного виробника. Основною задачею є дослідити, як реалізація функціональності OpenFlow впливає на продуктивність комутатора, а відповідно – на якість обслуговування мультисервісних потоків.

На основі отриманих результатів у роботі розроблено модель програмно-керованого комутатора, яка дає змогу відтворити основні характеристики продуктивності програмно-керованих комутаторів. Необхідність створення такої моделі пояснюється відсутністю на сьогодні моделей, які можуть бути використанні для швидкого аналізу продуктивності певної мережевої конфігурації. Зокрема, основна частина дослідників використовує емулятор *Mininet* [70], який базується на реальному програмному комутаторі *OpenVSwitch* [71]. Моделювання на основі такого комутатора є не завжди виправданим, оскільки він є надзвичайно деталізованим, що ускладнює впровадження та тестування нової функціональності. Варто зауважити, що OpenVSwitch в основному використовуються дослідниками для тестування програмних додатків для ПКМ, продуктивності контролерів, оптимізації передачі даних та балансування навантаження з урахуванням QoS. Проте дуже часто ці дослідження не враховують особливості реалізації та функціонування апаратної частини комутаторів, абстрагуючись від них. У результаті значна частина рішень, які в теорії дають покращення та виграш в якості й ефективності, на практиці показують зовсім протилежні результати. Крім того, ці рішення досить часто проблематично реалізувати на практиці через складність налаштування та підтримки складного комплексного програмного забезпечення.

У роботі проведено дослідження параметрів функціонування апаратного програмно-керованого комутатора *HP3500yl*. Цей комутатор є гібридним, тобто підтримує як традиційну обробку пакетів, так і обробку пакетів відповідно до специфікації OpenFlow.

Основною характеристикою комутатора є його пропускна здатність. Для гібридних комутаторів пропускна здатність за методами традиційної обробки та обробки відповідно до специфікації OpenFlow може відрізнятися. Для цього у роботі проведено тестування пропускної здатності комутатора *HP3500yl* у трьох режимах: стандартному, апаратному OpenFlow та програмному OpenFlow. Результати тестування відображено на рис.3.4.



(а) (б)

Рис. 3.4. Порівняння пропускної здатності (а) та тривалості обігу петлі (б) програмно-керованого комутатора в трьох режимах роботи

Графіки пропускної здатності та тривалості обігу петлі на рис.3.4 демонструють пропускну здатність (а) та тривалість обігу петлі (б) для трьох режимів обробки пакетів. З отриманих результатів можна зробити висновок, що програмна комутація характеризується продуктивністю, що майже в 100 разів менша за апаратну комутацію. Крім того, обидва режими апаратної обробки показують однакові результати продуктивності. У всіх трьох режимах тривалість обігу петлі зростає з часом при максимальному завантаженні інтерфейсу.

Дослідження залежності продуктивності програмно-керованого комутатора від розміру пакету.

Ще один експеримент проведено для оцінки пропускної здатності комутатора в трьох режимах залежно від розміру пакету. Розмір змінювався від

60 до 1520 байт з кроком, рівним 20 байт. Результати тестування відображено на рис.3.5.

На основі результатів експерименту можна стверджувати, що апаратна комутація в обох режимах характеризується однаковою продуктивністю, в той час, як програмна комутація знову показує в рази меншу продуктивність. При чому, продуктивність у випадку програмної комутації зростає лінійно та в незначному діапазоні, тоді як апаратна комутація характеризується продуктивністю, що змінюється нелінійно.

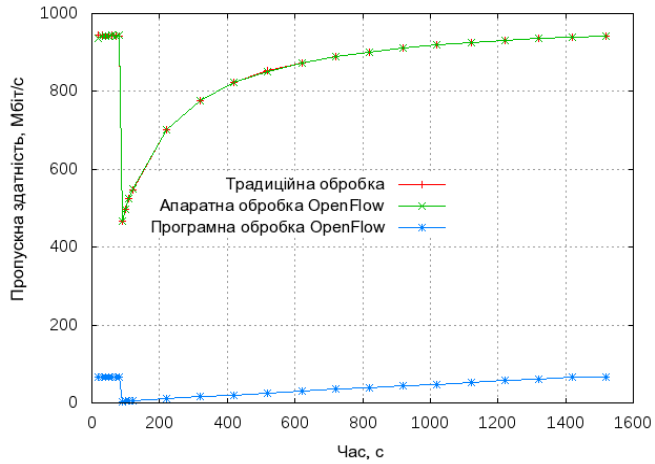


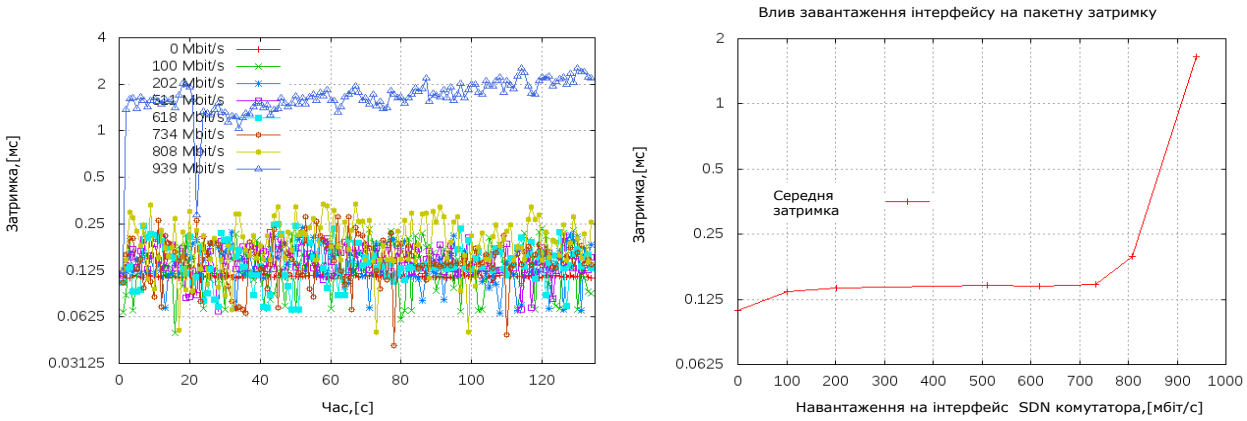
Рис. 3.5. Дослідження залежності продуктивності програмно-керованого комутатора від розміру пакетів для трьох режимів роботи.

Високі значення пропускної здатності на початку кривих апаратної комутації є результатом того, що мінімальний розмір пакету, який можна встановити за допомогою утиліти *Iperf*, може бути не меншим, ніж 60 байт. Якщо вручну встановити значення розміру пакету менше від 60 байт, тоді *Iperf* встановлює розмір пакету за замовчуванням (1520 байт).

Дослідження затримки оброблення пакету залежно від завантаження інтерфейсу програмно-керованого комутатора.

Дослідження затримки пакету проводилося за таким самим методом, як у попередніх тестах, за винятком того, що використовувалася утиліта *Ping*. Обидві програми *Iperf* та *Ping* функціонують одночасно. Тестування проводилося в декілька ітерацій. У кожній ітерації швидкість потоку збільшувалася на 100, починаючи зі 100 Мбіт/с. Кожна ітерація тривала 140

секунд. Для комутації обох потоків використовувалася одна й та ж сама пара правил, що й у всіх попередніх тестах. Результати тестування відображено на рис.3.6.



(а) (б)

Рис. 3.6. Характеристика затримки пакетів залежно від завантаження інтерфейсу для апаратної комутації в режимі OpenFlow

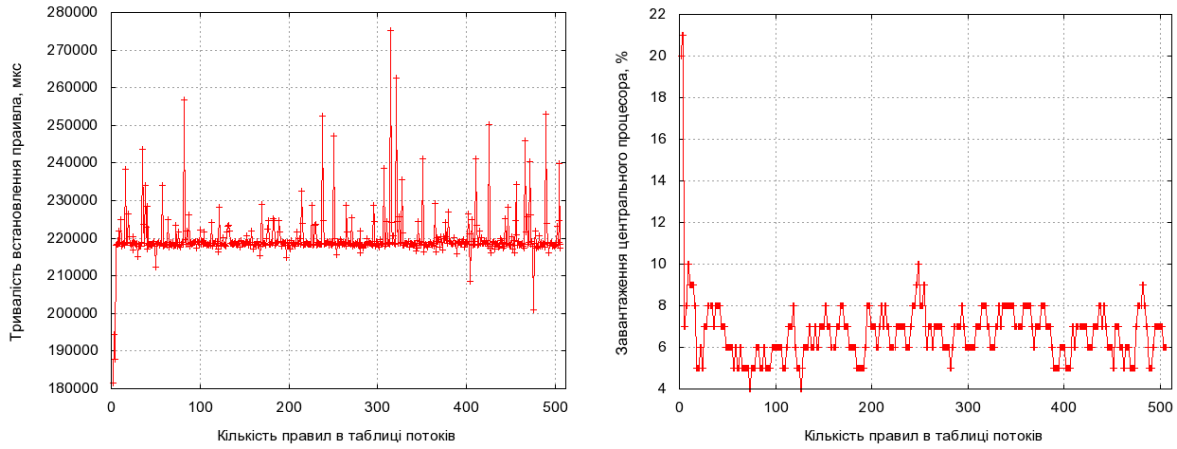
Миттєві значення затримок при різних швидкостях потоків відображено на рис.3.6а, а характеристику зміни затримки – на рис.3.6б. Результати тестування показують, що затримка для пакетів залишається сталою майже при будь-якій пропускній здатності, нижчій від 800Мбіт/с. У тому разі, коли швидкість потоку перевищує 800 Мбіт/с, затримка пакетів починає стрімко та майже лінійно зростати. Уже при 940Мбіт/с затримка досягає значення 2 мс, що майже в 200 разів більше, ніж значення затримки, яке спостергіається при швидкості потоку до 800 Мбіт/с.

Дослідження залежності між пропускною здатністю та встановленням нових правил в таблицю потоків програмно-керованого комутатора.

Перед проведенням експерименту було здійснено оцінку затримки передачі сигнального повідомлення каналом OpenFlow між контролером і комутатором. Для експерименту створено повідомлення типу *flow_mod*, яке використовується контролером для модифікації/встановлення правил у таблиці потоків. У відповідь на кожне повідомлення щодо модифікації таблиці потоків комутатор відправляв повідомлення відмови з кодом помилки BAD_MATCH_FIELDS. Таких повідомлень було відправлено 1000 з інтервалом

5 секунд між двома послідовними повідомленнями. У результаті встановлено, що середній час обходу петлі (контролер – комутатор – контролер) становить 15 мс.

Після цього проведено тестування тривалості встановлення правил залежно від розміру таблиці потоків. Для тесту сформовано набір правил (всього 600), поля яких містять значення, що ніколи на співпадуть із жодним пакетом. Після цього запущено генератор трафіку, який створив максимально можливе навантаження на інтерфейс комутатора (940Мбіт/с). Навантаження генерувалося безперервно впродовж усього експерименту. Паралельно контролер встановлював правила одне за одним з мінімальним інтервалом. Контролер розпочинав встановлення наступного правила в момент отримання відповіді-підтвердження про встановлення попереднього правила від комутатора. Тестування закінчилося в той момент, коли таблиця потоків була переповнена. Система моніторингу опитувала значення завантаження з інтервалом, рівним одній секунді. У результаті отримано характеристики трьох параметрів: пропускна здатність, швидкість встановлення правил, завантаження центрального процесора. Упродовж всього експерименту пропускна здатність залишалася незмінною. Характеристику завантаження центрального процесора та тривалості встановлення потоків відображено на рис.3.7а та .3.7б.



(а) (б)

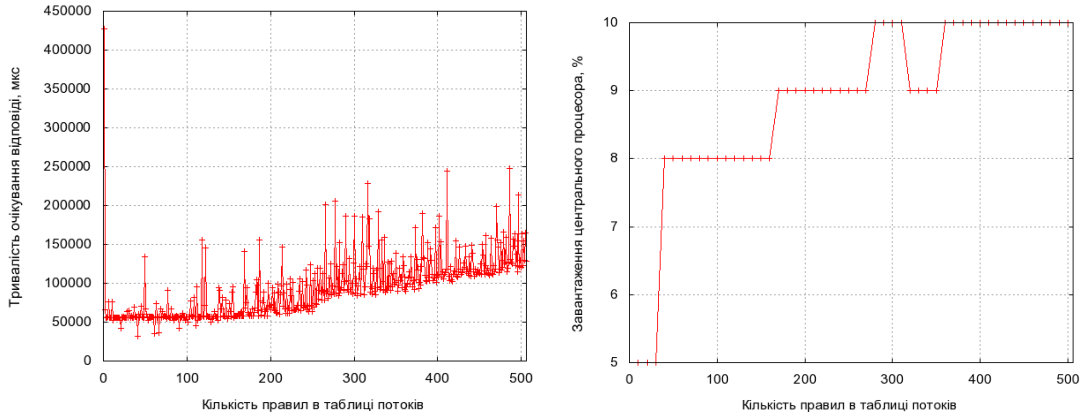
Рис. 3.7. Характеристика завантаження центрального процесора (а) та тривалості встановлення нових правил (б) залежно від кількості правил у таблиці потоків при максимальному завантаженні інтерфейсу

Аналіз отриманих характеристик показує, що для перших 20 правил тривалість встановлення становила приблизно 20мс, а середнє значення завантаження центрального процесора рівне 5%. Отже, встановлення правил не впливає на пропускну здатність комутатора.

Дослідження впливу інтенсивності моніторингу на пропускну здатність інтерфейсу та завантаження центрального процесора програмно-керованого комутатора.

Експерименти в цій категорії проводилися таким же ж самим чином, як і в попередньому випадку. На початку експерименту таблиця потоків була порожньою. Максимальний розмір апаратної таблиці потоків становить 1520 правил. Експеримент складався з 1520 ітерацій. У кожній ітерації встановлювалося одне правило, для якого замірювався час встановлення та робився запит до комутатора на предмет завантаження центрального процесора. Упродовж всього експерименту інтерфейс комутатора був завантажений на максимум за допомогою *Iperf*. Результати експерименту відображено на рис.3.8.

Тривалість отримання відповіді на запит щодо надання інформації з таблиці потоків відображено на рис.3.8а. Зі збільшенням розміру таблиці тривалість очікування зростає, при чому мінімально, але зростає і завантаження центрального процесора.



(а) (б)

Рис. 3.8. Залежність тривалості очікування відповіді про стан таблиці потоків залежно від її розмірів (а) та завантаження центрального процесора (б)

У випадку отримання інформації з лічильників інтерфейсів тривалість очікування залишається сталою так само, як і завантаження центрального процесора. В обох випадках зміна інтенсивності не впливає на пропускну здатність інтерфейсу. З урахуванням часу очікування відповіді на запит інтенсивність моніторингу може досягати 5 запитів/секунду.

Розроблення моделі програмно-керованого комутатора HP.

У роботі розроблено модель програмно-керованого комутатора для використання в системах моделювання на основі дискретних подій. Структурну схему моделі представлено на рис.3.9.

Основними категоріями, на які поділяються всі компоненти моделі, є мережеві інтерфейси, апаратна та програмна частини. Компонент мережевого інтерфейсу моделює роботу Ethernet інтерфейсу, містить вхідну та вихідну черги, а також чотири лічильники, що відповідають за облік вхідних та вихідних пакетів/байт. Апаратна частина містить:

- вхідний буфер, поділений на області для зберігання трафіку різних класів;
- таблиці потоків, яких може бути декілька;

процесор пакетів, який виконує операції над пакетом (модифікація заголовку), здійснює керування лічильниками таблиці потоків та відправляє пакет на зазначений вихідний мережевий інтерфейс. Вхідний буфер забезпечує роботу таких алгоритмів: відкидання пакетів для уникнення перевантаження та обслуговування черг (FIFO, PQ, FQ, WQF).

- Програмна частина складається з операційної системи, яка керує апаратною частиною та конфігурує її. Всі дії, пов’язані з налаштуванням та роботою комутатора, за виключенням самого процесу комутації, виконуються операційною системою (обробка сигнальних повідомлень, моніторинг стану апаратної частини). Операційна система містить декілька компонентів, які відповідають за збір інформації про стан всіх інших компонентів комутатора. Менеджер таблиць потоків відповідає за встановлення нових потоків та інші

маніпуляції, пов’язані з таблицями (переміщення потоків між таблицями, модифікація полів окремих правил, вилучення правил).

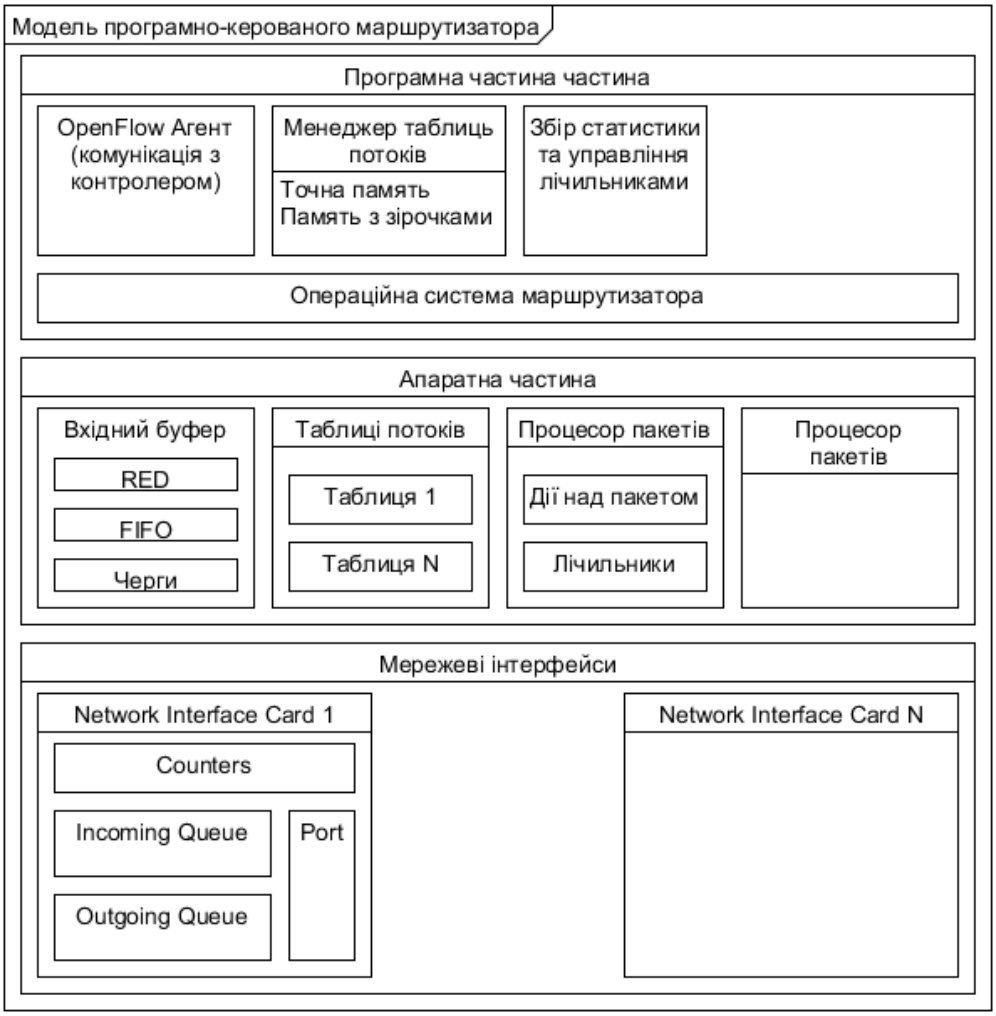


Рис. 3.9. Структурна схема розробленої моделі програмно-керованого апаратного комутатора *HP3500yl*

Для комунікації між контролером та комутатором використовується компонент OpenFlow агент, який містить базовий набір повідомлень відповідно до специфікації OpenFlow 1.0.

3.3. Дослідження ефективності моделі адаптації системи моніторингу

Відповідно до специфікації OpenFlow 1.3, класифікацію повідомлень, що використовуються для пересилання значень параметрів комутатора та його елементів на контролер, відображено на рис.3.10.

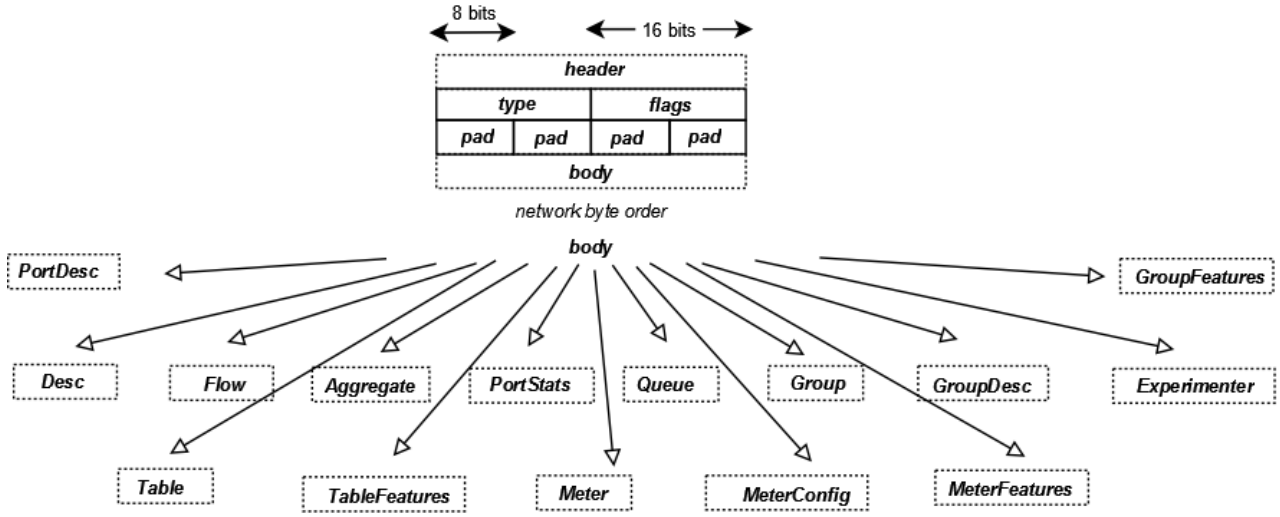


Рис. 3.10. Класифікація повідомлень для здійснення моніторингу параметрів комутатора та його елементів відповідно

Специфікація OpenFlow надає змогу здійснювати моніторинг різних елементів комутатора, проте в роботі увагу зосереджено на повідомленнях, зібраних у табл.3.1.

