ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ТЕЛЕКОМУНІКАЦІЙ
Кафедра Мобільних та відеоінформаційних технологій

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Моделі підвищення якості обслуговування у телекомунікаційних програмно-конфігурованих мережах»

на здобуття освітнього ступеня магістра
зі спеціальності 172 Телекомунікації та радіотехніка
освітньо-професійної програми Апаратура радіозв'язку, радіомовлення і телебачення

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ Андрій КАРНАСЮК
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

Виконав: здобувач вищої освіти
групи АРДМ-62
Андрій КАРНАСЮК

Керівник: Лариса ДАКОВА

Рецензент: _____

Київ – 2023

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
Навчально-науковий інститут телекомунікацій

Кафедра	Мобільних та відеоінформаційних технологій
Ступінь вищої освіти	Магістр
Спеціальність	172 Телекомунікації та радіотехніка
Освітньо-професійна програма	Апаратура радіозв'язку, радіомовлення і телебачення

ЗАТВЕРДЖУЮ
Завідувачка кафедрою МВТ
_____ Наталія РУДЕНКО
« » _____ 2023 року

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Карнасюку Андрію В'ячеславовичу

1. Тема кваліфікаційної роботи: «Моделі підвищення якості обслуговування у телекомунікаційних програмно-конфігурованих мережах»,
керівник кваліфікаційної роботи Лариса ДАКОВА, к.т.н., доцент,
затверджені наказом Державного університету інформаційно-комунікаційних технологій від 19.10.2023 р. №145.

2. Строк подання кваліфікаційної роботи 25.12.2023 р.
3. Вихідні дані до кваліфікаційної роботи:
    1. Програмно-конфігурована мережа.
    2. Технологічна основа побудови транспортної мережі.
    3. Обслуговування інформаційних потоків.
    4. Науково-технічна література.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):
    1. Аналіз моделей функціонування програмно-конфігурованих телекомунікаційних мереж.
    2. Моделі обслуговування інформаційних потоків у програмно-конфігурованих мережах.
    3. Дослідження процесів обслуговування навантаження та моніторингу мережевих ресурсів.
5. Перелік ілюстративного матеріалу: презентація.

6. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської роботи	Строк виконання етапів роботи	Примітка
1.	Підбір науково-технічної літератури	19.10.23	Викон.
2.	Аналіз моделей функціонування програмно-конфігурованих телекомунікаційних мереж	31.10.23	Викон.
3.	Моделі обслуговування інформаційних потоків у програмно-конфігурованих мережах	15.11.23	Викон.
4.	Дослідження процесів обслуговування навантаження та моніторингу мережевих ресурсів	29.11.23	Викон.
5.	Висновки, вступ, реферат	11.12.23	Викон.
6.	Розробка презентації	18.12.23	Викон.

Здобувач вищої освіти Андрій КАРНАСЮК

(підпис) *(Ім'я, ПРІЗВИЩЕ)*

Керівник кваліфікаційної роботи Лариса ДАКОВА

(підпис) *(Ім'я, ПРІЗВИЩЕ)*

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра: 82 стор., 34 рис., 7 табл., 19 дж.

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

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

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

Короткий зміст роботи: У роботі представленно розроблення методів та моделей управління процесами передавання даних у телекомунікаційних ПКМ для підвищення якості обслуговування користувачів та ефективного використання мережевих ресурсів.

Описаний 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. 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.

КЛЮЧОВІ СЛОВА: МАРШРУТИЗАЦІЯ ПОТОКІВ; ПРОГРАМНО-КОНФІГУРОВАНА МЕРЕЖА, ТРАФІК, МАРШРУТИЗАТОР, КОМУТАТОР, АЛГОРИТМ, ШВИДКІСТЬ, РОУМІНГ

ABSTRACT

Text part of the master's qualification work: 82 pages, 34 pictures, 7 table, 19 sources.

The purpose of the work - is to improve the quality of service in software-configured networks by reducing delay and packet loss and increasing the uniformity of network channel loading.

Object of research – the process of maintaining information flows in software-configured networks.

Subject of research – models and algorithms for maintenance of information flows in software-configured networks.

Summary of the work: The work presents the development of methods and models for managing data transmission processes in telecommunication PCMs to improve the quality of user service and effective use of network resources.

The described method of adaptation of the monitoring system makes it possible to ensure the adequacy of the characteristics of traffic transmission processes, in particular, the loading of network interfaces, which makes it possible to more accurately estimate the probability of blocking and avoid channel overload in the case of using dynamic routing. A study of the performance characteristics of the hardware OpenFlow switch HP3500yl was conducted. A feature of the switch is the use of both hardware and software flow tables, and the latter is characterized by significantly lower bandwidth.