Таблиця 3.1

Типи повідомлень, використані в роботі, та їхні розміри

Тип опитування	Розмір запиту, байт	Розмір відповіді, байт
Статистика окремого потоку	44	88
Статистика декількох потоків	44	24
Статистика таблиці	8	64
Статистика порту	8	104
Статистика черги	8	32

Для дослідження ефективності запропонованої моделі адаптації системи моніторингу було проведено серію тестів на апаратному комутаторі *HP3500yl*. Конфігурацію тестового середовища подано на рис.3.11, зокрема, відображено агреговане навантаження, яке передається двома шляхами: К1-К2-К4 та К1-К3-К4 між серверами С1 і С2. Метою експерименту є встановити, як частота опитування різних параметрів комутатора впливає на його продуктивність. Мінімально можливий інтервал часу між двома послідовними опитуваннями комутатора *HP3500yl* може становити 1 секунду.

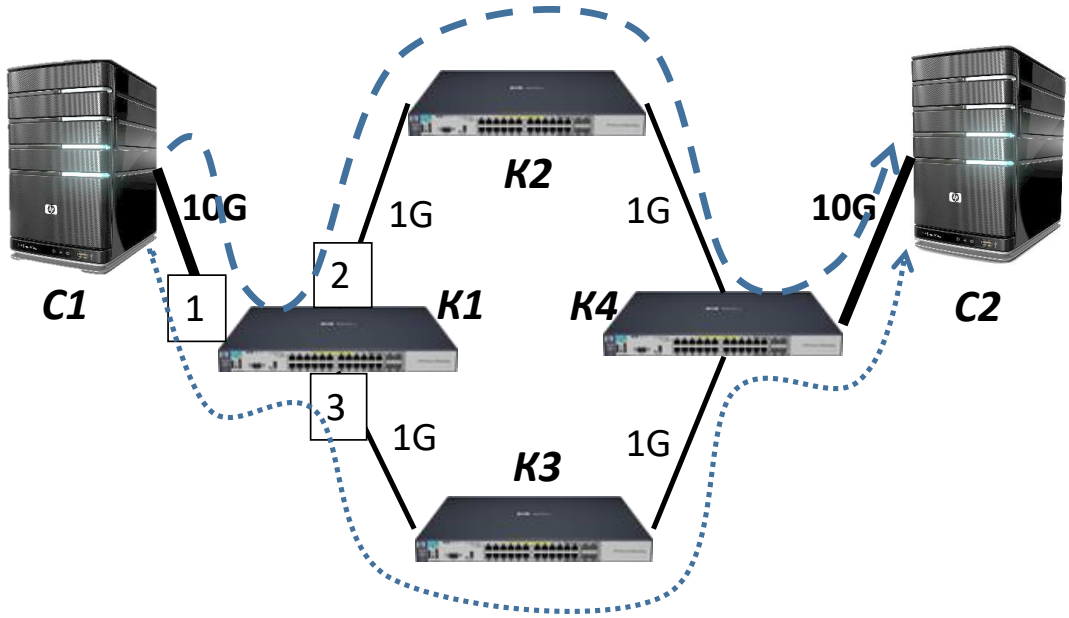


Рис. 3.11. Конфігурація тестового середовища для дослідження впливу інтенсивності моніторингу на пропускну здатність апаратного комутатора *HP3500yl*

Цей інтервал визначено технічним паспортом пристрою. Його може змінювати адміністратор мережі через віддалений термінал у довільних межах, проте не менше від 1 секунди. Для оцінки впливу частоти опитування на продуктивність до комутатора під'єднано два сервери для передачі даних. На серверах було запущено програму *Iperf*, за допомогою якої тестувалася максимальна пропускна здатність інтерфейсу. Пропускна здатність вимірювалася впродовж 1000 секунд. Початковий інтервал моніторингу встановлено рівним 20 секундам. Через кожні 50 секунд експерименту інтервал зменшувався на 1 секунду. У результаті, на 950-ій секунді інтервал моніторингу становив 1 секунду. Було проведено три ітерації цього експерименту. У кожній ітерації проводилося опитування тільки одного типу, а саме: перша ітерація – завантаження інтерфейсів, друга ітерація – статистика потоку, третя ітерація – статистика таблиці потоків. Результати всіх трьох ітерацій представлено на графіках на рис.3.12.

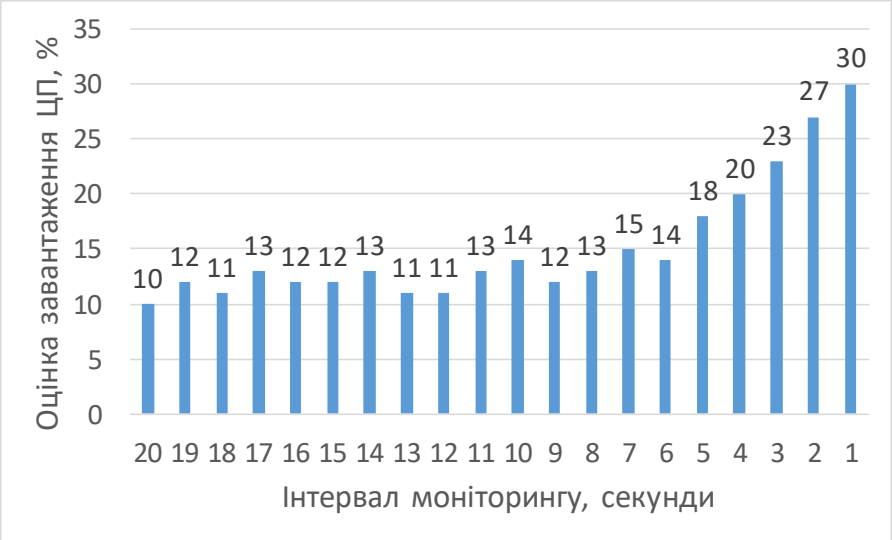


Рис. 3.12. Залежність пропускної здатності інтерфейсу комутатора та завантаження центрального процесора від частоти опитування лічильників інтерфейсу

З отриманих графіків можна зробити висновок, що збільшення інтенсивності моніторингу до максимально можливого значення не впливає на швидкість обробки пакетів і, як результат, на пропускну здатність, проте частково збільшує завантаження центрального процесора.

На основі отриманих характеристик у роботі проведено порівняльне дослідження методів із статичними та адаптивними параметрами моніторингу на апаратному комутаторі *HP3500yl*.

Для цього використовувалося те ж саме тестове середовище, що і в попередньому експерименті, проте програму *Iperf* замінено розробленою в роботі системою генерації мультисервісного трафіку. Всі інтерфейси комутатора характеризуються максимальною пропускною здатністю 1 Гбіт/с. Проте є два додаткових інтерфейси, які підтримують швидкість 10 Гбіт/с кожен. Два сервери під’єднано до двох інтерфейсів комутатора, які виділені в окремий OpenFlow VLAN. Між серверами створено агрегований мультисервісний потік з середньою інтенсивністю, що рівна 70% завантаження каналу. На першому етапі система моніторингу налаштована на роботу в статичному режимі з інтервалом опитування рівним 5 секунд, на другому етапі – на роботу в адаптивному режимі зі змінним інтервалом опитування.

Характеристику агрегованого мультисервісного трафіку представлено на рис.3.13.

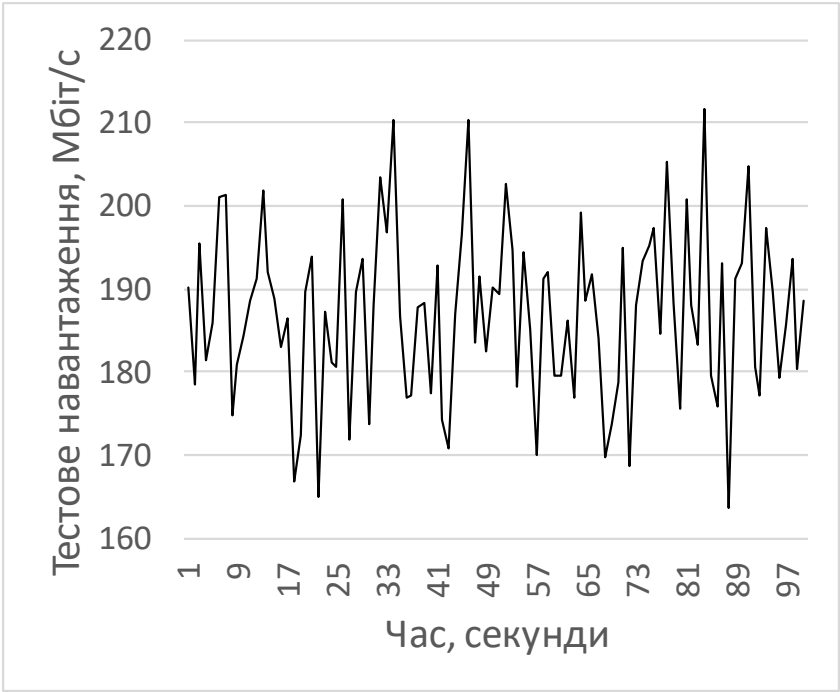


Рис. 3.13. Згенерований мультисервісний агрегований потік на інтерфейс №1 комутатора К1

Результати моніторингу завантаження інтерфейсу комутатора з використанням зазначених методів моніторингу представлено на рис.3.14.

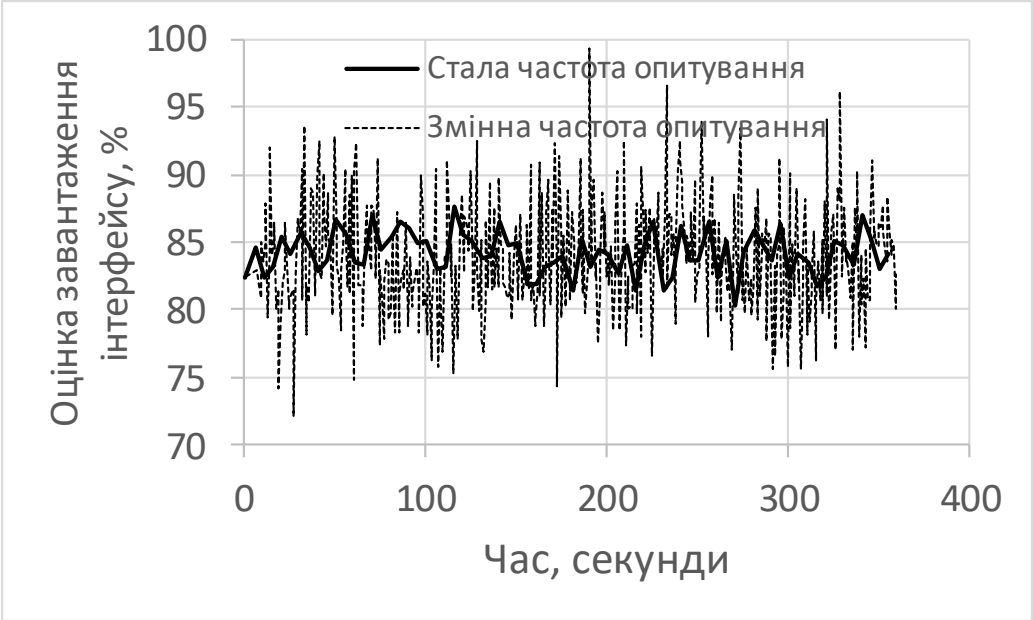


Рис. 3.14. Завантаження інтерфейсу, отримана в результаті використання методів моніторингу зі статичним та змінним інтервалом опитування

Для обох рядів чисел розраховано середнє значення та середньоквадратичне відхилення, значення яких наведено в табл.3.2.

Таблиця 3.2

Статистичні характеристики отриманих значень завантаження інтерфейсу

Інтервал опитування	Середнє значення, %	Середньоквадратичне відхилення, %	Коефіцієнт варіації,%
Сталий (5 с)	84,2018	1,5719	1,86
Змінний (10с – 1с)	83,8865	9,0414	10,77
Сталий (1 с)	84,1581	10,1604	12,07

Середньоквадратичне відхилення показує, що розмах значень завантаження інтерфейсу при статичному методі моніторингу на 9% менший, ніж при методі моніторингу з адаптивною інтенсивністю. У випадку, якщо швидкість інтерфейсу рівна 1 Гбіт/с, 9% становить 90 Мбіт/с. Ця інформація враховується в процесі пошуку шляху для нового потоку. Це означає, що, у випадку використання статичного методу моніторингу, для передачі потоку може бути обраний шлях, характеристики якого, внаслідок короткочасних стрибків інтенсивності навантаження, не дадуть змогу забезпечити необхідну якість обслуговування.

Для перевірки ефективності отриманих результатів проведено два експерименти. Тестове середовище для експериментів відображено на рис.3.15.

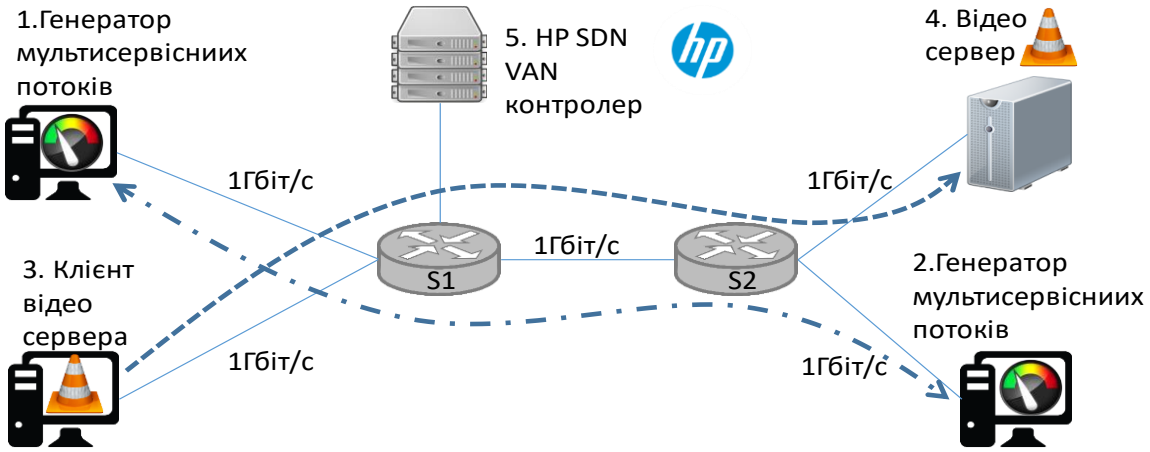


Рис. 3.15. Схема тестового середовища для тестування впливу завантаження каналу на якість обслуговування потоків реального часу

Тестове середовище складається з чотирьох серверів. Розроблену систему генерації мультисервісного трафіку встановлено на першому та другому

серверах. Сервер *VLC* для транслювання відео встановлено на четвертому сервері. Програму-клієнт, яка здійснює запит до відео сервера, встановлено на третьому сервері, а контролер *HP SDN VAN 2.5* – на п'ятому сервері. Характеристика серверів, на яких встановлено компоненти тестового середовища, є такою:

- центральний процесор: *Intel Quad Core 2.5 GHz*;
- оперативна пам'ять: *DDR3 8Gb 1333 MHz*;
- жорсткий диск: *500 Gb*.
- мережевий інтерфейс: 1 Port PCI Express PCIe Gigabit Network Server Adapter.

Топологія мережі утворена двома комутаторами *HP3500yl*. Всі з'єднання характеризуються пропускною здатністю 1 Гбіт/с. Така конфігурація забезпечує виникнення вузького місця між першим та другим комутаторами, що потрібно в цьому експерименті для дослідження впливу високого завантаження інтерфейсу на якість надання послуг реального часу.

Між сервером 1 та 3 циркулюють потоки даних у пропорції (2 *IPTV*, 5 *VoIP*, 1 *Skype Video call*, 130 *HTTP*, 20 *Cloud*). Потоки *IPTV* спрямовані до відео сервера. Разом вони завантажують канал у середньому на 86%. Експеримент полягає в оцінюванні рішення контролера щодо відкриття нового *IPTV* потоку з сервера відео до клієнта. На першому етапі експеримент проводився з використанням статичної моделі моніторингу з інтервалом моніторингу, рівним 5 секунд. Контролер оцінював можливість використання каналу для маршрутизації нового *IPTV* потоку. На другому етапі використовувалася модель моніторингу зі змінною частотою опитування з мінімальним інтервалом тривалістю 1 секунда. Розподіл інтенсивності вхідного навантаження представлено гістограмою на рис.3.16. Розподіли завантаження каналу між комутаторами 1 та 2 відображено на рис.3.17 та рис.3.18.

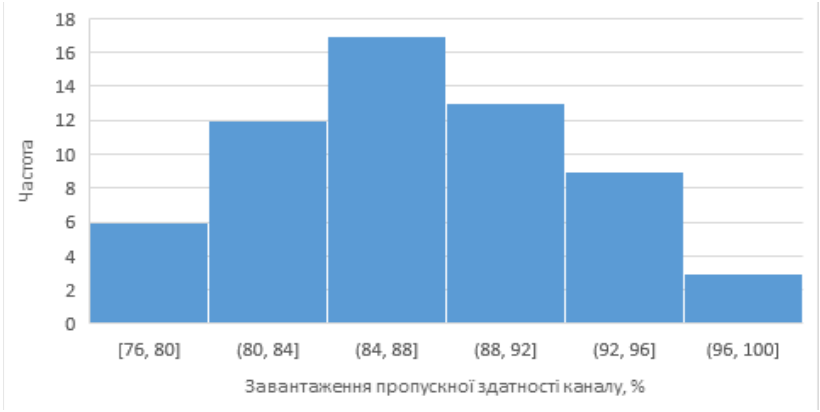


Рис. 3.16. Розподіл інтенсивності вхідного навантаження на канал між комутаторами 1 та 2

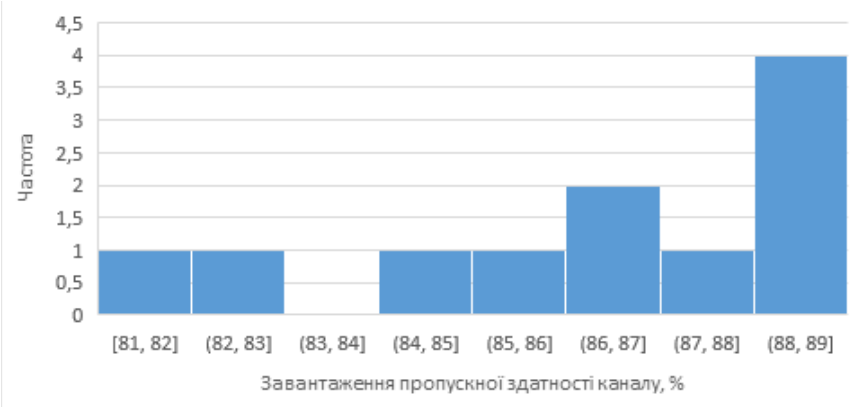


Рис. 3.17. Розподіл завантаження каналу між комутаторами 1 та 2, отриманий в результаті використання моделі моніторингу зі сталою частотою опитування

Рис. 3.18. Розподіл завантаження каналу між комутаторами 1 та 2, отриманий в результаті використання моделі моніторингу зі змінною частотою опитування

З отриманих гістограм можна зробити висновок, що у випадку сталої частоті опитування контролер не має змоги врахувати стрибки інтенсивності навантаження. Для статичного методу середнє значення завантаження каналу

становить 86,56%, а стандартне відхилення – 2,43%. Для адаптивного моніторингу середнє значення завантаження каналу становить 86,91%, а стандартне відхилення – 5,43%. У результаті, для розрахунку ймовірності перевантаження каналу береться верхнє значення завантаження. Розрахунок імовірності блокування для методу зі статичною (3.1) та з адаптивною (3.2) інтенсивністю моніторингу:

$$p_{blocking} = \frac{86{,}56 + 2{,}43}{100} = 0.8899, \tag{3.1}$$

$$p_{blocking} = \frac{86{,}91 + 5{,}43}{100} = 0.9234. \tag{3.2}$$

У результаті оцінки ймовірності блокування шляху за результатами методу моніторингу з адаптивною інтенсивністю встановлено, що ймовірність блокування є вищою. Завантаження буферів практично відсутнє і не перевищує одного пакету для кожної черги.

У тому випадку, якщо контролер все ж таки прокладе додатковий IPTV потік каналом між комутаторами 1 та 2 з описаними вище умовами, характеристика його завантаження буде виглядати таким чином, як показано на (рис.3.19).

Рис. 3.19. Характеристика завантаження каналу між комутаторами 1 та 2 у випадку прокладання додаткового IPTV потоку

Оцінка якості обслуговування цього IPTV потоку, а також його вплив на всі ніші потоки в мережі охарактеризовано у табл. 3.3.

Таблиця 3.3

Характеристика впливу перевантаження каналу на якість обслуговування

Основний трафік	Фоновий трафік	Параметри якості обслуговування		
		Середня затримка основного потоку, секунди	Середньоквадратичне відхилення затримки, секунди	Втрати основного потоку, %
IPTV	Відсутній	0,002733015	0,000476221	0
Мультисервісний	Відсутній	0,00614871	0,004314631	0
IPTV	Мультисервісний	0,005993914	0,003437946	0,3
Мультисервісний	IPTV	0,007400735	0,005225187	0,73

У табл.3.3 в перших двох рядках відображено середню затримку, середньоквадратичне відхилення та втрати окремо для IPTV потоку та фонового трафіку, експерименти для яких проводилися незалежно. Третій рядок містить характеристики для IPTV потоку, який передавався разом з фоновим трафіком, а четвертий рядок – відповідні характеристики фонового трафіку.

Аналізуючи отримані результати, можна зробити висновок, що для *IPTV* потоку затримка зросла майже вдвічі, а джиттер збільшився в 10 разів. Також з'явилися втрати як для IPTV потоку, так і для потоків фонових даних. Це пов'язано з тим, що канал між першим та другим комутатором перевантажений і не в змозі передати вхідне навантаження з першого та другого сервера.

В останньому експерименті проведено оцінку обсягів сигнальної інформації, створеної внаслідок використання статичної та адаптивної моделі моніторингу. На рис.3.20 представлено зміну частоти опитування трьох інтерфейсів першого комутатора протягом 60 секунд.

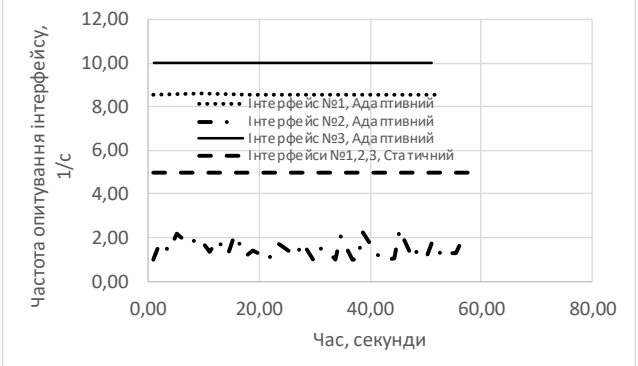
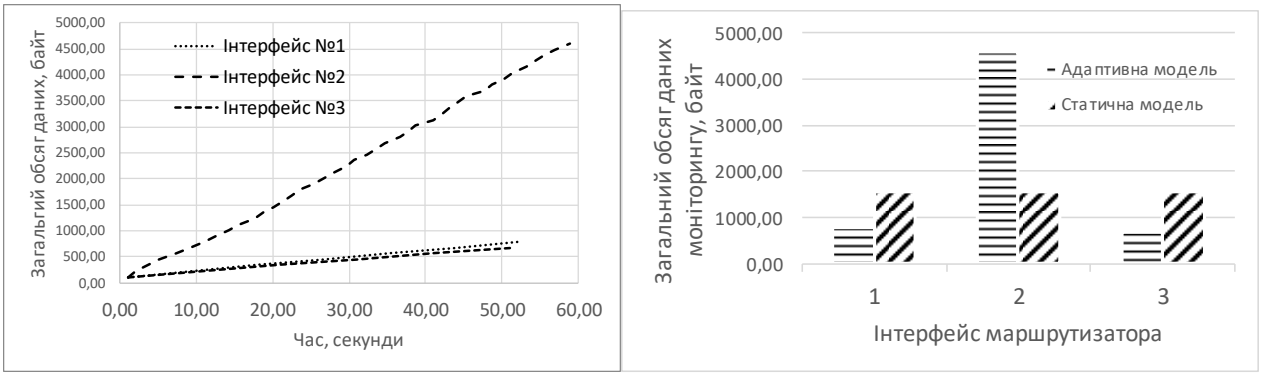


Рис. 3.20. Інтенсивність моніторингу трьох інтерфейсів

Порівняння обсягів сигнальної інформації, згенерованої для трьох інтерфейсів першого комутатора, представлено на рис.3.21.



(а) (б)

Рис. 3.21. Порівняння обсягів сигнальної інформації для трьох інтерфейсів першого комутатора

На рис.3.21а представлено зростання кількості сигнальної інформації для кожного інтерфейсу першого комутатора індивідуально, на рис.3.21б – гістограму порівняння обсягів згенерованої інформації за весь час вимірювання для двох методів моніторингу. Аналізуючи гістограму на рис.3.21б, можна стверджувати, що для завантаженого інтерфейсу обсяг сигнальної інформації, згенерованої з використанням адаптивної моделі моніторингу, суттєво перевищив такий обсяг у випадку використання статичної моделі. Проте, за рахунок низької частоти опитування незавантажених інтерфейсів сумарний обсяг сигнальної інформації загалом практично однаковий в обох експериментах.

3.4. Дослідження ефективності удосконаленої моделі балансування навантаження

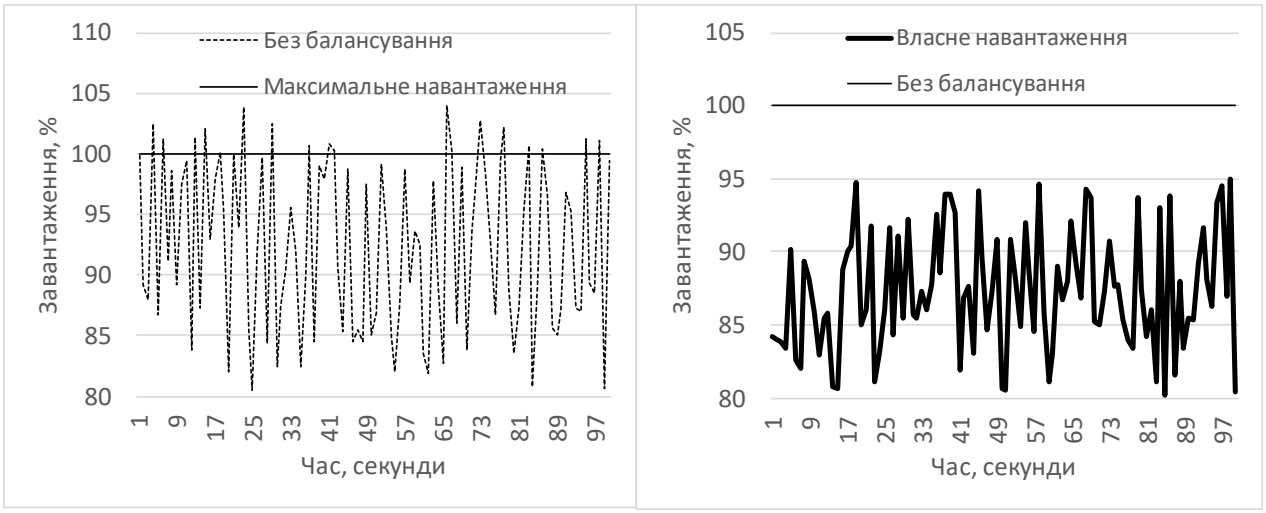
У роботі вдосконалено модель балансування навантаження, яка за критерій перенаправлення пакетів використовує не відносний пріоритет шляху, а максимально допустиме завантаження наступного каналу. Відповідно до такого критерію комутатор направляє всі пакети потоку в один канал, якщо його завантаження не перевищує максимально допустиме. У випадку, коли сумарна кількість біт, переданих в канал за одну секунду, перевищує 90% пропускної

здатності, для передавання наступного пакету вибирається інший доступний шлях. Якщо доступного шляху не знайдено, то пакет відкидається.

Для аналізу ефективності роботи запропонованого методу балансування навантаження мультипотокового керування трафіком проведено експеримент, схему якого відображено на рис.3.11. Для експерименту згенеровано навантаження, яке використовувалося для оцінювання ефективності адаптивної моделі моніторингу (рис.3.13).

На першому етапі досліджувався параметр втрати пакетів без балансування навантаження. Для цього на перший інтерфейс комутатора К1 згенеровано мультисервісне навантаження з використанням розробленої системи генерації мультисервісного трафіку, інтенсивність якого відображено на рис.3.13. У зв'язку з тим, що інтенсивність навантаження є більшою від 1 Гбіт/с і не може бути передана до сервера С2 одним каналом, контролер проклав два шляхи в мережі, а саме: С1 – К1-К2-К4 – С2 та С1 – К1-К3-К4 – С2. Маршрутизація реалізована з використанням протоколу EIGRP. Ураховуючи високу інтенсивність навантаження, контролер розділив агрегований трафік з інтерфейсу К1 на два потоки майже однакового розміру. У результаті, в процесі передавання даних проводився моніторинг завантаження всіх портів 1-го комутатора. Завантаження 2-го та 3-го інтерфейсів комутатора К1 відображено на рис.3.22.

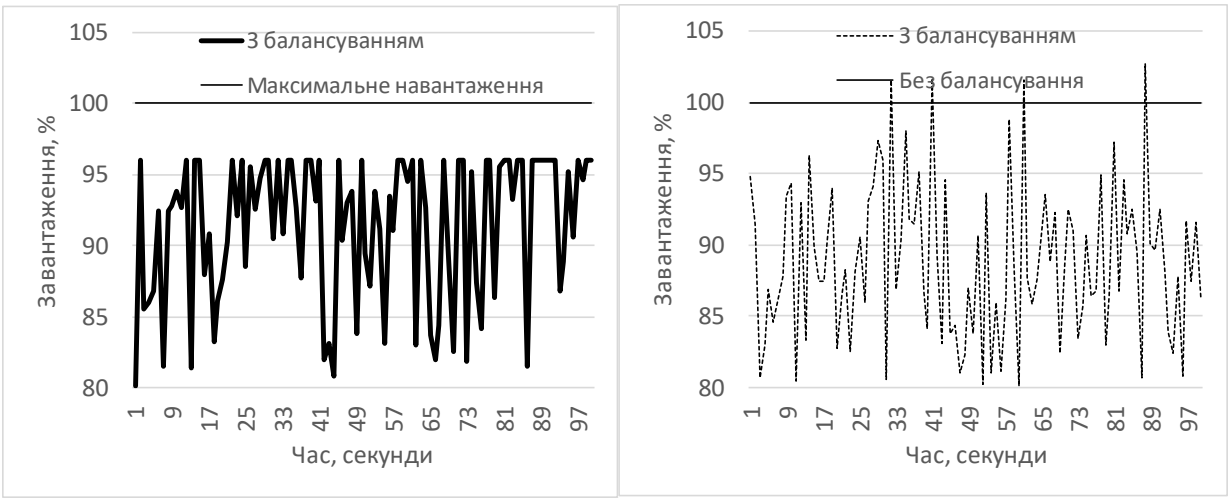
У ході експерименту спостерігаються втрати в каналі К1-К2-К4 (рис.3.22). В результаті експерименту втрати пакетів у каналі К1-К2-К4 становили 10%. Це означає, що, хоч протокол EIGRP і забезпечив балансування навантаження з пропорційним розподілом трафіку між допустимими каналами, існуючий механізм балансування навантаження на основі групової таблиці потоків не дав змогу уникнути втрат у моменти різких стрибків інтенсивності вхідного навантаження.



(а) (б)

Рис. 3.22. Завантаження каналів К1-К2-К4 (а) та К1-К3-К4 (б) при маршрутизації та балансуванні навантаження на основі протоколу EIGRP

На другому етапі проведено дослідження удосконаленої моделі балансування навантаження. У цьому випадку максимально допустимий рівень завантаження каналу К1-К2-К4 становив 95% від його пропускної здатності. Для каналу К1-К3-К4 ніякого обмеження не встановлено. Для цього дослідження використовувалося навантаження, ідентичне до попереднього експерименту. Результати проведеного експерименту відображено на рис .3.23.



(а) (б)

Рис. 3.23. Завантаження каналів К1-К2-К4 (а) та К1-К3-К4 (б) при маршрутизації на основі протоколу EIGRP з використанням удосконаленої моделі балансування навантаження

Після застосування запропонованої моделі балансування завантаження каналу К1-К2-К4 не перевищувало 95% без втрат пакетів. Весь надлишковий трафік перенаправлявся в канал К1-К3-К4, завантаження якого в середньому також не перевищувало 95%, проте в деякі моменти досягало 100%, у зв'язку з чим втрати в каналі К1-К3-К4 становили 1,5%. Це пояснюється тим, що трафік не міг бути переданим ні одним шляхом внаслідок перевантаження.

У результаті експерименту встановлено, що удосконалена модель балансування навантаження, за умов середнього завантаження каналів на рівні 90% та середньоквадратичного відхилення інтенсивності вхідного навантаження 9,981%, дає змогу знизити втрати в середньому в 6 разів.

Висновки до 3-го розділу

У третьому розділі проведено дослідження характеристик функціонування апаратного OpenFlow комутатора *HP3500yl*. Особливістю комутатора є використання як апаратної, так і програмної таблиці потоків, при чому, остання характеризується суттєво нижчою пропускною здатністю. У результаті дослідження отримано характеристики пропускної здатності комутатора залежно від режимів його роботи (стандартний, OpenFlow програмний та апаратний), а також залежно від розміру пакету, на основі яких можна стверджувати, що апаратна комутація в обох режимах характеризується однаковою продуктивністю, в той час, як програмна комутація знову показує в рази меншу продуктивність. При чому, продуктивність у випадку програмної комутації зростає лінійно та в незначному діапазоні, тоді як апаратна комутація характеризується продуктивністю, що змінюється нелінійно.

Також отримано характеристики тривалості встановлення нових потоків в таблицю та тривалість отримання відповіді на запит системи моніторингу. Визначено залежність між інтенсивністю моніторингу та завантаженням центрального процесора, а також інтенсивністю встановлення нових правил та завантаженням центрального процесора. Встановлено, що моніторинг не впливає на пропускну здатність навіть у найбільш інтенсивному режимі, проте

підвищує завантаження центрального процесора до 30%. На основі отриманих характеристик розроблено модель OpenFlow комутатора для симуляторів на основі дискретних подій. Проведено дослідження запропонованого методу балансування навантаження та підтверджено, що розроблений метод, за умов середнього завантаження каналів на рівні 90% та середньоквадратичного відхилення інтенсивності вхідного навантаження 9,981%, дає змогу знизити втрати в середньому в 6 разів.

На основі експерименту підтверджено, що розроблений метод адаптації системи моніторингу дає змогу забезпечити адекватність характеристики процесів передачі трафіку, зокрема завантаження мережевих інтерфейсів, що надає можливість точніше оцінити ймовірність блокування та уникнути перевантаження каналів у разі використання динамічної маршрутизації. На основі експерименту встановлено, що розмах значень завантаження інтерфейсу при статичному методі моніторингу на 9% менший, ніж при методі моніторингу з адаптивною інтенсивністю. У випадку, якщо швидкість інтерфейсу рівна 1 Гбіт/с, 9% становить 90 Мбіт/с. Ця інформація враховується в процесі пошуку шляху для нового потоку. Це означає, що, у випадку використання статичного методу моніторингу, для передачі потоку може бути обраний шлях, характеристики якого, внаслідок короткочасних стрибків інтенсивності навантаження, не дадуть змогу забезпечити необхідну якість обслуговування.

РОЗДІЛ 4. ПРАКТИЧНА РЕАЛІЗАЦІЯ РОЗРОБЛЕНИХ МОДЕЛЕЙ ТА АЛГОРИТМІВ ПРОГРАМНО-КОНФІГУРОВАНОЇ МЕРЕЖІ

У четвертому розділі проведено дослідження розробленого методу вимірювання затримки та продемонстровано його ефективність порівняно з існуючим методом *Traceroute*. Досліджено удосконалену модель маршрутизації та продемонстровано ефективність її застосування порівняно з протоколом EIGRP у ситуаціях, коли окремі канали в мережі близькі до перевантаження.

4.1. Конфігурація тестового середовища з використанням апаратних комутаторів *HP3500yl*

Для побудови мережі тестового середовища використано 6 комутаторів моделі HP 3500yl-24G J8692A ($K.15.17.0007$) та 1 комутатор HPE FF 5700-32XGT-8XG-2QSFP+ ($2422P01$). Всі елементи тестового середовища зведено в табл.4.1.

Таблиця 4.1

Елементи тестового середовища

Параметр	Конфігурація
Пропускна здатність ребра	1 Гбіт/с
Кількість портів комутатора	24
Кількість серверів	12
Кількість комутаторів	7
Кількість ребер	10
Контролер	HP SDN VAN 2.5

Всі з’єднання між комутаторами та комутаторами і серверами мають швидкість 1 Гбіт/с. Схему тестового середовища представлено на рис .4.1.

В основі тестового середовища лежить мережева топологія типу зірки, сформована з 6-ти комутаторів *HP3500yl*, які розташовані по краях топології. У центрі розташовано комутатор *HP5700*. До крайових комутаторів підключено по два сервери, кожен з яких має такі характеристики:

- центральний процесор: Intel® Xeon® E5-2600 v4 product family;
- кількість процесорів: 1;

- кількість ядер: 16;
- form factor (fully configured): 1U;
- оперативна пам’ять: 128GB;
- кількість слотів оперативної пам’яті: 8 DIMM slots;
- тип оперативної пам’яті: DDR4 SmartMemory;
- опис жорсткого диску: SFF SAS/SATA/SSD;
- мережевий адаптер: 1Gb 361i Ethernet Adapter 0.

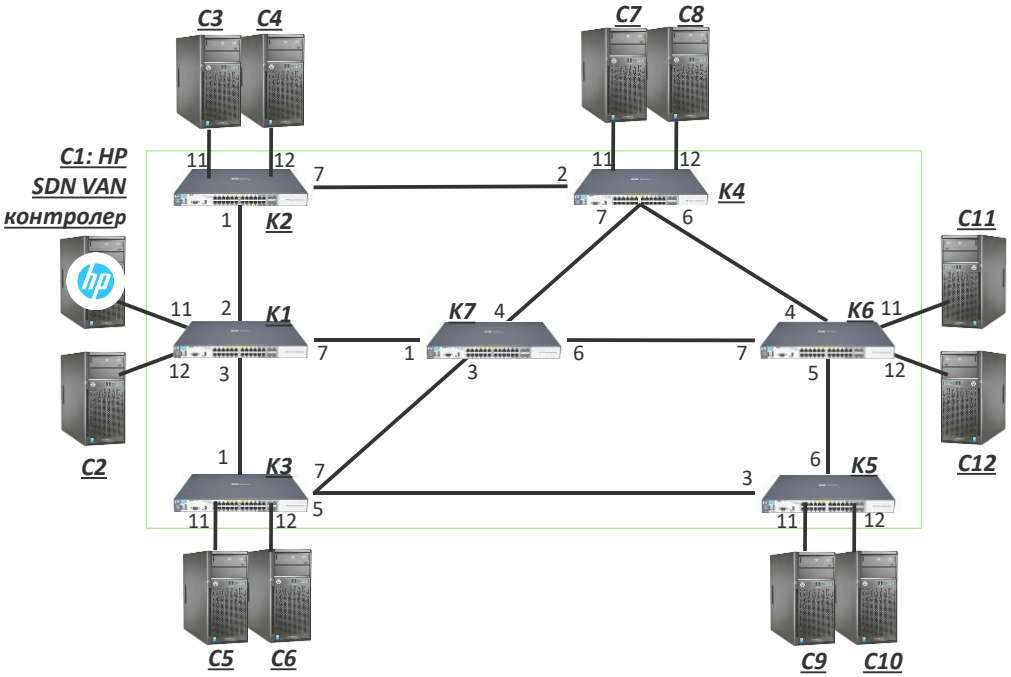


Рис. 4.1. Схема тестового середовища для оцінки ефективності методів підвищення якості обслуговування на основі реальних апаратних пристроїв

Комутатори *HP3500yl* забезпечують високу продуктивність та простоту використання на межі корпоративних мереж. Вони характеризуються низькою затримкою та оптимізованими механізмами буферизації. Стандартна конфігурація комутатора може складатися з 24 або 48 портів, кожен з яких підтримує максимальну пропускну здатність 1 Гбіт/с. Крім того, комутатор має можливість розширення конфігурації за рахунок двох додаткових портів по 10 Гбіт/с кожен. Ці порти можуть використовуватися для агрегації трафіку. Мережевий рівень представлено такою функціональністю: IPv4 BGP, політика маршрутизації, IPv4/IPv6 OSPF, управління якістю обслуговування. Комутатор

підтримує функціональність OpenFlow версії 1.0 та 1.3. Комутатор *HP3500yl* є гібридним, тобто може паралельно працювати в обох режимах: традиційному з використанням децентралізованих протоколів управління та OpenFlow. Для використання порту в режимі OpenFlow його необхідно зробити членом віртуальної локальної мережі, яка контролюється одним з локальних екземплярів OpenFlow. Зведену інформацію про комутатори представлено в табл.4.2

Таблиця 4.2

Характеристика комутаторів, використаних для побудови тестового середовища

Модель комутатора	Версія операційної системи	Пропускна здатність, мегапакетів/с	Пропускна здатність комутаційної матриці, Гбіт/с	Затримка комутації, мкс
HP 3500yl-24G (J8692A)	K.15.17.0007	75,7	101,8 Gbps	3,4
HPE FF 5700-32XGT-8XG-2QSFP+ (JG898A)	2422P01	714,2	960,0 Gbps	1,5

Комутатор *HP FlexFabric 5700 Switch Series* призначений для використання в корпоративних мережах та дата центрах. Він забезпечує високу масштабованість за рахунок функцій локальної комутації, що реалізована засобами додаткової апаратної таблиці з високою місткістю та продуктивністю. Комутатор підтримує порти 1, 10 та 40 Гбіт/с. Комутаційна фабрика забезпечує пропускну здатність 960 Гбіт/с, а відповідно й дуже низьку затримку (1,5 мкс).

Для управління мережею використано контролер *HP SDN VAN* версії *2.5.2*. Його встановлено на сервері №2. На цьому ж сервері встановлено систему моніторингу, розроблену в роботі, а також систему генерації мультисервісного трафіку. Остання використовувала всі інші сервери для генерації трафіку.

4.2. Проблеми інтеграції компонентів системи управління та різних версій операційної системи комутаторів

ПКМ є відносно новою концепцією, тому деякі архітектурні аспекти все ще перебувають на стадії дослідження та розробки, наприклад, визначення

оптимальної кількості контролерів для управління мережею. У роботі [54] представлено дослідження оптимальної кількості контролерів та їхнього фізичного розміщення для управління мережею. На основі аналізу декількох реально існуючих мережних конфігурацій встановлено, що в більшості випадків достатньо одного контролера для виконання основних завдань, пов'язаних з керуванням мережею. У деяких роботах пропонується розподілена система контролерів (рис.4.2), які синхронізуються та функціонують як логічна одиниця. Наприклад, *HyperFlow* [55] – це додаток, побудований на основі контролера *NOX* [56]. Для забезпечення масштабованості мережі *NOX* використовує декілька контролерів, встановлених на кількох серверах у мережі. Кожен контролер керує власним доменом, який складається з обмеженої кількості комутаторів. *HyperFlow*, встановлений на кожному з контролерів, оперує узагальненою мережевою топологією, що складається з відповідних доменів. Платформа *Onix* [57] пропонує три способи забезпечення масштабованості мережі: по-перше, керуючі додатки мають змогу розділяти навантаження таким чином, що збільшення кількості додатків призводить до покращення продуктивності мережі, а завдання оптимально розподіляються між усіма компонентами; по-друге, мережа під контролем *Onix* представляється для всіх інших систем як єдиний обслуговуючий вузол, що дає змогу створювати ієрархічну структуру управління; по-третє, *Onix* надає всім іншим додаткам, які використовують мережу, можливість отримувати інформацію про стан мережі та її зміни, а також змогу приймати власні рішення щодо управління.

Централізований контролер не має проблем з масштабованістю і може обробити весь трафік, що на нього надходить. Наприклад, контролер *Maestro* [58] побудований з метою вирішення проблем масштабованості. Він використовує віртуалізацію для того, щоб досягнути практично лінійного зростання продуктивності на багатоядерних системах. Іншим покращенням у цьому напрямку є багатопотоковий контролер *NOX-MT* [59]. Автори статті [59] використовують додаток *cbench* для тестування та вимірювання продуктивності

оброки нових потоків, їх затримки та пропускної здатності. Ураховуючи, що *NOX-MT* здатний обробити 1,6 мільйонів запитів за секунду з середнім часом відгуку, рівним 2 мілісекунди, автори зазначають, що цей показник повинен використовуватися як нижнє граничне значення продуктивності одного контролера.

Більшість компонентів та додатків у програмно-керованих мережах функціонують як частина ядра контролера. Таким чином, вони мають доступ до всієї інформації про мережу, включаючи інформацією маршрутизації. Вони також мають змогу напряму спілкуватися з будь-яким комутатором у мережі. Більшість розроблених контролерів має програмний компонент для пошуку комутаторів та побудови топології мережі включно зі збором інформації про характеристики комутаторів. Цей компонент відповідає за створення структур даних, які відображають комутатор. Для кожного нового комутатора в мережі створюється така структура даних. Кожна структура зберігає інформацію про особливості конкретного комутатора. Контролер повністю готовий до керування мережею в тому разі, коли компонент для вивчення мережі та побудови шляхів передачі даних запущений і працює без помилок. Коли ці компоненти запущені, додаток реєструє свій процес в ядрі контролера і таким чином отримує інформацію про зміни в топології чи параметрах мережі в асинхронному режимі. Все, що відомо контролеру, так само стає відомим й іншим додаткам: активні комутатори та їх параметри, активні потоки та глобальна інформації маршрутизації, стан каналів.

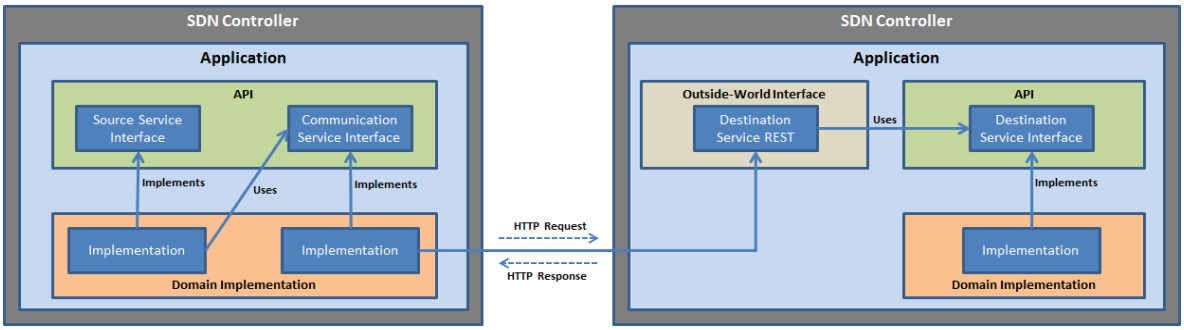


Рис. 4.2. Архітектура та інтеграція розподілених контролерів *POX*

Ядро контролера *POX* надає велику кількість функцій. Основною метою центрального компонента цього контролера є забезпечення взаємодії між різними компонентами. Замість того, щоб імпортувати велику кількість компонентів з перехресними посиланнями, центральний компонент реєструє посилання на всі компоненти, які потребують отримувати інформацію про зміни в мережі, та інформує їх у випадку відповідних змін. Це забезпечує гнучкий спосіб заміни компонентів та розширення функціональності. Окрім реєстрації компонентів, центральний компонент підтримує низку функцій, які охарактеризовано нижче.

Ініціація подій. У контролері певна частина коду очікує на події, які можуть виникнути в мережі (новий потік). Кожна подія характеризується типом та параметрами, які зберігаються у сформованому об'єкті. Ядро контролера генерує відповідний об'єкт при виникненні події та передає його на опрацювання у всі методи, які зареєструвалися та очікують на виникнення події саме цього типу.

Робота з пакетами. Контролер надає засоби для легкого та швидкого аналізу OpenFlow повідомлення чи IP пакету, а також для створення власних пакетів з необхідними полями заголовків.

Потоки та задачі. Контролер використовує власну бібліотеку для забезпечення взаємодії між компонентами, що дає змогу розробникам додатків не турбуватися про проблеми синхронізації.

Однією з основних обов'язків контролера є взаємодія з OpenFlow комутаторами. З цією метою окремий компонент (*openflow.of_01*) реєструється в центральному компоненті одразу після запуску контролера. Інший компонент (*openflow.discovery*) використовує протокол *LLDP* для вивчення мережевої топології. Кожного разу, коли новий комутатор з'являється в мережі, подія *ConnectionUp* генерується ядром та наповнюється відповідна структура даних, на основі якої створюється об'єкт Connection, що використовується всіма додатками для взаємодії з цим комутатором. Кожному комутатору присвоюється власний ідентифікаційний номер *DatapathID* (*DPID*). Значення

цього ідентифікатора може бути встановлене довільно, наприклад, на основі MAC адреси інтерфейсу комутатора. Подія реєстрації нового комутатора спричиняє створення об'єкту *Switch*. Цей об'єкт містить ідентифікатор комутатора та прослуховує всі події, які можуть виникнути в результаті роботи цього комутатора. Компонент відповідає за вивчення топології мережі, будує граф, який відображає фізичну мережу, та слідкує за її змінами, а саме – у разі виникнення подій *ConnectionUp* та *ConnectionDown*.

Компонент маршрутизації відповідає за обробку повідомлень *PacketIn*. Кожного разу, коли крайовий комутатор реєструє невідомий пакет, для якого немає правила в таблиці потоків, комутатор відправляє на контролер повідомлення, що містить заголовки пакету та ідентифікатор самого пакету для його локалізації в пам'яті комутатора. Ця подія повідомляє контролер про те, що в мережі зареєстровано невідомий новий потік, для якого потрібно прокласти маршрут. Шлях може розраховуватися за допомогою будь-якого алгоритму маршрутизації. Правила можуть встановлюватися в реактивному та проактивному режимах. Контролер створює шлях для нового потоку, встановлюючи відповідні правила на комутаторах вздовж цього шляху в мережі. Інформація про кожен потік зберігається компонентом до тих пір, поки не виникне подія, що сигналізує про видалення потоку з мережі *FlowRemoved*. Це трапляється тоді, коли комутатор видаляє потік з таблиці, якщо час його існування вичерпано.

Контролери відомих виробників мережевого обладнання надають розширений програний інтерфейс для створення різноманітних додатків. Контролери, написані на мові *Java* (*OpenDayLight, FloodLight та HP VAN SDN*), дають змогу динамічно встановлювати нові додатки. Така інтеграція можлива за рахунок використання технології OSGi, в основі якої лежить компонентний принцип побудови програмного забезпечення. Архітектуру контролера *HP SDN VAN* зображено на рис.4.3. Кожен компонент характеризується інтерфейсом, формалізованим за допомогою засобів *XML*. За рахунок цього компоненти можуть без проблем бути видалені чи додані в процесі роботи системи. Одним

з варіантів реалізації запропонованої системи моніторингу в роботі є створення такого компонента та інтеграція його в контролер. Проте, в такому випадку для кожного контролера необхідно буде створити окремий додаток, оскільки програмні інтерфейси та структури даних кожного з них різні.

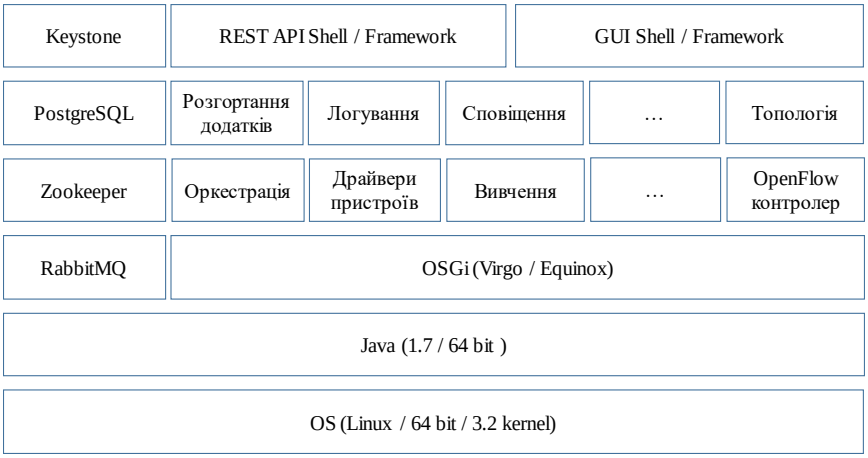


Рис. 4.3. Архітектура контролера HP SDN VAN

З іншого боку, така інтеграція могла б забезпечити високу швидкодію та надійність, оскільки ці функції бере на себе сам контролер. У випадку інтеграції система моніторингу використовуватиме структури даних конкретного контролера та заповнюватиме їх відповідно до зібраної інформації. Позитивним аспектом тут є можливість легкого розширення структур під власні потреби.

Інший варіант інтеграції полягає у використанні північного інтерфейсу, який надається більшістю контролерів у формі REST API. У такому разі комунікація між контролером та системою моніторингу буде відбуватися віддалено. Структура повідомлень та функціональні можливості через цей інтерфейс визначаються виробником самого контролера. Зазвичай функціональність цього інтерфейсу співпадає з функціональністю OpenFlow протоколу, тому для інтеграції системи моніторингу та контролера в такий спосіб достатньо завантажити структури повідомлень та замінити їх у відповідному модулі, який відповідає за комунікацію. Ще одним важливим аспектом використання цього інтерфейсу є можливість інтегрувати елемент системи моніторингу та встановити його локально на кожному комутаторі,

тобто розташувати його фізично ближче до мережевих пристроїв (сам контролер може знаходитися не в підконтрольній ним фізичній мережі). Основним недоліком такого способу є низька швидкодія, оскільки значна частина часу затрачається на створення та аналіз повідомлень у форматі JSON та на пересилання таких повідомлень між контролером і системою моніторингу.

У роботі вибрано останню модель інтеграції для контролера HP SDN VAN. Вона забезпечила достатню швидкодію обміну повідомленнями та реагування мережі. Для прикладу, тривалість встановлення одного правила фіксувалася системою моніторингу з моменту відправлення повідомлення на контролер до моменту отримання відповіді від контролера з результатом операції. Мінімальна тривалість становила 20мс. Варто зауважити, що ця тривалість не є тривалістю реакції контролера на події в мережі, а лише відображає затримку впливу системи моніторингу на мережу з використанням північного інтерфейсу контролера. Ця затримка без сумніву залежить від фізичного розташування обох елементів. Система моніторингу може бути встановлена на тому ж сервері, що й контролер. А це зведе до мінімуму тривалість комунікації, оскільки повідомлення будуть передаватися між процесами всередині однієї операційної системи, не проходячи через весь мережевий стек. У такому випадку затримка може становити до 2 мс.

4.3. Експериментальна оцінка точності вимірювання затримки передавання пакетів у програмно-конфігурованій мережі.

Зазвичай двома основними засобами вимірювання затримки між двома вузлами, а також оцінювання тривалості передачі пакету по певному шляху є *Ping* та *Traceroute*. *Ping* дає змогу встановити доступність вузла призначення, а також надає статистичні дані про втрати пакетів та час обходу петлі пакетом. *Traceroute* показує всі вузли на шляху від вузла-відправника до вузла-отримувача, а також тривалість обходу петлі від вузла-ініціатора вимірювання до кожного вузла вздовж шляху. У випадку, коли стоїть задача оцінювання

затримки в мережі, дуже важливо враховувати географічну відстань між двома вузлами, оскільки більша відстань зумовлює довший час передачі. Як тільки проблемні вузли виявлено, використовується додаток *Ping* для більш детального дослідження затримки між двома вузлами. Для прикладу, *Traceroute* дає змогу виявити вузол, на якому саме починається проблемна ділянка. Приклад роботи додатку *Traceroute* представлено на рис.4.4.

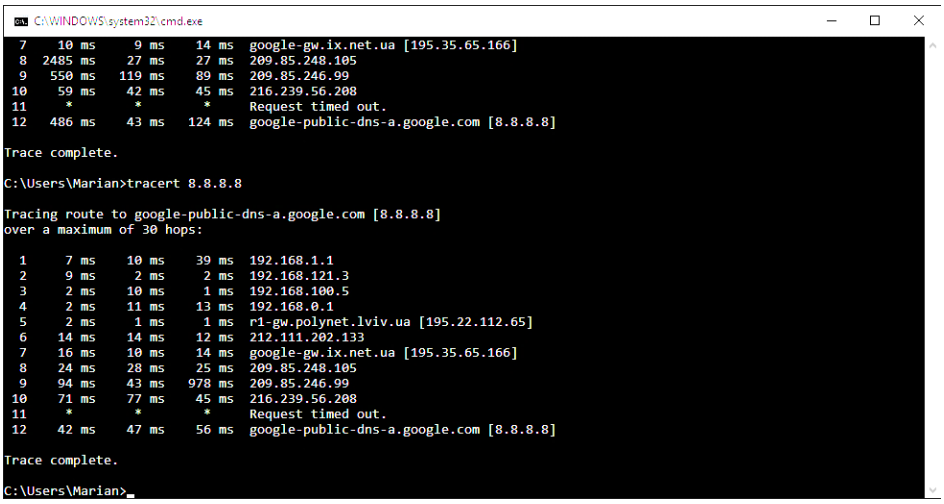


Рис. 4.4. Приклад роботи додатку *Traceroute* для вимірювання затримки та визначення перевантаженої ділянки мережі

Оцінювання затримки передачі пакету між двома комутаторами проводиться на основі розробленого в роботі алгоритму вимірювання затримки з кінця в кінець. Проте в цьому випадку суттєві похибки виміряного значення можуть виникати внаслідок великої кількості процесів, кожен з яких вносить свою затримку. Для точного оцінювання значення затримки між двома вузлами необхідно врахувати такі процеси:

- тривалість обробки комутатором пакету типу “PacketOut” відповідно до специфікації OpenFlow;
- тривалість передачі пакету контролером по мережевому стеку з моменту фіксації часу відправлення пакету;
- тривалість отримання пакету, його розпакування та аналіз вмісту при його отриманні з кінцевого комутатора;
- поточне завантаження комутатора та каналу;

- поточне завантаження центрального процесора комутатора.

Для оцінювання всіх випадкових впливів проведемо експеримент щодо оцінювання затримки передачі пакету. Схему передачі пакету відображено на рис.4.5.

Контролер використовує два типи повідомлень OpenFlow для зчитування пакетів з площини передачі даних та для введення пакетів у цю площину. При чому, обидва повідомлення характеризуються певними особливостями використання.

Повідомлення типу *PacketIn* використовується для упаковування пакетів даних та передачі їх від комутатора до контролера. Є дві причини, що можуть призвести до генерації цього повідомлення. Перша причина – це створення повідомлення на основі явної інструкції конкретного правила. А саме, комутатор передає на контролер всі пакети загорнуті у повідомлення *PacketIn,* що підпадають під правило, яке явно вказує, що наступним вузлом на шляху передавання пакету є контролер. Іншою причиною може бути відсутність правила для пакету в таблиці потоків або ж помилка *TTL*.

Повідомлення *PacketIn* складається з заголовку, після якого наводиться ідентифікатор пакету даних у буфері комутатора. Довжина пакету даних записана в полі *total_len*. Поле *in_port* містить ідентифікатор фізичного порту, через який отримано відповідний пакет. Поле причини описує, чому саме цей пакет було відправлено на контролер. Отже, частина самого пакету передається у цьому повідомленні як масив байт. Контролер також має можливість вводити пакети в площину передачі даних через будь-який комутатор у мережі. Ця процедура здійснюється за допомогою повідомлення типу *PacketOut*. Таке повідомлення може містити як сам пакет даних, так й ідентифікатор місця збереження пакету у вхідному буфері комутатора. Пакет аналізується центральним процесором, який проводить пошук можливих операцій із отриманим пакетом у списку всіх можливих операцій. Якщо знайдена операція вказує, що такий пакет потрібно обробити стандартною апаратною таблицею потоків, тоді поле *port_id* використовується як вхідний порт пакету даних.

Структура повідомлення *PacketOut* дуже подібна до структури повідомлення *PacketIn*.

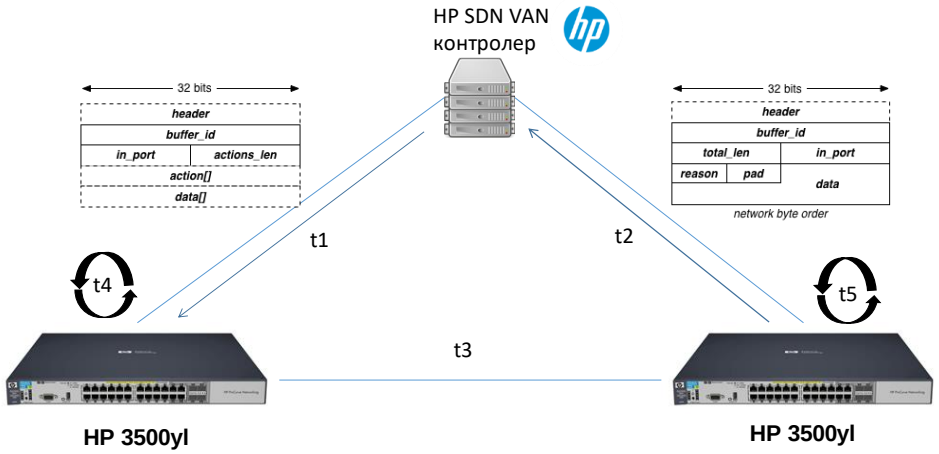


Рис. 4.5. Схема проведення експерименту для вимірювання затримки між двома комутаторами з використанням контролера

Вона також містить ідентифікатор місця збереження пакету у вхідному буфері комутатора. Якщо встановлено значення цього поля *0xffffffff*, це свідчить про те, що пакет зберігається у масиві даних *data*, який передається з цим повідомленням. Список дій, які можуть для виконуватися над пакетом, передається у масиві *action*.

Для реалізації такого експерименту в першому комутаторі встановлюється правило, яке відповідає за обробку пакету, введеного контролером у площину даних, і передачу його до наступного комутатора. Наступний комутатор містить правило, відповідно до якого такий пакет видаляється з площини передачі даних, інкапсулюється у повідомлення *PacketOut* та відправляється до контролера. Структура такого правила може бути довільною, а значення полів повинні бути такими, щоб не перекривалося жодне з уже існуючих правил в таблиці.

В експерименті вибрано правило з максимальною кількістю полів, а кожному полю задано максимально можливе значення. На першому комутаторі правило містить інструкцію «відправити пакет через вихідний порт», на другому комутаторі – інструкцію «відправити пакет на контролер». У разі відсутності інтенсивності навантаження на комутатори, канал звантажений на

0-0,001% за рахунок протоколу *LLDP*, пакети якого пересилалися з інтервалом 2 секунди. Протягом експерименту було переслано 10000 тестових пакетів. Розподіл тривалості передачі пакетів відображено у формі гістограми на рис.4.6.

Отримані результати показують, що середнє значення затримки становить 333 мкс, а стандартне відхилення рівне 159,7 мкс. Максимальне та мінімальне значення становлять 1012 та 100 мкс відповідно. Ураховуючи розподіл затримки пакетів залежно від завантаження інтерфейсу, можна стверджувати, що затримка суттєво зростає лише після проходження рівня завантаження інтерфейсу, що становить 800 Мбіт/с, а саме, збільшення завантаження інтерфейсу на 10 Мбіт/с призводить до збільшення затримки 100 мкс. Вже у разі завантаження каналу на рівні 850 Мбіт/с затримка передачі пакету перевищує 400 мкс. Відповідно до отриманих статистичних показників щодо результатів вимірювання на основі розробленого алгоритму можна зробити висновок, що вже на рівні завантаження каналу 850 Мбіт/с можна відфільтрувати затримку передачі пакету з отриманого масиву значень. Отримані результати також використовуються для оцінки затримки всього шляху, оскільки на основі них враховано тривалість обробки OpenFlow повідомлень та тривалість їх передачі по сигнальному каналу, а також тривалість обробки цих повідомлень комутатором.

Рис. 4.6. Розподіл тривалості передачі пакету в розробленому тестовому середовищі

Для порівняння ефективності запропонованого алгоритму вимірювання затримки та традиційного методу вимірювання на основі *Traceroute* було прокладено шлях у мережі між комутаторами 3 та 5. Шляхи проходження тестових пакетів для *Traceroute* (штрихпунктирна крива) та розробленого методу (штрихова крива) відображено на рис.4.7.

Вимірювання проводилося для різних рівнів завантаження шляху. При чому, в каналах циркулював мультисервісний трафік, а основна задача полягала у вимірюванні тривалості передачі даних 1-го класу трафіку, а конкретніше – потоку *VoIP*. Завантаження шляху змінювалося від 10% до 100% з кроком 10. У кожній ітерації вимірювання проводилося 1000 разів окремо з використанням розробленого методу та додатку *Traceroute*.

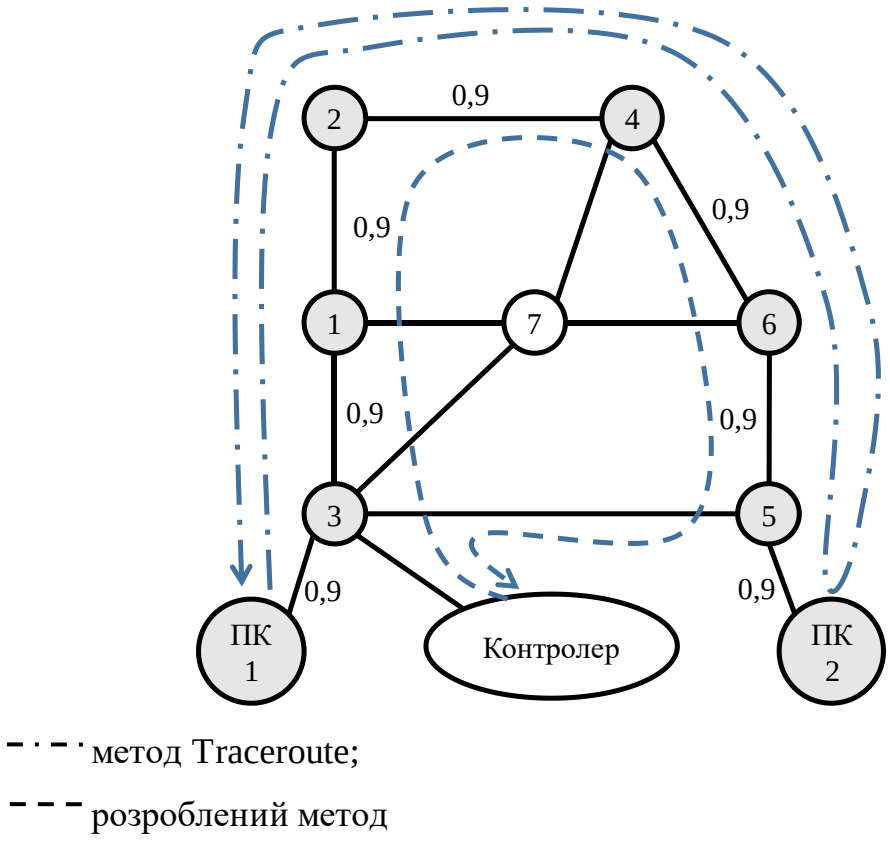


Рис. 4.7. Порівняльна схема вимірювання тривалості проходження пакету вибраним маршрутом в мережі

Порівняти результати моніторингу можна на основі діаграми, представленої на рис.4.8.

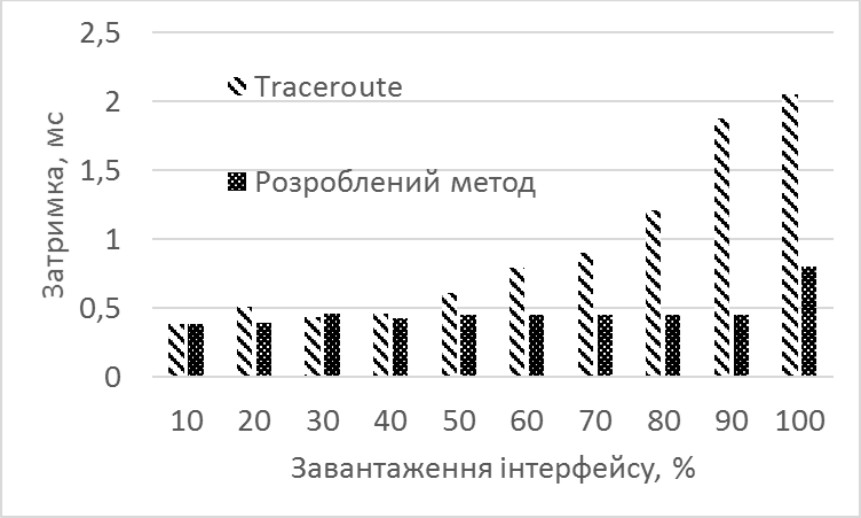


Рис. 4.8. Порівняння результатів оцінки затримки на основі *Traceroute* та розробленого алгоритму

На основі розрахунку встановлено, що тривалість передавання пакету розміром 1500 байт відповідним шляхом, що складається з 7 каналів зі швидкістю 1Гбіт/с та 6 комутаторів зі швидкістю комутації 4 мкс, становить 98 мкс. Аналізуючи отримані результати, можна стверджувати, що у випадку незавантаженого каналу (менше від 50%) затримка, виміряна стандартним методом та алгоритмом, запропонованим у роботі, є однаковою. Коли інтенсивність завантаження перевищує 50%, затримка, виміряна стандартним методом, підвищується, що пов’язано з використання протоколу TCP для реалізації вимірювань. У разі збільшення завантаження шляху тривалість передавання пакетів, а також повернення підтверджень зростає внаслідок виникнення черг.

Підсумовуючи зазначене, можна стверджувати, що у випадку вимірювання затримки шляху в каналі з максимальним завантаженням запропонований алгоритм дає змогу підвищити точність вимірювання затримки у середньому в 2,5 рази. Це, у свою чергу, дає змогу точніше оцінити затримку шляху в реальних умовах перевантаження мережі та провести оптимальний перерозподіл навантаження з врахуванням критеріїв якості обслуговування.

4.4. Дослідження параметрів якості обслуговування потоків та ефективності розподілу навантаження з використанням удосконаленої моделі маршрутизації

Для дослідження ефективності моделей маршрутизації використано розроблений метод вимірювання затримки для оцінювання затримки передавання пакетів між всіма парами вузлів у мережі тестового середовища. Результати вимірювань представлено на рис.4.9.

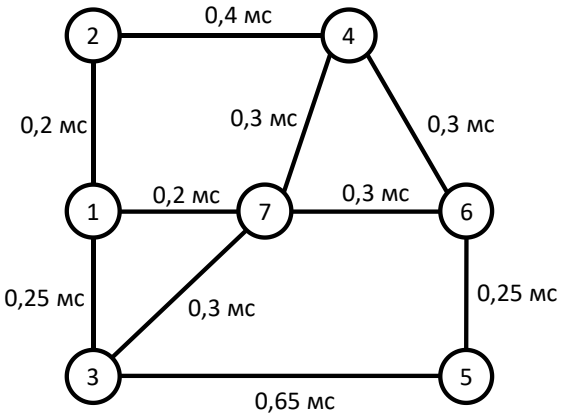


Рис. 4.9. Характеристика затримки в каналах побудованої тестової мережі

Число, що стоїть біля кожного ребра, відповідає затримці, яка враховує тривалість передачі пакету в цьому каналі, а також тривалість обробки пакету обома вузлами.

Пропускну здатність між усіма вузлами для всіх портів встановлено на рівні 100 Мбіт/с. Для проведення експерименту в мережі згенеровано фонове навантаження за допомогою системи генерації мультисервісного трафіку. Матрицю вимог щодо пропускної здатності між вузлами відображено в табл.4.3.

Таблиця 4.3

Вимоги щодо пропускної здатності між вузлами для створення фонового мультисервісного навантаження, Мбіт/с

	1	2	3	4	5	6
1	-----	12,5	16,5	5	9	7
2	11	-----	12,5	20	14	13
3	5	16,5	-----	21	12,6	5
4	10	14,5	12	-----	13	11
5	8	18	6	8	-----	15
6	12	8	9	15	5	-----

Вузол №7 є проміжним (представляє рівень агрегації), а тому до нього не під'єднано кінцеві пристрої (сервери).

Побудована мережа використовувалася студентами університету м. Вюрцбурга (Німеччина) для доступу до сервісів даних у локальному центрі обробки даних, тому усереднені вимоги щодо пропускної здатності отримано в результаті моніторингу трафіку в цій мережі впродовж двох днів.

Відповідно до матриці вимог згенеровано список потоків для всіх типів трафіку. Для смуги пропускання 100 Мбіт/с згенеровано набір клієнтів та послуг, які вони використовують. Список потоків для 9-ти клієнтів, агрегована швидкість яких становить 100 Мбіт/с, відображено в табл.**Error! Reference source not found.** (*C* – пропускна здатність, Мбіт/с; *P* – пріоритет).

Таблиця 4.4

Список клієнтських потоків, сумарна швидкість яких становить 100 Мбіт/с

Послуга		Клієнт								
		1	2	3	4	5	6	7	8	9
VoIP	C	0,17	0,19	0,16	0,19	0,18	0,18	-	-	-
	P	5	10	15	20	25	30	-	-	-
Skype	C	1,25	1,01	1,17	1,25	1,17	1,27	1,11	1,09	1,44
	P	260	265	270	275	280	285	290	295	300
IPTV	C	1,91	2,09	2,13	2,21	2,15	2,17	1,95	2,33	2,42
	P	515	520	525	530	535	540	545	550	555
Interactive	C	1,59	1,60	1,54	1,80	1,68	1,53	1,90	1,46	1,81
	P	770	775	780	785	790	795	800	805	810
VoD	C	2,06	2,22	1,71	1,84	2,15	2,13	1,97	2,16	2,14
	P	1025	1030	1035	1040	1045	1050	1055	1060	1065
Дані	C	5,56	5,57	3,99	5,50	5,72	5,72	5,50	-	-
	P	1280	1285	1290	1295	1300	1305	1310	-	-
Cloud	C	9,96	9,90	9,48	10,63	10,06	10,49	8,30	12,98	9,82
	P	1540	1545	1550	1555	1560	1565	1570	1575	1580

З табл.Таблиця 4.4 видно, що для пропускної здатності 100 Мбіт/с згенеровано 9 клієнтів. Кожен клієнт користується певним набором послуг. У тому разі, коли клітинка на перетині стовпця та рядка є порожньою, цю послугу слід вважати такою, для якої відсутнє будь-яке гарантування якості обслуговування у випадку, якщо клієнт нею скористується. Всі інші послуги

мають вимоги щодо пропускної здатності та параметрів обслуговування для конкретного клієнта, і, відповідно до його пріоритету, гарантуються сервісним договором між клієнтом та оператором мережі. Як видно з табл. Таблиця 4.4,кожному клієнту присвоєно пріоритет у відповідному діапазоні пріоритетів для конкретного типу сервісу. Такий метод генерації проведено для кожної пари серверів, що використовуються для генерації клієнтського навантаження.

Наступний крок полягав у розрахунку відносних пріоритетів потоків з урахуванням класифікації сервісів на категорії, запропонованої у другому розділі роботи.

Для першої категорії сервісів, яка визначає потоки реального часу, надзвичайно чутливі до флуктуації часових параметрів якості обслуговування, належать: голосовий дзвінок, відеодзвінок, IPTV та потокове відео.

Результати розрахунку відносних пріоритетів для всіх клієнтів згідно методики, вдосконаленої у другому розділі роботи, відображено в табл. 4.5.

Таблиця 4.5

Відносні пріоритети потоків з урахуванням пріоритету клієнта та його вимог щодо якості обслуговування

Номер клієнта	1	2	3,00	4,00	5,00	6,00	7,00	8,00	9,00
VoIP	0,25	0,22	0,22	0,21	0,21	0,21	----	----	----
Відео Skype	0,188	0,187	0,186	0,185	0,184	0,183	0,182	0,181	0,18
IPTV	0,148	0,147	0,146	0,145	0,144	0,143	0,142	0,141	0,42
Інтерактивні сервіси	0,128	0,127	0,126	0,125	0,124	0,123	0,122	0,121	0,12
VoD	0,108	0,107	0,106	0,105	0,104	0,103	0,102	0,101	0,1
Дані	0,088	0,087	0,086	0,085	0,084	0,083	0,082	----	----
Cloud	0,048	0,047	0,046	0,045	0,044	0,043	0,042	0,041	0,04

Оцінювання маршрутизації з використанням протоколу EIGRP.

Зазвичай найпоширенішими протоколами маршрутизації в традиційних IP мережах на основі стану каналу є OSPF та IS-IS. Протокол OSPF (*Open Shortest Path First*) — це протокол динамічної маршрутизації, який належить до протоколів, що обирають маршрут на основі стану каналу (завантаження чи параметрів якості обслуговування). OSPF використовує алгоритм Дейкстри для

знаходження найкоротшого шляху. IS-IS – протокол також використовує стан каналів для знаходження найкоротших шляхів на основі алгоритму Дейкстри. Протокол EIGRP (Enhanced Interior Gateway Routing Protocol) – один з найбільш поширених протоколів, що визначають маршрути на основі стану каналу, використовуючи алгоритм Diffused Update Algorithm (DUAL) для пошуку найкоротшого шляху на основі інтегральної метрики. Основними параметрами, які EIGRP використовує для обчислення метрики, є мінімальна пропускна здатність та затримка. Проте перевагою протоколу EIGRP є можливість балансування навантаження та знаходження маршруту на основі метрики, яка враховує параметри якості обслуговування і стан каналів. Тому в роботі для дослідження вибрано протокол EIGRP, який реалізований як програмний компонент контролера ПКМ [72].

Для тестування ефективності запропонованих рішень використано сервіс IPTV, що транслює відео з одинадцятого сервера. Мережу заповнено фоновим трафіком відповідно до вимог, наведених у табл.4.3, за допомогою системи генерації мультисервісного трафіку, яка встановлена на кожному сервері, що під'єднаний до кожного комутатора. Отримано гістограму завантаження мережних каналів з використанням протоколу EIGRP, яку відображено на рис.4.10.

Рис. 4.10. Гістограма завантаження мережних каналів унаслідок маршрутизації фонового навантаження за допомогою протоколу EIGRP

У результаті роботи протоколу EIGRP мережа була заповнена фоновим трафіком. Всі шляхи для всіх вимог відображено в табл.4.6.

Таблиця 4.6

Шляхи проходження фонового трафіку відповідно до протоколу EIGRP

Потік	Шлях	Вимога, Мбіт/с	Потік	Шлях	Вимога, Мбіт/с
1-2	1-2	12,5	4-1	4-7-1	10
1-3	1-3	16,5	4-2	4-2	14,5
1-4	1-7-4	5	4-3	4-7-3	12
1-5	1-7-6-5	9	4-5	4-6-5	13
1-6	1-7-6	7	4-6	4-6	11
2-1	2-1	11	5-1	5-3-1	8
2-3	2-1-3	12,5	5-2	5-6-4-2	18
2-4	2-4	20	5-3	5-3	6
2-5	2-4-6-5	14	5-4	5-6-4	8
2-6	2-1-7-6	13	5-6	5-6	15
3-1	3-1	5	6-1	6-7-1	12
3-2	3-1-2	16,5	6-2	6-4-2	12
3-4	3-7-4	21	6-3	6-7-3	9
3-5	3-5	12,6	6-4	6-4	15
3-6	3-7-6	5	6-5	6-5	5

На наступному етапі було створено навантаження від сервісу IPTV, який встановлено на одинадцятому сервері. У результаті роботи EIGRP протоколу обрано маршрути для IPTV потоків, які подано в табл.4.7.

Таблиця 4.7

Маршрутизація IPTV потоків на основі EIGRP протоколу

Пара вузлів	Вимога, Мбіт/с	Шлях	Частина навантаження, %
6-1	23	Немає вільної пропускної здатності	100
6-2	18	6-4-2	100
6-3	32	6-7-3	100
6-4	21	6-4	100
6-5	14	6-5	100

Діаграму завантаження каналів мережі після запуску сервісу IPTV відображено на рис.4.11.

Рис. 4.11. Завантаження каналів мережі після маршрутизації IPTV потоків за допомогою протоколу EIGRP

У результаті маршрутизації за допомогою протоколу EIGRP у мережі виникло нерівномірне завантаження каналів. Особливу увагу слід звернути на канали 4-6, 3-7 та 6-7. Ці канали перебувають у стані, близькому до перевантаження з високою імовірністю блокування. Крім того, потік між вузлами 6 та 1 не зміг бути прокладений у зв'язку з тим, що всі канали, які виходять від вузла 6, є перевантаженими. Навіть у випадку застосування балансування навантаження потік обсягом 23 Мбіт/с не зможе бути переданий. Для детального аналізу якості обслуговування потоків IPTV обрано канал 3-7 як найбільш близький до перевантаження. На основі аналізу мережевих потоків отримано схему основних потоків, що призводять до перевантаження каналу 3-7 (рис.4.12).

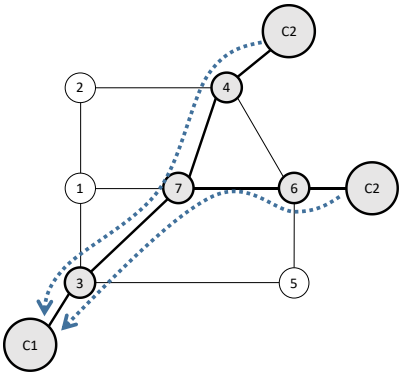


Рис. 4.12. Схема проходження основних потоків, що призводять до перевантаження каналу 3-7

Агреговану інтенсивність навантаження на канал 3-7 (комутатор №7, порт №3) відображено на рис.4.13.

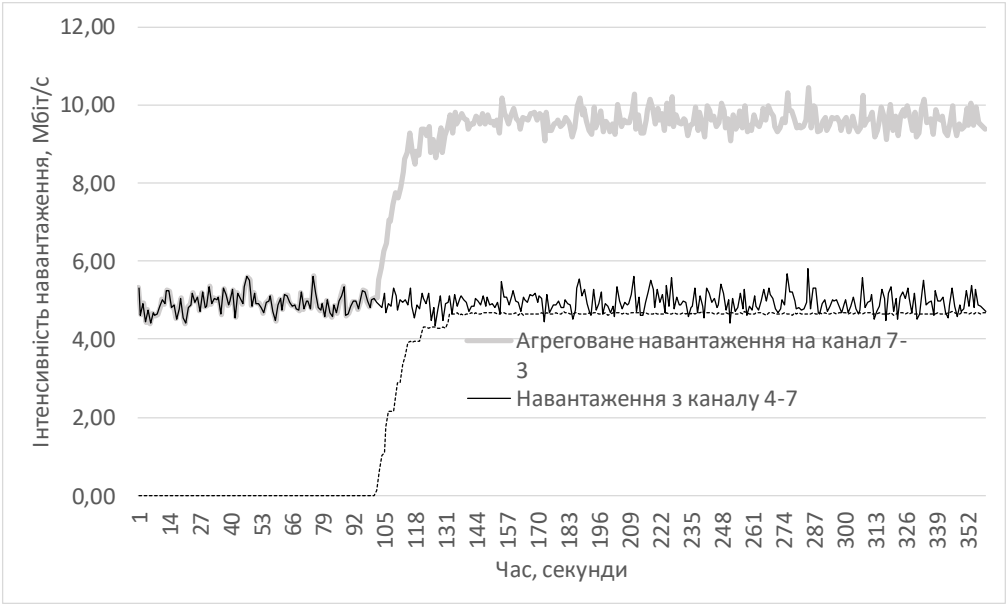
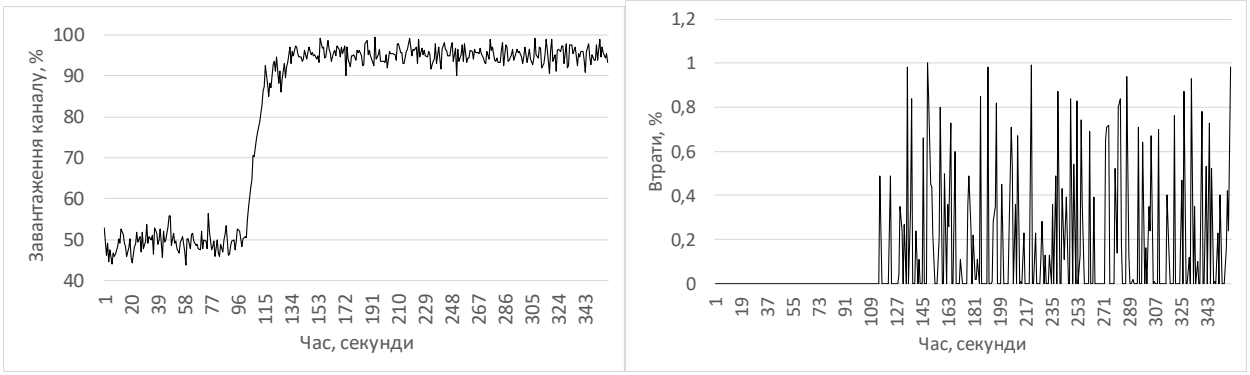


Рис. 4.13. Агрегована інтенсивність мультисервісного навантаження на канал 3-7 з каналів 6-7 та 4-7

Втрати пакетів, що виникають у каналі 7-3 внаслідок перевантаження, відображено на рис.4.14. Аналіз завантаження каналів проводиться, починаючи з найбільш завантаженого.



(а) (б)

Рис. 4.14. Завантаження (а) каналу 7-3 та втрат(б) у ньому

У результаті використання алгоритму вимірювання затримки потоків побудовано графік середніх затримок всіх потоків, що проходять каналом 3-7. Виміряні затримки відображено на рис.4.15.

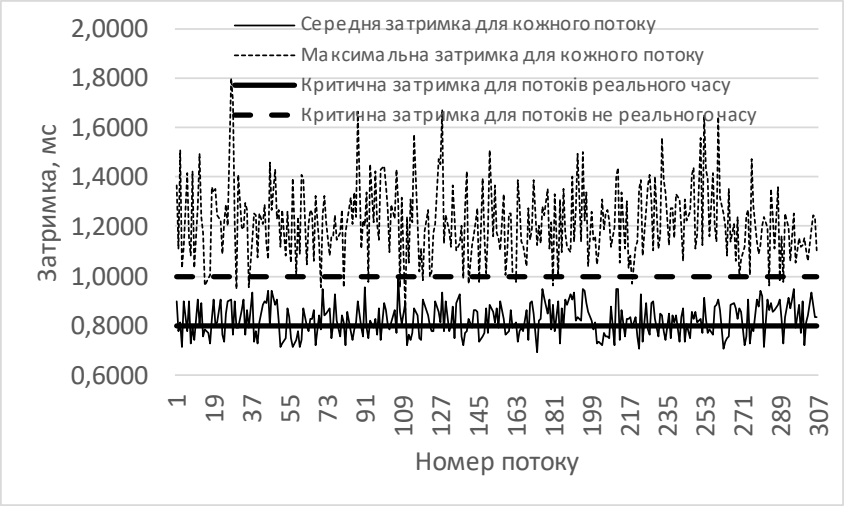


Рис. 4.15. Середні та максимальні значення затримок для всіх потоків, що проходять через канал 3-7

З отриманого графіка на рис.4.15 можна стверджувати, що всі потоки, які проходять через канал 3-7, зазнають погіршення якості обслуговування. Зокрема, критичний рівень затримки для потоків реального часу становить 0,8 мс, а для потоків класу нереального часу (друга категорія) – 1 мс. Штриховою кривою відображено максимальні значення затримки, якої зазнавали пакети кожного потоку. Для детальнішого аналізу затримок проведемо сортування потоків за значенням відносного пріоритету (рис.4.16.4.15).

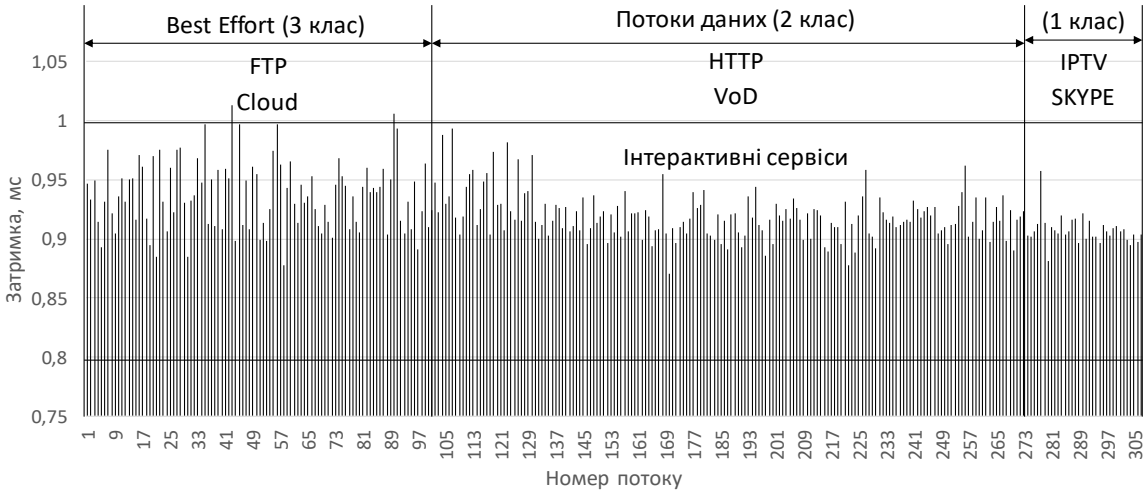


Рис. 4.16. Потоки, що проходять через канал 3-7, відсортовані за значенням відносного пріоритету

Діаграма, наведена на рис.4.17, демонструє те, що середня затримка всіх потоків реального часу (1-ий клас) перевищує критичне значення 0,8 мс.

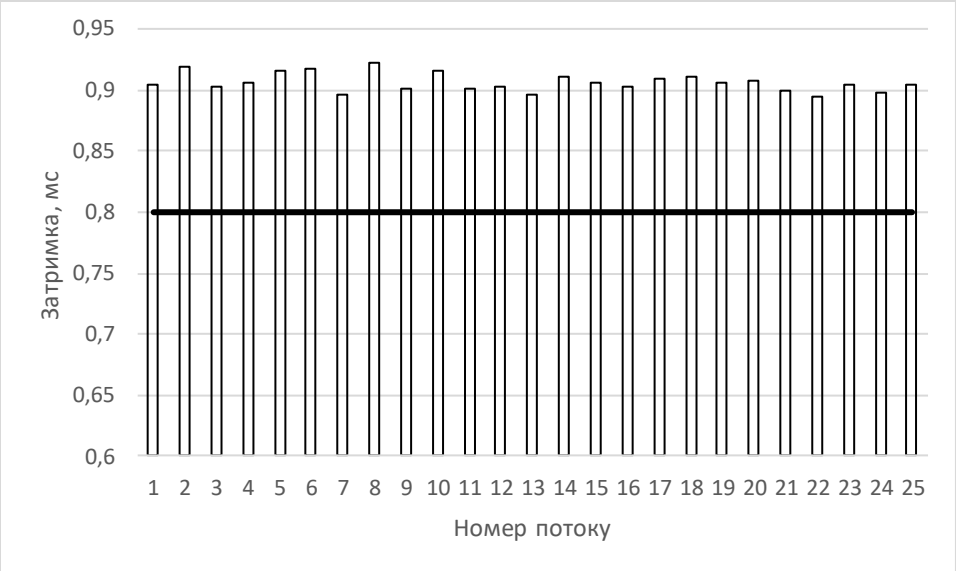


Рис. 4.17. Характеристика середньої затримки для потоків реального часу, що проходять через канал 3-7

Діаграма, відображена на рис.4.18, демонструє затримки для потоків даних, що належать до 2-го та 3-го класу.

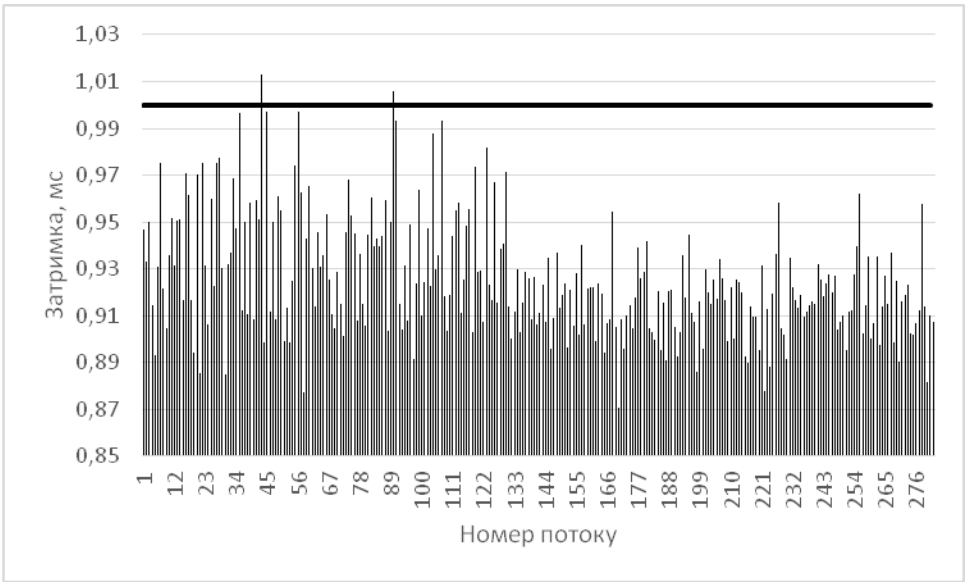


Рис. 4.18. Характеристика середньої затримки для потоків реального часу, що проходять через канал 3-7

Очевидним є те, що середня затримка для 99% потоків не перевищує критичного значення 1мс.

Для підвищення якості обслуговування потоків реального часу та уникнення перевантаження каналів 3-7, 4-7 та 6-7 на контролері увімкнено використання удосконаленої моделі маршрутизації. Алгоритм розвантаження

каналу в кожній ітерації розраховує оптимальний маршрут для потоків з найменшим відносним пріоритетом, а в кожній наступній ітерації потоки вибираються за зростанням їхнього пріоритету. Розподіл завантаження шляхів у мережі відображено на рис.4.19.

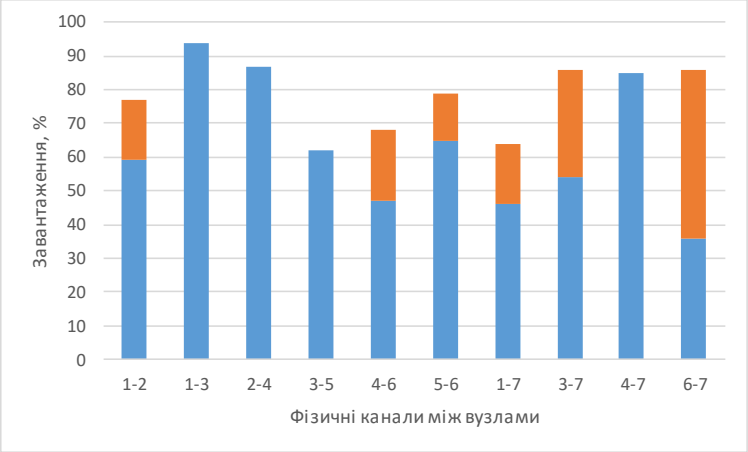


Рис. 4.19. Розподіл завантаження мережевих каналів після використання удосконаленої моделі маршрутизації

У результаті аналізу потоків встановлено, що основними потоками, за допомогою яких вдалося розвантажити мережу, є потоки, відображені схемою на рис.4.20.

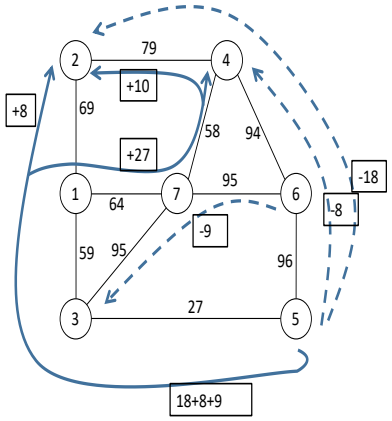


Рис. 4.20. Схема перерозподілу навантаження відповідно до результатів маршрутизації з використанням удосконаленої моделі

Порівняння середнього значення та стандартного відхилення для значень завантаження мережевих каналів показало, що на першому етапі (з використанням лише протоколу EIGRP) середнє завантаження каналу становило 73%, а стандартне відхилення – 22,73%. Після застосування удосконаленої моделі маршрутизації середнє завантаження каналів зросло до

79%, стандартне відхилення зменшилося до 10,86%, а коефіцієнт варіації знизився з 0,31 до 0,14.

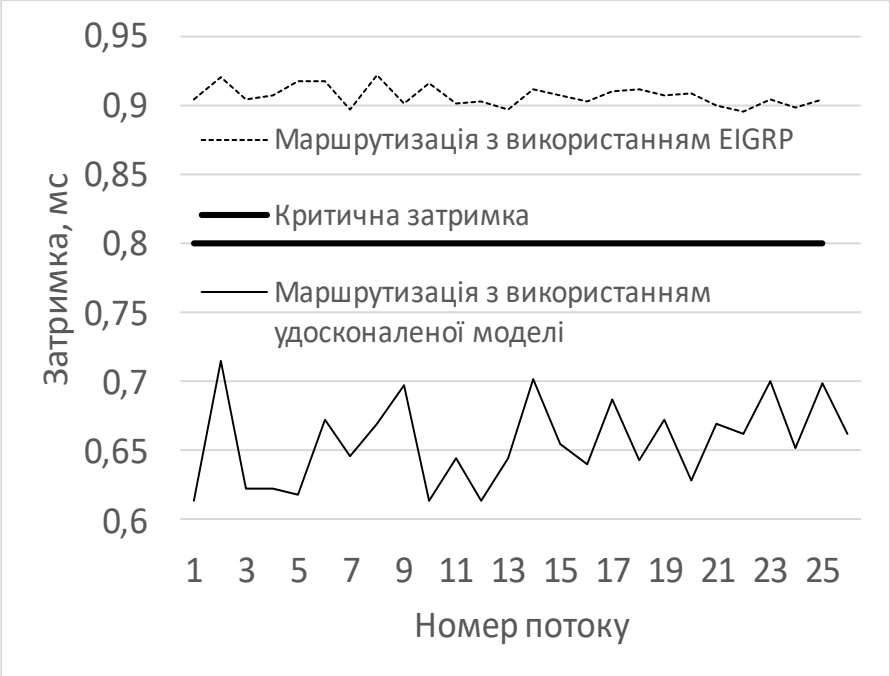


Рис. 4.21. Характеристика затримки потоків у мережі після перерозподілу потоків для рівномірного завантаження каналів

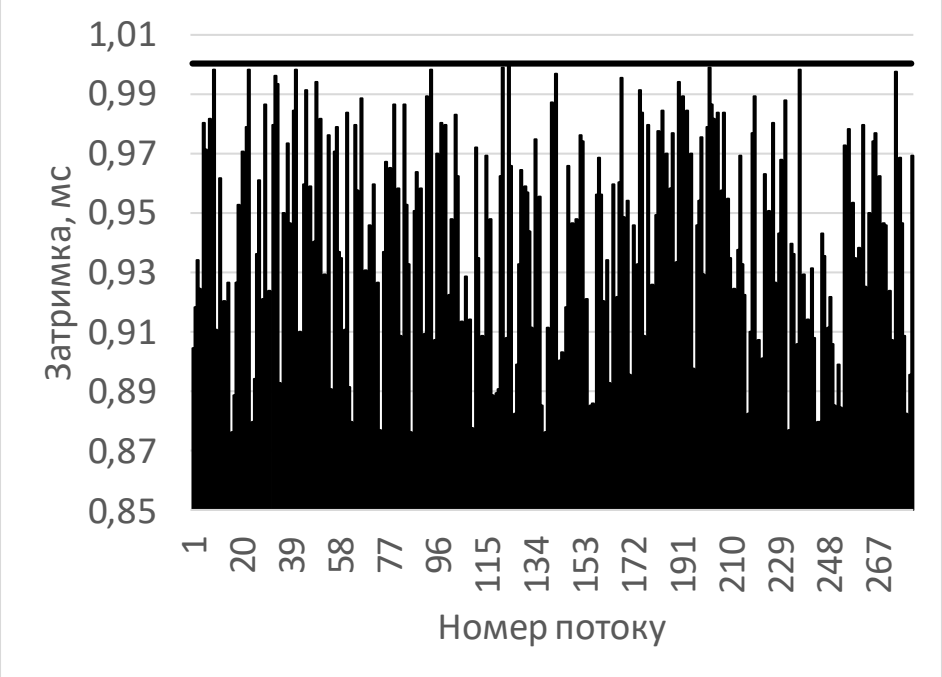


Рис. 4.22. Характеристика середньої затримки пакетів для потоків у каналі 1-3 після застосування удосконаленого методу маршрутизації

У процесі дослідження продемонстровано, що удосконалена модель маршрутизації дає змогу досягнути більш рівномірного розподілу

навантаження в мережі порівняно з протоколом EIGRP та забезпечити необхідну якість обслуговування для всіх потоків.

У результаті проведення описаних вище експериментів доведено, що комплексне застосування запропонованих у роботі наукових рішень дало змогу покращити затримку для потоків реального часу на 15% (див. рис.4.22) та рівномірність завантаження мережевих ресурсів на 30%.

Висновки до 4-го розділу

У четвертому розділі описано конфігурацію тестового середовища, сформованого на основі апаратних OpenFlow комутаторів *HP3500yl*. Проведено аналіз способів інтеграції контролерів у випадку розподіленого керування мережею, а також особливостей інтеграції сторонніх модулів у систему управління SDN. Подано рекомендації щодо інтеграції розробленої системи моніторингу для забезпечення оптимального функціонування мережі в різних сценаріях.

Проведено дослідження ефективності методу вимірювання затримки на основі реальної топології, який, порівняно з існуючими методами вимірювання затримки, дає змогу підвищити точність оцінки затримки до 2,5 разів залежно від завантаження шляху.

Оцінено ефективність використання удосконаленого методу маршрутизації потоків на основі їхніх пріоритетів порівняно з протоколом EIGRP. Продемонстровано, що розроблений метод дає змогу досягнути більш рівномірного розподілу навантаження в мережі з гарантованою якістю обслуговування для всіх потоків. Загалом, у результаті експерименту вдалося встановити, що запропонований метод дав змогу покращити затримку на 10% для потоків реального часу, звести втрати до нуля та підвищити рівномірність завантаження каналів на 30%.

ОСНОВНІ РЕЗУЛЬТАТИ ТА ВИСНОВКИ

У дисертаційній роботі розв'язано наукове завдання розроблення методів та моделей управління процесами передавання даних у телекомунікаційних ПКМ для підвищення якості обслуговування користувачів та ефективного використання мережевих ресурсів.

Основні результати роботи такі:

Проведено аналіз основних моделей функціонування телекомунікаційних ПКМ. Встановлено, що існуючі моделі управління трафіком не забезпечують диференціювання окремих потоків користувачів. Система управління ПКМ не гарантує виконання вимог до якості обслуговування окремих потоків, оскільки оперує неточною інформацією про стан мережевих ресурсів та не здатна оптимально обрати маршрут передавання даних. Оптимізація процесу передавання даних у ПКМ здійснюється або за критерієм якості обслуговування трафіку, або за критерієм рівномірності розподілу ресурсів, що у першому випадку призводить до перевантаження основних маршрутів, а в другому – до низької якості обслуговування потоків реального часу.

Запропоновано спосіб ідентифікації потоку даних певного класу окремого клієнта на основі визначення відносного пріоритету. Для розрахунку відносного пріоритету введено два додаткових параметри: чутливість до зміни порядку пакетів та пріоритет клієнта в межах одного класу трафіку. Область значень відносних пріоритетів перебуває в межах від нуля до одиниці та може бути розділена на окремі підобласті з метою вибору індивідуальної політики керування відповідними потоками. Перевагою запропонованого способу є можливість однозначно ідентифікувати потік та вимоги щодо якості його обслуговування.

Розроблено систему моніторингу структурно-функціональних параметрів ПКМ, яка надає засоби для збору, обробки та представлення інформації для гетерогенних апаратних та програмних OpenFlow комутаторів. Система використовує набір різнорідних каналів доступу до параметрів мережі,

включаючи програмні інтерфейси контролера та безпосередній доступ до закритих апаратних параметрів комутатора через віддалений термінал або спеціально виділений OpenFlow канал. Система надає змогу розширювати функціональність моніторингу та впроваджувати нові рішення на основі використання узагальнених моделей зберігання параметрів комутаторів та процесів передавання даних.

Вперше запропоновано модель адаптації системи моніторингу, яка дає змогу підвищити точність оцінки статистичних характеристик використання мережевих ресурсів та ймовірності блокування певного елементу мережі. На основі експерименту з апаратними OpenFlow комутаторами встановлено, що запропонована модель дає змогу підвищити точність оцінки завантаження мережевих каналів не менше, ніж на 10% залежно від статистичних характеристик мультисервісного трафіку.

Набув подальшого розвитку метод вимірювання затримки передавання пакетів окремого потоку, який, порівняно з існуючими методами, дає змогу підвищити точність оцінки затримки у 2,5 рази, якщо завантаження цього шляху наближається до максимального значення. Метод використано в розробленій системі моніторингу.

Удосконалено потокову модель маршрутизації з використанням відносного пріоритету потоку, яка дає змогу підвищити якість обслуговування потоків реального часу та забезпечити необхідну якість обслуговування потоків передавання даних, а також досягнути рівномірного завантаження мережних каналів, що знижує імовірність виникнення втрат внаслідок різких перепадів інтенсивності навантаження мультисервісного трафіку.

Зменшено втрати пакетів у мережі у 6 разів за рахунок використання удосконаленої моделі балансування навантаження, яка, на відміну від базової моделі балансування OpenFlow, враховує завантаження наступного каналу та відносний пріоритет потоку.

Зменшено середню затримку пакетів потоків реального часу у мережі на 15% та підвищено рівномірність завантаження мережних каналів на 30% у

випадку використання удосконаленої потокової моделі маршрутизації порівняно з протоколом EIGRP.

Запропоновані в роботі моделі та алгоритми впроваджено у розроблену систему моніторингу як програмні модулі. Цю систему використано для проведення експериментів на апаратній програмно-конфігурованій мережі, в результаті яких отримано значення всіх наведених у роботі кількісних показників та перевірено ефективність розроблених рішень.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. McKeown N., et. al. OpenFlow: Enabling Innovation in Campus Networks [Electronic resourse] / N.McKeown, et. al. // Newsletter ACM SIGCOMM Computer Communication. – USA, NY, 2008. – Vol.38. – Issue 2. – P.69-74. – Mode of access: http://dl.acm.org/citation.cfm?id=1355746.

2. Software-Defined Networking: The New Norm for Networks [Electronic resourse] // ONF White Paper. – Palo Alto, CA, USA, 2012. – Mode of access: https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf.

3. OpenFlow Switch Specification [Electronic resourse]// ONF. – 2014. – Mode of access: https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-switch-v1.5.0.noipr.pdf.

4. Gude N., et. al. NOX: Towards an Operating System for Networks [Electronic resourse] / N.Gude, et. al. // ACM SIGCOMM Computer Communication Review. – 2008. – Vol.38, No.3 – P.105–110. – Mode of access: http://benpfaff.org/papers/nox.pdf.

5. About POX [Electronic resourse]. –
Mode of access: http://www.noxrepo.org/pox/about-pox/ (Accessed 1 June 2016).

6. Beacon [Electronic resourse]. –
Mode of access: https://openflow.stanford.edu/display/Beacon (Accessed 1 June 2016).

7. Floodlight [Electronic resourse]. – Mode of access:
http://Floodlight.openflowhub.org/ (Accessed 1 June 2016).

8. Ryu [Electronic resourse]. – Mode of access: http://osrg.github.com/ryu/ (Accessed 1 June 2016).

9. Trema [Electronic resourse]. – Mode of access: http://trema.github.com/trema/ (Accessed 1 June 2016).

10. Chowdhury N.M.M.K. A survey of network virtualization [Electronic resourse] / N.M.M.K.Chowdhury, R.Boutaba // Computer Networks. – 2010. – Vol.54. – P.862–

876. Mode of access: http://nsm1.cs.uwaterloo.ca/rboutaba/Papers/Journals/2010/Mosharaf10.pdf.

11. Sherwood R., et. al. Carving research slices out of your production networks with OpenFlow [Electronic resourse] / R.Sherwood, et. al. // ACM SIGCOMM Computer Communication Review. – 2010. – Vol.40, No.1. – P.129. – Mode of access: http://conferences.sigcomm.org/sigcomm/2009/demos/sigcomm-pd-2009-final65.pdf.

12. Khan A., Zugenmaier A., Jurca D., and Kellerer W. Network virtualization: a hypervisor for the Internet? [Electronic resourse] / A.Khan, A.Zugenmaier, D.Jurca, W.Kellerer // IEEE Communications Magazine. – 2012. – Vol.50, No.1. – P.136–143. – Mode of access: http://ieeexplore.ieee.org/document/6122544/.

13. Phemius K., Bouet M. Monitoring latency with OpenFlow [Electronic resourse] / K.Phemius, M.Bouet // Proceedings of the 9th International Conference on Network and Service Management (CNSM 2013) (Switzerland, Zurich, October 14-18, 2013). – P.122- 125. –

Mode of access: https://www.researchgate.net/publication/271556605_Monitoring_latency_with_OpenFlow.

14. Handigol N., Heller B., Jeyakumar V., Maziéres D., McKeown N. Where is the debugger for my software-defined network? [Electronic resourse] / N.Handigol, B.Heller, V.Jeyakumar, D.Maziéres, N.McKeown // HotSDN '12 Proceedings of the first workshop on Hot topics in software defined networks (USA, New York, August 13, 2012). – P.55-60. – Mode of access: http://dl.acm.org/citation.cfm?id=2342453.

15. Sherwood R., Yiakoumis Y. Oftrace: An OpenFlow Debugging and Analysis Tool [Electronic resourse] / R.Sherwood, Y.Yiakoumis // Stanford Clean Slate Lab, 2009. – Mode of access: archive.openflow.org/wk/images/f/fb/Oftrace.pdf (viewed on June 10, 2016).

16. Rotsos C., Antichi G., Bruyere M., Owezarski P., Moore A.W. OFLOPS-Turbo: Testing the next-generation OpenFlow switch [Electronic resourse] / C.Rotsos, G.Antichi, M.Bruyere, P.Owezarski, A.W.Moore // IEEE International Conference on Communications (ICC) (UK, London, June 8-12, 2015). – P.5571 – 5576. – Mode of access: http://www.cl.cam.ac.uk/~awm22/publications/rotsos2014oflops-turbo.pdf.

17. Sherwood R., Yap K.-K. Cbench (controller benchmarker) [Electronic resourse] / R.Sherwood, K.- K.Yap. –
Mode of access: http://www.openflowswitch.org/wk/index.php/Oflops.
18. Isolani P.H., Wickboldt G.A., Both C.B., Rochol J., Granville L.Z. Interactive monitoring, visualization, and configuration of OpenFlow-based SDN [Electronic resourse] / P.H.Isolani, J.A.Wickboldt, C.B.Both, J.Rochol, L.Z.Granville // IFIP/IEEE International Symposium on Integrated Network Management (IM) (Canada, Ottawa, May 11- 15, 2015). – P.207- 215. –
Mode of access: https://www.ietf.org/proceedings/93/slides/slides-93-nmrg-2.pdf.
19. Grover N., Agarwal N., Kataoka K. LiteFlow: Lightweight and distributed flow monitoring platform for SDN [Electronic resourse] / N.Grover, N.Agarwal, K.Kataoka // 1st IEEE Conference (NetSoft 2015) (UK, London, April 13-17, 2015). – Mode of access: http://toc.proceedings.com/26340webtoc.pdf.
20. Mate A., Trujillo J., Koci E., Zoumpatianos K., Mylopoulos J. Monitoring Strategic Business Goals with Argus [Electronic resourse] / A.Mate, J.Trujillo, E/Koci, K.Zoumpatianos, J.Mylopoulos // IEEE 19th International Enterprise Distributed Object Computing Conference (EDOC) (Australia, Adelaide, September 21-25, 2015). – P.1-8. – Mode of access: http://toc.proceedings.com/28285webtoc.pdf.
21. Case J.D., Davin J.R., Fedor M.S., Schoffstall M.L. Internet network management using the simple network management protocol [Electronic resourse] / J.D.Case, J.R.Davin, M.S.Fedor, M.L.Schoffstall // Proceedings 14th Conference on Local Computer Networks (USA, Mineapolis, October 10-12, 1989). – P.156-159. – Mode of access: http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=256.
22. Weiwei Z., Jian G., Wenjie G., Shaomin C. NetFlow-based network traffic monitoring [Electronic resourse] / Z.Weiwei, G.Jian, G.Wenjie, C.Shaomin // 13th Asia-Pacific Network Operations and Management Symposium (APNOMS) (Taiwan, Taipei, September 21-23, 2011). – P.1-4. – Mode of access: http://www.jslab6.edu.cn/media/jslab/paper/2D/2DIcFtmT5jUElX6nufrHvRopkzOAsxdi.pdf.

23. Schmidt R. de O., Sadre R., Sperotto A., Pras A. Lightweight link dimensioning using sFlow sampling [Electronic resourse] / R. de O. Schmidt, R.Sadre, A.Sperotto, A.Pras // Proceedings of the 9th International Conference on Network and Service Management (CNSM 2013) (Switzerland, Zurich, October 14-18, 2013). – P.152-155. – Mode of access: http://doc.utwente.nl/87216/1/main.pdf.

24. Fraleigh C., Moon S., Lyles B., Cotton C., Khan M., Moll D., et al. Packet-Level Traffic Measurements from the Sprint IP Backbone) [Electronic resourse] / C.Fraleigh, S.Moon, B.Lyles, C.Cotton, M.Khan, D.Moll, et al. – Mode of access: http://www.ece.ucdavis.edu/~chuah/classes/EEC274/refs/FML03-ipmon.pdf.

25. How to Accurately Detect and Correct Packet Loss [Electronic resourse]. – Silver Peak. – Mode of access: http://www.silver-peak.com/info-center/how-accurately-detect-and-correct-packet-loss.

26. Heller B., Seetharaman S., Mahadevan P., Yiakoumis Y., Sharma P., Banerjee S., et al. ElasticTree: Saving energy in data center networks [Electronic resourse] / B.Heller, S.Seetharaman, P.Mahadevan, Y.Yiakoumis, P.Sharma, S.Banerjee, et al. // Proceedings of the 7th USENIX conference on Networked systems design and implementation, NSDI'10 (USA, CA, Berkeley, 2010). – Berkeley: USENIX Association, 2010. –
Mode of access: https://www.usenix.org/legacy/event/nsdi10/tech/full_papers/heller.pdf.

27. Tootoonchian A., Ghobadi M., Ganjali Y. OpenTM: Traffic matrix estimator for OpenFlow networks [Electronic resourse] / A.Tootoonchian, M.Ghobadi, Y.Ganjali // Proceedings of the 11th international conference on Passive and active measurement, PAM'10 (Springer-Verlag, Berlin, Heidelberg, 2010). – P.201-210. – Mode of access: http://www.pam2010.ethz.ch/papers/full-length/21.pdf.

28. Ballard Jeffrey R., Rae I., Akella A. Extensible and scalable network monitoring using OpenSAFE [Electronic resourse] / Jeffrey R. Ballard, Ian.Rae, A.Akella // Proceedings of the 2010 internet network management conference on Research on enterprise networking, INM/WREN'10 (USA, CA, Berkeley, 2010). – Berkeley:

USENIX Association, 2010. – P.8. –
Mode of access: http://pages.cs.wisc.edu/~ballard/papers/opensafe-inmwren.pdf.

29. Yu C., Lumezanu C., Zhang Y., Singh V., Jiang G., Madhyastha H.V. FlowSense: Monitoring network utilization with zero measurement cost [Electronic resourse] / C.Yu, C.Lumezanu, Y.Zhang, V.Singh, G.Jiang, H.V.Madhyastha // Proceedings of the 14th international conference on Passive and Active Measurement, PAM'13 (Springer-Verlag, Berlin, Heidelberg, 2013). – P.31-41. – Mode of access: http://alumni.cs.ucr.edu/~cyu/papers/pam13.pdf.

30. Yu M., Jose L., Miao R. Software defined traffic measurement with OpenSketch [Electronic resourse] / M.Yu, L.Jose, R.Miao // Proceedings of the 10th USENIX conference on Networked Systems Design and Implementation, nsdi'13 (USA, CA, Berkeley, 2013). – Berkeley: USENIX Association, 2013. – P.29-42. – Mode of access: http://web.stanford.edu/~lavanyaj/papers/opensketch12.pdf.

31. Kim H., Feamster N. Improving network management with software defined networking [Electronic resourse] / H.Kim, N.Feamster // IEEE Communications Magazine. – 2013. – Vol.51, No.2. – P.114-119. – Mode of access: https://users.ece.cmu.edu/~vsekar/Teaching/Fall14/18859K/papers/procera.pdf.

32. Yu Y., Qian C., Li X. Distributed and collaborative traffic monitoring in software defined networks [Electronic resourse] / Y.Yu, C.Qian, X.Li // Proceedings of the Third Workshop on Hot Topics in Software Defined Networking, ser. HotSDN '14 (USA, NY, New York, 2014). – P.85-90. – Mode of access: http://conferences.sigcomm.org/sigcomm/2014/doc/slides/197.pdf.

33. Shirali-Shahreza S., Ganjali Y. FleXam: flexible sampling extension for monitoring and security applications in OpenFlow [Electronic resourse] / S.Shirali-Shahreza, Y.Ganjali // Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking, ser. HotSDN '13 (USA, NY, New York, 2013). – P.167-168.
Mode of access: https://pdfs.semanticscholar.org/82a8/2fd74ef0cb078c2444efcca072d673821893.pdf.

34. Adrichem van N.L.M., Doerr C., Kuipers F.A. OpenNetMon: Network monitoring in openflow software-defined networks [Electronic resourse] / N.L.M. van Adrichem, C.Doerr, F.A.Kuipers // IEEE Network Operations and Management Symposium, NOMS 2014 (Poland, Krakow, May 5-9, 2014). – P.1-8. – Mode of access: https://www.researchgate.net/publication/271473174_OpenNetMon_Network_monitoring_in_OpenFlow_Software-Defined_Networks.

35. Suh J., Kwon T., Dixon C., Felter W., Carter J. OpenSample: A low-latency, sampling-based measurement platform for commodity SDN [Electronic resourse] / J.Suh, T.Kwon, C.Dixon, W.Felter, J.Carter // Distributed Computing Systems (ICDCS), IEEE 34th International Conference (June 30 – July 3, 2014). – P.228-237. Mode of access: https://www.computer.org/csdl/proceedings/icdcs/2014/5169/00/5169a228-abs.html.

36. Argyropoulos C., Kalogeras D., Androulidakis G., Maglaris V. PaFloMon – a slice aware passive flow monitoring framework for OpenFlow enabled experimental facilities [Electronic resourse] / C.Argyropoulos, D.Kalogeras, G.Androulidakis, V.Maglaris // Software Defined Networking (EWSDN). – European Workshop, 2012. – P.97- 102. –

Mode of access: https://www.ewsdn.eu/files/Presentations/EWSDN%202012/4_2_PaFloMon.pdf.

37. Georgopoulos P., Elkhatib Y., Broadbent M. et al. Towards networkwide QoE fairness using OpenFlow-assisted adaptive video streaming [Electronic resourse] / P.Georgopoulos, Y.Elkhatib, M.Broadbent et al. // Proceedings of the 2013 ACM SIGCOMM Workshop on Future Human Centric Multimedia Networking (FhMN 2013) (China, Hong Kong, 2013). – P.15–20. – Mode of access: http://conferences.sigcomm.org/sigcomm/2013/papers/fhmn/p15.pdf.

38. Zinner T., Jarschel M., Blenk A. et al. Dynamic application-aware resource management using software-defined networking: implementation prospects and challenges [Electronic resourse] / T.Zinner, M.Jarschel, A.Blenk et al. // Proceedings of the 2014 IEEE Network Operations and Management Symposium (NOMS'14) (Poland, Krakow, 2014). – P.1- 6. –

Mode of access: https://www.researchgate.net/publication/271417879_Dynamic_application- aware_resource_management_using_Software-Defined_Networking_Implementation_prospects_and_challenges.

39. Mann V., Vishnoi A., Iyer A. et al. VMPatrol: dynamic and automated QoS for virtual machine migrations [Electronic resourse] / V.Mann, A.Vishnoi, A.Iyer et al. // Proceedings of the 8th International Conference on Network and Service Management (CNSM) (USA, Las Vegas, 2012). – P.174-178. – Mode of access: http://ieeexplore.ieee.org/document/6380009/.

40. Kim W., Sharma P., Lee J. et al. Automated and scalable QoS control for network convergence [Electronic resourse] / W.Kim, P.Sharma, J.Lee et al. // Proceedings of the 2010 Internet Network Management Conference on Research on Enterprise Networking (INM/WREN'10) (Canada, San Jose, 2010). – Mode of access: http://shiftleft.com/mirrors/www.hpl.hp.com/personal/Sung-Ju_Lee/abstracts/papers/inm-wren2010.pdf.

41. Nguyen-Ngoc A., Lange S., Gebert S. et al. Investigating isolation between virtual networks in case of congestion for a Pronto 3290 switch [Electronic resourse] / A.Nguyen-Ngoc, S.Lange, S.Gebert et al. // Proceedings of the Workshop on Software-Defined Networking and Network Function Virtualization for Flexible Network Management (SDNFlex 2015) (Germany, Cottbus, 2015). – Mode of access: http://ieeexplore.ieee.org/document/7089061/.

42. Mohan P.M., Divakaran D.M., Gurusamy M. Performance study of TCP flows with QoS-supported OpenFlow in data center networks [Electronic resourse] / P.M.Mohan, D.M.Divakaran, M.Gurusamy // Proceedings of the 19th IEEE International Conference on Networks (ICON) (Singapore, 2013). – Mode of access: https://www.researchgate.net/publication/259010839_Performance_study_of_TCP_flows_with_QoS-supported_OpenFlow_in_Data_Center_networks.

43. Egilmez H., Tekalp M. Distributed QoS architectures for multimedia streaming over software defined networks [Electronic resourse] / H.Egilmez, M.Tekalp // IEEE Trans. on Multimedia (October 2014). – Vol.16, No.6. – P.1597-1609. – Mode of

access: https://www.researchgate.net/publication/265689502_Distributed_QoS_Architectures_for_Multimedia_Streaming_Over_Software_Defined_Networks.

44. Yu T.F., Wang K.C., Hsu Y.H. Adaptive routing for video streaming with QoS support over SDN networks [Electronic resourse] / T.F.Yu, K.C.Wang, Y.H.Hsu // Proceedings IEEE International Conference on Information Networking (Cambodia, January. 2015). – P.318- 323. –

Mode of access: https://www.computer.org/csdl/proceedings/icoin/2015/8342/00/07057904.pdf.

45. Akella A.V., Xiong K. Quality of Service (QoS) guaranteed network resource allocation via Software Defined Networking (SDN) [Electronic resourse] / A.V.Akella, K.Xiong // Proceedings IEEE 12th International Conference on Dependable, Autonomic and Secure Computing (Dalian, August, 2014). – P.7-13. – Mode of access: http://dl.acm.org/citation.cfm?id=2680339.

46. Egilmez H., Gorkemli B., Tekalp A., Civanlar S. Scalable video streaming over OpenFlow networks: An optimization framework for QoS routing [Electronic resourse] / H.Egilmez, B.Gorkemli, A.Tekalp, S.Civanlar // Image Processing (ICIP), 18th IEEE International Conference (September, 2011). – P.2241-2244. – Mode of access: http://ieeexplore.ieee.org/document/6116083/.

47. Egilmez H.E., Dane S.T., Bagci K.T., Tekalp A.M. OpenQoS: An OpenFlow controller design for multimedia delivery with end-to-end Quality of Service over Software-Defined Networks [Electronic resourse] / H.E.Egilmez, S.T.Dane, K.T.Bagci, A.M.Tekalp // Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC) (Asia-Pacific, December, 2012). – P.1-8. – Mode of access: http://www.ieee.org/conferences_events/conferences/conferencedetails/index.html?Conf_ID=20988.

48. Egilmez H., Civanlar S., Tekalp A. An Optimization Framework for QoS-Enabled Adaptive Video Streaming Over OpenFlow Networks [Electronic resourse] / H.Egilmez, S.Civanlar, A.Tekalp // IEEE Transactions on Multimedia. – 2013. – Vol.15. – Issue 3. – P.710- 715. –

Mode of access: http://ieeexplore.ieee.org/document/6376227/.

49. Civanlar S., Parlakisik M., Tekalp A., Gorkemli B., Kaytaz B., Onem E. A QoS-enabled OpenFlow environment for Scalable Video streaming [Electronic resourse] / S.Civanlar, M.Parlakisik, A.Tekalp, B.Gorkemli, B.Kaytaz, E.Onem // GLOBECOM Workshops (GC Wkshps), IEEE, 2010. – P.351-356. – Mode of access: https://www.researchgate.net/publication/224214589_A_QoS-Enabled_OpenFlow_Environment_for_Scalable_Video_Streaming.

50. Georgopoulos P., Elkhatib Y., Broadbent M., Mu M., Race N. Towards Network-wide QoE Fairness Using Openflow-assisted Adaptive Video Streaming [Electronic resourse] / P.Georgopoulos, Y.Elkhatib, M.Broadbent, M.Mu, N.Race // Proceedings of the 2013 ACM SIGCOMM Workshop on Future Human-centric Multimedia Networking, FhMN '13 (USA, New York, NY). – ACM, 2013. – P.15-20. – Mode of access: http://conferences.sigcomm.org/sigcomm/2013/papers/fhmn/p15.pdf.

51. Liu H., Hu Y., Shou G., Guo Z. Software Defined Networking for HTTP video quality optimization [Electronic resourse] / H.Liu, Y.Hu, G.Shou, Z.Guo // Communication Technology (ICCT), 15th IEEE International Conference (November, 2013). – P.413- 417. – Mode of access: http://ieeexplore.ieee.org/document/6820411/authors.

52. Ishimori A., Farias F., Cerqueira E., Abelem A. Control of Multiple Packet Schedulers for Improving QoS on OpenFlow/SDN Networking [Electronic resourse] / A.Ishimori, F.Farias, E.Cerqueira, A.Abelem // Software Defined Networks (EWSDN), Second European Workshop (October, 2013). – P.81-86. – Mode of access: http://ieeexplore.ieee.org/document/6680563/.

53. Hong C.-Y., Kandula S., Mahajan R., Zhang M., Gill V., Nanduri M., Wattenhofer R. Achieving High Utilization with Software-driven WAN [Electronic resourse] / C.-Y.Hong, S.Kandula, R.Mahajan, M.Zhang, V.Gill, M.Nanduri, R.Wattenhofer // Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM, SIGCOMM '13 (USA, New York, NY). – ACM, 2013. – P.15-26. – Mode of access: http://research.microsoft.com/en-us/um/people/srikanth/data/swan_sigcomm13.pdf.

54. Heller B., Sherwood R., McKeown N. The controller placement problem WAN [Electronic resourse] / B.Heller, R.Sherwood, N.McKeown // SIGCOMM Computer Communication Review (September, 2012). – Vol.42 (4). – P.473-478. – Mode of access: http://yuba.stanford.edu/~nickm/papers/hot21-heller.pdf.

55. Tootoonchian A., Ganjali Y. HyperFlow: a distributed control plane for OpenFlow [Electronic resourse] / A.Tootoonchian, Y.Ganjali // Proceedings of the 2010 internet network management conference on Research on enterprise networking, INM/WREN'10 (USA, CA, Berkeley, 2010). – USENIX Association, 2010. – P.3. – Mode of access: https://pdfs.semanticscholar.org/f7bd/dc08b9d9e2993b363972b89e08e67dd8518b.pdf.

56. Gude N., Koponen T., Pettit J., Pfaff B., Casado M., McKeown N., et al. NOX: towards an operating system for networks [Electronic resourse] / N.Gude, T.Koponen, J.Pettit, B.Pfaff, M.Casado, N.McKeown, et al. // Computer Communication Review (July, 2008). – Vol.38 (3). – P.105-110. – Mode of access: http://benpfaff.org/papers/nox.pdf.

57. Koponen T., Casado M., Gude N., Stribling J., Poutievski L., Zhu M., et al. Onix: a distributed control platform for large-scale production networks [Electronic resourse] / T.Koponen, M.Casado, N.Gude, J.Stribling, L.Poutievski, M.Zhu, et al. // Proceedings of the 9th USENIX conference on Operating systems design and implementation, OSDI'10 (USA, CA, Berkeley, 2010). – USENIX Association, 2010. – P.1-6. – Mode of access: http://www.icsi.berkeley.edu/pubs/networking/onix10.pdf.

58. Cai Z., Cox A.L. Eugene Ng T.S. Maestro: A System for Scalable OpenFlow Control [Electronic resourse] / Z.Cai, T.S. Eugene Ng, A.L.Cox. // Technical report. – Rice University, 2011. –
Mode of access: https://www.cs.rice.edu/~eugeneng/papers/TR10-11.pdf.

59. Tootoonchian A., Gorbunov S., Ganjali Y., Casado M., Sherwood R. On controller performance in software-defined networks [Electronic resourse] / A.Tootoonchian, S.Gorbunov, Y.Ganjali, M.Casado, R.Sherwood // Proceedings of the 2nd USENIX conference on Hot Topics in Management of Internet, Cloud, and

Enterprise Networks and Services, Hot-ICE'12 (USA, CA, Berkeley, 2012). – USENIX Association, 2012. – P.10. – Mode of access: https://www.usenix.org/system/files/conference/hot-ice12/hotice12-final33_0.pdf.

60. Python Software Foundation US [Electronic resourse]. – Mode of access: https://www.python.org/about/ (Accessed 1 June 2016).

61. Kandula S., Katabi D., Sinha S., Berger A. Dynamic load balancing without packet reordering [Electronic resourse] / S.Kandula, D.Katabi, S.Sinha, A.Berger // ACM SIGCOMM Computer Communication Review (April, 2007). – Vol.37, No.2. –P.53-62. –

Mode of access: https://www.akamai.com/jp/ja/multimedia/documents/technical-publication/dynamic-load-balancing-without-packet-reordering-technical-publication.pdf.

62. Cao Z., Wang Z., Zegura E. Performance of hashing based schemes for Internet load balancing [Electronic resourse] / Z.Cao, Z.Wang, E.Zegura // Proceedings IEEE INFOCOM (Israel, Tel Aviv, Mart, 2000). – P.332-341. – Mode of access: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.41.3646&rep=rep1&type=pdf.

63. Leung K.C., Li V.O.K. Generalized load sharing for packetswitching networks: Theory and packet-based algorithm [Electronic resourse] / K.C.Leung, V.O.K. Li // IEEE Trans. Parallel and Distributed Systems (July, 2006). – Vol.17, No.7. – P.694-702. –

Mode of access: http://hub.hku.hk/bitstream/10722/44787/1/121835.pdf?accept=1.

64. Kim J., Ahn B. Next-hop selection algorithm over ECMP [Electronic resourse] / J.Kim, B.Ahn // Proceedings sia Pacific Conference on Communications (APCC 2006) (Korea, Busan, August, 2006). Mode of access: http://ieeexplore.ieee.org/document/4023075/.

65. Fernandez J.C., Taleb T., Guizani M., Kato N. Bandwidth aggregation-aware dynamic QoS negotiation for real-time video streaming in next-generation wireless networks [Electronic resourse] / J.C.Fernandez, T.Taleb, M.Guizani, N.Kato // IEEE

Trans. Multimedia (October, 2009). – Vol.11. – Issue 6. – P.1082-1093. – Mode of access: http://dl.acm.org/citation.cfm?id=1653060.

66. Chim T.W., Yeung K.L., Lui K.-S. Traffic distribution over equal-cost-multi-paths [Electronic resourse] / T.W.Chim, K.L.Yeung, K.-S.Lui // Computer Networks (November, 2005). – Vol.49 (4). – P.465-475. – Mode of access: http://www.sciencedirect.com/science/article/pii/S1389128605000411?np=y.

67. Commons Math: The Apache Commons Mathematics Library [Electronic resourse]. – Mode of access: http://commons.apache.org/proper/commons-math/ (Accessed 1 June 2016).

68. OMNeT++ Discrete Event Simulator [Electronic resourse]. – Mode of access: https://omnetpp.org/ (Accessed 1 June 2016).

69. OPNET Technologies – Network Simulator [Electronic resourse]. – Mode of access: www.riverbed.com/products/steelcentral/opnet.html?redirect=opnet (Accessed 1 June 2016).

70. Mininet: An Instant Virtual Network on your Laptop (or other PC) – Mininet [Electronic resourse]. – Mode of access: mininet.org/ (Accessed 1 June 2016).

71. Open vSwitch [Electronic resourse]. – Mode of access: openvswitch.org/ (Accessed 1 June 2016).

72. GitHub – pfa/python-eigrp: An implementation of EIGRP in Python [Electronic resourse]. – Mode of access: https://github.com/pfa/python-eigrp (Accessed 1 June 2016).

73. Shpur O. Improving the Quality of Composite Services Through Improvement of Cloud Infrastructure Management / O.Shpur, M.Klymash, M.Seliuchenko, B.Strykhaliuk, O.Lavriv // International Journal of Computer Science and Information Security (IJCSIS). – 2015. – Vol.13, No.9. – P.36-44 (Index Copernicus).

74. Beshley M.M. Increasing the efficiency of real-time content delivery by improving the technology of priority assignment and processing of IP traffic / M.Beshley, M.Seliuchenko, O.Lavriv, V.Chervenets, H.Kholiavka, M.Klymash // Smart Computing Review. – 2015. – Vol.5, No.2. – P.76-88.

75. Климаш М.М. Метод підвищення ефективності використання мережевих ресурсів інформаційно-телекомунікаційних систем / М.М.Климаш, О.М.Шпур, М.О.Селюченко, Б.В.Киричук, Т.В.Мельник // Вісник Національного університету «Львівська політехніка» “Радіоелектроніка та телекомунікації”. – Львів, 2015. – №818. – С.137-151 (Index Copernicus, Google scholar).

76. Бешлей М.І. Оцінка адекватності функціонування програмного маршрутизатора у процесі обслуговування мультимедійного трафіку / М.І.Бешлей, О.М.Селюченко, О.А.Лаврів, А.Р.Масюк, Г.В.Холявка // Вісник Національного університету «Львівська політехніка» “Радіоелектроніка та телекомунікації”. – Львів, 2015. – №818. – С.162 – 173 (Index Copernicus, Google scholar).

77. Klymash M.A. Novel approach of optimum multicriteria vertical handoff algorithm for heterogeneous wireless networks / M.Klymash, B.Stryhalyuk, I.Demydov, M.Beshley, M.Seliuchenko // International Journal of Engineering and Innovative Technology (IJEIT). – 2014. – Vol.4. – Issue 5(4). – P.42-52 (Index Copernicus).

78. Strykhalyuk B. Implementation of wireless heterogeneous network based on LTE core virtualization formilitary communication systems / B.Strykhalyuk, I.Kahalo, M.Brych, M.Beshley, M.Seliuchenko // Системи озброєння і військова техніка: наук. журнал. – Х: Харк. ун-т повітр. сил ім. Івана Кожедуба, 2014. – №4 (40). – С.125-132 (Index Copernicus).

79. Стрихалюк Б.М. Алгоритми пошуку шляху за критерієм мінімальної затримки для центру обробки даних / Б.М.Стрихалюк, О.М.Шпур, М.О.Селюченко, Т.В.Андрусів // Вісник Національного університету «Львівська політехніка» “Радіоелектроніка та телекомунікації”. – Львів, 2014. – №796. – С.176-181 (Index Copernicus, Google scholar).

80. Коваль Б.В. Дослідження площини управління програмно-конфігурованих мереж на основі розподіленої системи функцій віртуалізації / Б.В.Коваль, М.О.Селюченко, Г.В.Мельник, А.В.Ковальчук // Вісник Національного

університету «Львівська політехніка» “Радіоелектроніка та телекомунікації”. – Львів, 2014. – №796. – С.164-175 (Index Copernicus, Google scholar).

81. Стрихалюк Б.М. Визначення доступності програмних комплексів у системах з сервісно-орієнтованою архітектурою / Б.М.Стрихалюк, О.М.Шпур, М.О.Селюченко // Наукові праці ДонНТУ. Серія: Обчислювальна техніка та автоматизація. – Донецьк, 2014. – №2 (27). – С.109-120.

82. Забезпечення якості обслуговування та оптимізація бізнес-процесів у розподілених системах на основі сервісно-орієнтованої архітектури / М.М.Климаш, І.В.Демидов, М.О.Селюченко, І.Д.Орлевич // Вісник Національного університету «Львівська політехніка» “Радіоелектроніка та телекомунікації”. – Львів, 2013. – №766. – С.150-155.

83. Дослідження методів побудови та доступу до спільного транспортного середовища у мережах нового покоління (NGN) / М.М.Климаш, Б.А.Бугиль, М.О.Селюченко, Л.В.Голейчук // Вісник Національного університету «Львівська політехніка» “Радіоелектроніка та телекомунікації”. – Львів, 2012. – №738. – С.123-129.

84. Seliuchenko M. Development of monitoring system for end-to-end packet delay measurement in Software-Defined Networks / M.Seliuchenko, M.Beshley, O.Panchenko, M.Klymash // Modern problems of radio engineering, telecommunications, and computer science. Proceedings of the International Conference TCSET’2016 (Lviv-Slavske, Ukraine, February 23–26, 2016). – Lviv: Publishing House of Lviv Polytechnic, 2016. – P.667-670.

85. Beshley M. Investigation the modified priority queuing method based on virtualized network testbed / M.Beshley, V.Romanchuk, M.Seliuchenko, A.Masiuk // Proceedings of The XIIIth International Conference “The experience of designing and application of CAD Systems in microelectronics” CADSM’2015 (Lviv-Poljana, Ukraine, February 24-27, 2015). – Lviv: Publishing House of Lviv Polytechnic, 2015. – P.1-4.

86. Klymash M. Mobility managementand vertical handover decisioninan always best connected heterogeneous network / M.Klymash, M.Seliuchenko, M.Beshley,

M.Brych // Proceedings of The XIIIth International Conference “The experience of designing and application of CAD Systems in microelectronics” CADSM’2015 (Lviv-Poljana, Ukraine, February 24-27, 2015). – Lviv: Publishing House of Lviv Polytechnic, 2015. – P.103-105.

87. Klymash M. Increasing wavelengths utilization efficiency in OTNoDWDM network based on local resource distribution method / M.Klymash, M.Seliuchenko, M.Beshley, S.Redchuk // Second IEEE International Scientific-Practical Conference “Problems of Infocommunications. Science and Technology” (PICS&T’2015): Conference proceedings (Kharkiv, Ukraine, October 13-15, 2015). – Kh: KHNURE, 2015. – P.157-160.

88. Klymash M. Enhancing reliability of transport software-defined networks using flow table mutual reservation method / M.Klymash, M.Seliuchenko, A.Kovalchyk, O.Lavriv // Modern problems of radio engineering, telecommunications and computer science: Proceedings of the International Conference TCSET’2014 dedicated to the 170th anniversary of Lviv Polytechnic National University (Lviv-Slavske, Ukraine, February 25 – March 1, 2014). – Lviv: Publishing House of Lviv Polytechnic, 2014. – P.573-576.

89. Seliuchenko M.O. Efficiency optimization of distributed systems using mechanism of “multivariate access” / M.O.Seliuchenko, M.M.Klymash // The Experience of Designing and Application of CAD Systems in Microelectronics (CADSM’2013): Proceedings of the 12th International Conference (Polyana-Svalyava, Ukraine, February 19-23, 2013). – Lviv: Polytechnic National University Publishing, 2013. – P.174-177.

90. Климаш М.М. Система моніторингу пакетної затримки в програмно-конфігурованих телекомунікаційних мережах / М.М.Климаш, М.О.Селюченко, О.М.Панченко // X Міжнародна науково-технічна конференція «Проблеми телекомунікацій» ПТ-2016: Збірник матеріалів конференції (м. Київ, 19-22 квітня 2016 р.). – К.: НТТУ «КПІ», 2016. – С.345-347.

91. Бешлей М.І. Розробка та впровадження нового алгоритму планування черг у мережах з диференціацією сервісів / М.І.Бешлей, М.О.Селюченко, Р.С.Колодій

// IX Міжнародна науково-технічна конференція «Проблеми телекомунікацій» ПТ-2015: Збірник матеріалів конференції (м. Київ, 21-24 квітня 2015 р.). – К.: НТТУ «КПІ», 2015. – С.119-121.

92. Стрихалюк Б.М. Метод балансування навантаження на основі інтегрованої архітектури управління з використанням функції NVF / Б.М.Стрихалюк, О.М.Шпур, М.О.Селюченко // IX Міжнародна науково-технічна конференція «Проблеми телекомунікацій» ПТ-2015: Збірник матеріалів конференції (м. Київ, 21-24 квітня 2015 р.). – К.: НТТУ «КПІ», 2015. – С.322-325.

93. Селюченко М.О. Багаторівневе управління ресурсами в гетерогенній мульти- операторській мережі / М.О.Селюченко, Г.В.Бешлей, А.Р.Масюк, М.І.Бешлей // 1st International Conference "Advanced Information and Communication Technologies-2015" (AICT'2015): Conference Proceedings (Lviv, Ukraine, October 29 – November 1, 2015). – Lviv: Publishing House of Lviv Polytechnic, 2015. – P.125-128.

94. Климаш М. Еволюція технологій оптичних WDM систем / М.Климаш, М.Селюченко // 1st International Conference "Advanced Information and Communication Technologies" (AICT'2015): Conference Proceedings (Lviv, Ukraine, October 29 – November 1, 2015). – Lviv: Publishing House of Lviv Polytechnic, 2015. – P.181-182.

95. Бешлей М.І. Підвищення ефективності роботи гетерогенних мереж методом динамічного перерозподілу ресурсів між різними безпроводовими технологіями / М.І.Бешлей, М.О.Селюченко, П.О.Гуськов, А.Р.Масюк // Міжнародна науково-технічна конференція «Сучасні інформаціно-телекомунікаційні технології»: Матеріали науково-технічної конференції (м. Київ, 17-20 листопада 2015 р.). – К: ДУТ, 2015. – Т.2. – С.49-50.

96. Селюченко М.О. Динамічне управління якістю послуг на основі SOCCA в конвергентних телекомунікаційних мережах / М.О.Селюченко, М.М.Климаш, М.І.Бешлей // Проблеми телекомунікацій: Матеріали VIII Міжнародної науково-технічної конференції (м. Київ, 22–25 квітня 2014 р.). – К.: НТУУ "КПІ", 2014. – С.50-52.

97. Klymash M. System for increasing quality of service of multimedia data in convergent networks // M.Klymash, B.Stryhaluk, M.Beshley, M.Seliuchenko / Международная научно-практическая конференция "Проблемы инфокоммуникаций. Наука и технологии" (PIC S&T-2014) 5-го Международного радиоэлектронного форума "Прикладная радиоэлектроника. Состояние и перспективы развития" МРФ-2014: Сборник научных трудов: материалы форума в 4-х томах (Харьков, 14-17 октября 2014 г.). – Харьков, 2014. – Т.2. – С.96-102.

98. Климаш М.М. Методи інтелектуального вертикального хендоверу в безпровідних системах доступу на основі хмарних технологій // М.М.Климаш, М.І.Бешлей, М.О.Селюченко // Фізико-технологічні проблеми радіотехнічних пристроїв, засобів телекомунікацій, нано- та мікроелектроніки: Матеріали IV Міжнародної науково-практичної конференції (м. Чернівці, 23-25 жовтня 2014 р.). – Чернівці, 2014. – С.108-109.

99. Beshley M.I. A novel approach for providing quality of service in multiservice network environment / M.I.Beshley, M.O.Seliuchenko // Сучасні проблеми телекомунікацій і підготовка фахівців в галузі телекомунікацій-2014: Матеріали конференції (м. Львів, 30 жовтня – 2 листопада 2014 р.). – Львів, 2014. – С.34-37.

100. Климаш М.М. Підвищення ефективності функціонування мультисервісної мережі на основі адаптивного вибору алгоритму обслуговування черг / М.М.Климаш, І.О.Кагало, М.О.Селюченко // Сучасні проблеми телекомунікацій і підготовка фахівців в галузі телекомунікацій: Матеріали конференції (м. Львів, 30 жовтня – 2 листопада 2014 р.). – Львів, 2014. – С.133-134.

101. Климаш М.М. Забезпечення відмовостійкості багаторівневої ієрархії управління у програмноконфігурованих мережах / М.М.Климаш, М.О.Селюченко, О.А.Лаврів // Сучасні проблеми телекомунікацій і підготовка фахівців в галузі телекомунікацій: Матеріали конференції (м. Львів, 30 жовтня – 2 листопада 2014 р.). – Львів, 2014. – С.225-228.

102. Климаш М.М. Система підвищення ефективності управління потоками в програмно-конфігурованих мережах / М.М.Климаш, М.І.Олексін, М.О.Селюченко // Сучасні проблеми телекомунікацій і підготовка фахівців в галузі телекомунікацій: Матеріали конференції (м. Львів, 30 жовтня – 2 листопада 2014 р.). – Львів, 2014. – С.229-232.

103. Климаш М.М. Розподілена система електронного урядування на основі інтеграції прогамно-конфігурованих технологій та моделей cloud сервісів / М.М.Климаш, М.О.Селюченко // Сучасні проблеми телекомунікацій і підготовка фахівців в галузі телекомунікацій: Матеріали конференції (м. Львів, 30 жовтня – 2 листопада 2014 р.). – Львів, 2014. – С.11-14.

104. Климаш М.М. Аналіз технологій впровадження мобільного веб-сервісу в системах стільникового зв'язку / М.М.Климаш, М.О.Селюченко // VI Международный научно-технический симпозиум «Новые технологии в телекоммуникацыях» (ГУИКТ-КАРПАТЫ'2013): Сборник тезисов (Киев, 21 – 25 января 2013г.). – К.: ГУИКТ, 2013. – С.62-64.

105. Seliuchenko M.O. Analysis of load balancing methods for improving of utilization efficiency of hardware resources of distributed systems / M.O.Seliuchenko, M.M.Klymash, O.A.Lavriv // 23rd International Crimean Conference "Microwave & Telecommunication Technology" (CriMiCo'2013): Conference Proceedings (Sevastopol, September 8-13, 2013). – Sevastopol: Weber Publishing, 2013. – Vol.1. – P.515-517.

106. Климаш М.М. Аналіз принципів побудови сучасних дата-центрів на основі Cloud-архітектури / М.М.Климаш, М.О.Селюченко // Сучасні проблеми телекомунікацій і підготовка фахівців в галузі телекомунікацій-2013: Матеріали конференції (м. Львів, 30 жовтня – 2 листопада 2013 р.). – Львів, 2013. – С.110-113.

107. Климаш М.М. Оптимізація і балансування навантаження в мобільних мережах за допомогою методу «поліваріантного доступу» / М.М.Климаш, М.О.Селюченко // VIII Міжнародна науково-технічна конференція «Сучасні інформаційно-комунікаційні технології» (COMINFO'2012-Livadia): Збірник тез (м. Київ, 1-5 жовтня 2012 р.). – К.: ДУІКТ, 2012. – С.32-34.

ДОДАТОК. АКТИ ВПРОВАДЖЕННЯ ДИСЕРТАЦІЙНИХ ДОСЛІДЖЕНЬ

"ЗАТВЕРДЖУЮ"
Директор Львівської філії
ПАТ "Укртелеком"
_____ Андрухів Т.В., к.т.н.
"21" 04 2016 р.

АКТ
про використання результатів дисертаційної роботи
аспіранта кафедри телекомунікацій
Національного університету "Львівська політехніка"
Селюченка Мар'яна Олександровича
на тему:
"Моделі та алгоритми підвищення якості обслуговування у телекомунікаційних програмно-конфігурованих мережах"

Даний акт складений про те, що у Львівській філії ПАТ "Укртелеком" використані результати дисертаційної роботи Селюченка М.О. "Моделі та алгоритми підвищення якості обслуговування у телекомунікаційних програмно-конфігурованих мережах". А саме:

- використано розроблену систему моніторингу, що дало змогу оперативно відстежувати стан та виявляти аномальну поведінку мережі та її компонентів на основі централізованого оброблення та предсталення результатів моніторингу процесу передавання даних у комутаційних пристроях;
- використано моделі адаптації системи у розподіленій інформаційно-телекомунікаційній інфраструктурі, що дало змогу підвищити точність оцінки завантаження мережних каналів та знизити обсяг службового трафіку;

Результати експериментальних досліджень, що виконані на виробничих потужностях Львівської філії ПАТ "Укртелеком", відповідають результатам досліджень, що представлені у дисертаційній роботі, похибка не перевищує 3%.

Начальник відділу
планування мереж з КК Качан В.М.

"ЗАТВЕРДЖУЮ"
Проректор з науково-педагогічної роботи
НУ "Львівська політехніка"
_____ доц. Давидчак О.Р.
"19" 04 2016 р.

АКТ

про використання результатів кандидатської дисертаційної роботи
Селюченка Мар'яна Олександровича
"Моделі та алгоритми підвищення якості обслуговування у телекомунікаційних програмно-конфігурованих мережах"
у навчальному процесі кафедри телекомунікацій

Даний акт складений комісією у складі:
- д.т.н., проф. Убізський С.Б., голова методичної ради Інституту телекомунікацій, радіоелектроніки та електронної техніки;
- к.т.н., доц. Озірковський Л.Д., декан базової вищої освіти Інституту телекомунікацій, радіоелектроніки та електронної техніки;
- д.т.н., проф. Пелішок В.О., професор кафедри телекомунікацій

про те, що в навчальному процесі кафедри телекомунікацій використано результати кандидатської дисертаційної роботи "Моделі та алгоритми підвищення якості обслуговування у телекомунікаційних програмно-конфігурованих мережах", а саме:

- модернізовано курси лекцій для студентів напряму 6.050903 «Телекомунікації» з дисциплін: «Телекомунікаційні та інформаційні мережі, ч.1» - у частині теоретичних основ управління розподіленими сервісно-орієнтованими телекомунікаційними системами з використанням контролера; «Телекомунікаційні системи передавання інформації» - у частині, що стосується методики побудови та модернізації транспортних телекомунікаційних мереж на основі модернізації площини керування з використанням розподілених програмних контролерів;

- модернізовано курс лекцій з дисципліни «Розподілені сервісні системи та Cloud-технології» для студентів спеціальності 8.05090301 «Інформаційні мережі зв'язку», у якому використано запропонований у роботі метод оцінки якості обслуговування потоків окремих сервісів при наданні послуг розподілених хмаринкових систем.

Члени комісії: _____ Убізський С.Б.
_____ Озірковський Л.Д.
_____ Пелішок В.О.

"ЗАТВЕРДЖУЮ"

Директор
ПП "Цифрові технології"

Танчак З.В.

"26" 04 2016 р.

АКТ

про використання результатів дисертаційної роботи
Селюченка Мар'яна Олександровича на тему:
"Моделі та алгоритми підвищення якості обслуговування у телекомунікаційних програмно-конфігурованих мережах"

Даний акт складений про те, що у ПП "Цифрові технології" для підвищення якості обслуговування користувачів у процесі надання мультисервісних послуг з використанням телекомунікаційних програмно-конфігурованих мереж використані результати дисертаційної роботи Селюченка М.О. "Моделі та алгоритми підвищення якості обслуговування у телекомунікаційних програмно-конфігурованих мережах", представленої на здобуття наукового ступеня кандидата технічних наук, а саме:

- для зменшення втрат внаслідок перевантаження каналів при неочікуваних стрибках інтенсивності навантаження використано модель балансування навантаження, яка дає змогу забезпечити високу ефективність використання магістральних каналів мережі та забезпечити необхідний QoS для мультимедійних потоків;
- використано алгоритм перерозподілу потоків з урахуванням ідентифікації потоків на основі відносних пріоритетів, що дало змогу забезпечити якість обслуговування потоків реального часу за рахунок перерозподілу даних в каналах з гіршими параметри якості обслуговування;

Внаслідок перевірки використаних моделей на мережному обладнанні у ПП "Цифрові технології" встановлено, що результати знаходяться в межах п'ятивідсоткового середньоквадратичного відхилення від поданих у дисертаційній роботі.

Провідний інженер Дрофяк А.М.

"ЗАТВЕРДЖУЮ"
Директор ТОВ "Літех"
_____
"_15_" квітня 2016 р.

АКТ
про використання результатів дисертаційної роботи
аспіранта кафедри телекомунікацій
Національного університету "Львівська політехніка"
Селюченка Мар'яна Олександровича
на тему:
"Моделі та алгоритми підвищення якості обслуговування у телекомунікаційних програмно-конфігурованих мережах"

Даний акт складений про те, що у ТОВ «Літех» використані результати дисертаційної роботи Селюченка М.О. "Моделі та алгоритми підвищення якості обслуговування у телекомунікаційних програмно-конфігурованих мережах". А саме:

- використання удосконаленої моделі маршрутизації потоків на основі ідентифікації потоку з використанням його відносного пріоритету дало змогу досягнути рівномірного завантаження каналів та підвищення якості обслуговування потоків реального часу у середньому на 15 % в умовах перевантаження магістральних каналів у мережі;
- для оцінки часових параметрів якості обслуговування мультимедійних потоків окремих користувачів використано метод вимірювання затримки передавання пакетів окремого потоку, який в умовах високого завантаження каналів, дає змогу більш точно оцінити погіршення затримки передавання пакетів, а отже вчасно здійснити перерозподіл навантаження з метою забезпечення необхідної якості обслуговування.

Результати експериментальних досліджень, виконаних на виробничих потужностях ТОВ«Літех» відповідають результатам досліджень, що представлені у дисертаційній роботі, похибка не перевищує 3%.

Головний інженер Калиняк В.Я.