KEYWORDS: FLOW ROUTING; SOFTWARE-CONFIGURED NETWORK, TRAFFIC, ROUTER, SWITCH, ALGORITHM, SPEED, ROAMING

ЗМІСТ

ВСТУП 9
1. АНАЛІЗ МОДЕЛЕЙ ФУНКЦІОНУВАННЯ ПРОГРАМНО-КОНФІГУРОВАНИХ ТЕЛЕКОМУНІКАЦІЙНИХ МЕРЕЖ 11
1.1. Архітектура програмно-керованих мереж 11
1.2. Комутація пакетів та балансування навантаження на основі таблиць потоків 16
1.3. Недоліки існуючих методів моніторингу та забезпечення якості обслуговування мультисервісних потоків 21
Висновки 25
2. МОДЕЛІ ОБСЛУГОВУВАННЯ ІНФОРМАЦІЙНИХ ПОТОКІВ У ПРОГРАМНО-КОНФІГУРОВАНИХ МЕРЕЖАХ 28
2.1. Системи моніторингу параметрів функціонування програмно-керованої мережі 28
2.2. Модель адаптації системи моніторингу до властивостей процесу передавання даних 35
2.3. Метод вимірювання затримки передавання пакетів для потоків окремого користувача 39
2.4. Модель балансування навантаження на основі критерію максимально допустимого завантаження каналу 44
Висновки 46
3. ДОСЛІДЖЕННЯ ПРОЦЕСІВ ОБСЛУГОВУВАННЯ НАВАНТАЖЕННЯ ТА МОНІТОРИНГУ МЕРЕЖЕВИХ РЕСУРСІВ 48
3.1. Розробка системи для генерації мультисервісних потоків 48
3.2. Розроблення імітаційної моделі апаратного програмно-конфігурованого комутатора 54
3.3. Дослідження ефективності моделі адаптації системи моніторингу 63
Висновки 72
ВИСНОВКИ 74
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 79
ДЕМОНСТРАТИВНІ МАТЕРІАЛИ 82

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.

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.

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

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.

1. АНАЛІЗ МОДЕЛЕЙ 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.

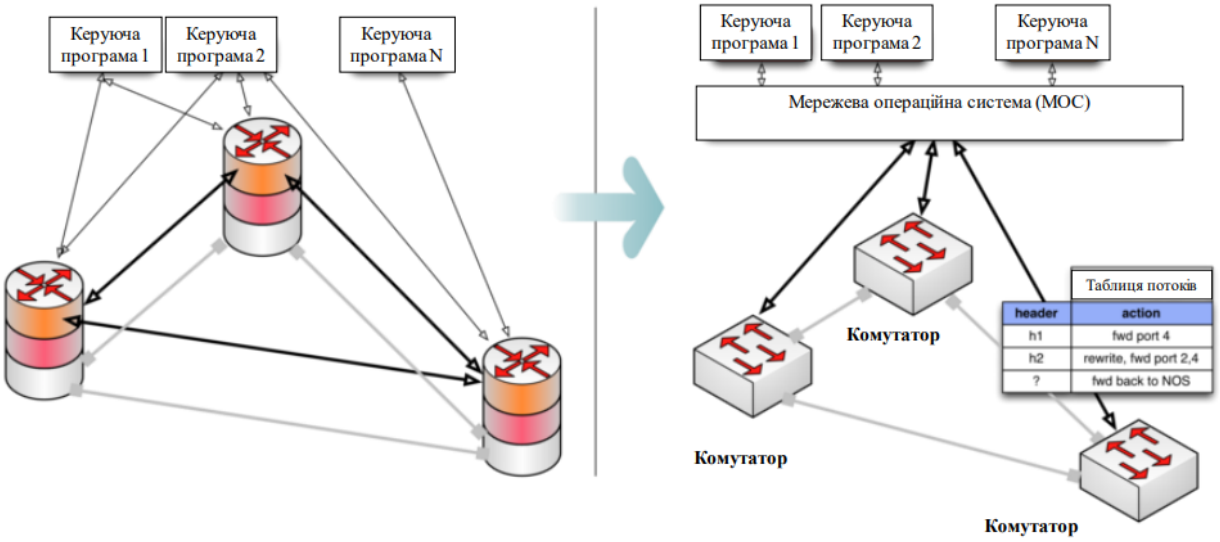


Рисунок 1.1 Архітектура SDN

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 будьякий 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.

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.



Рисунок 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).

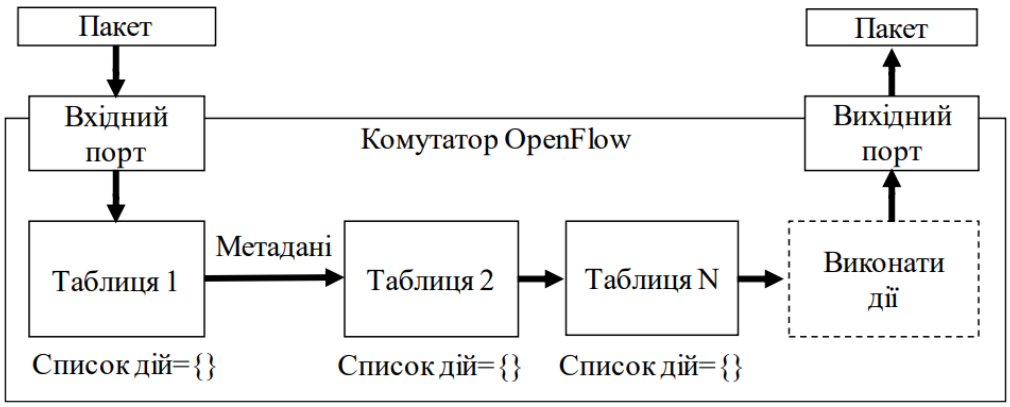


Рисунок 1.3. Модель гнучкої обробки пакетів на основі каскадування таблиць потоків

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.1.1.

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.

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

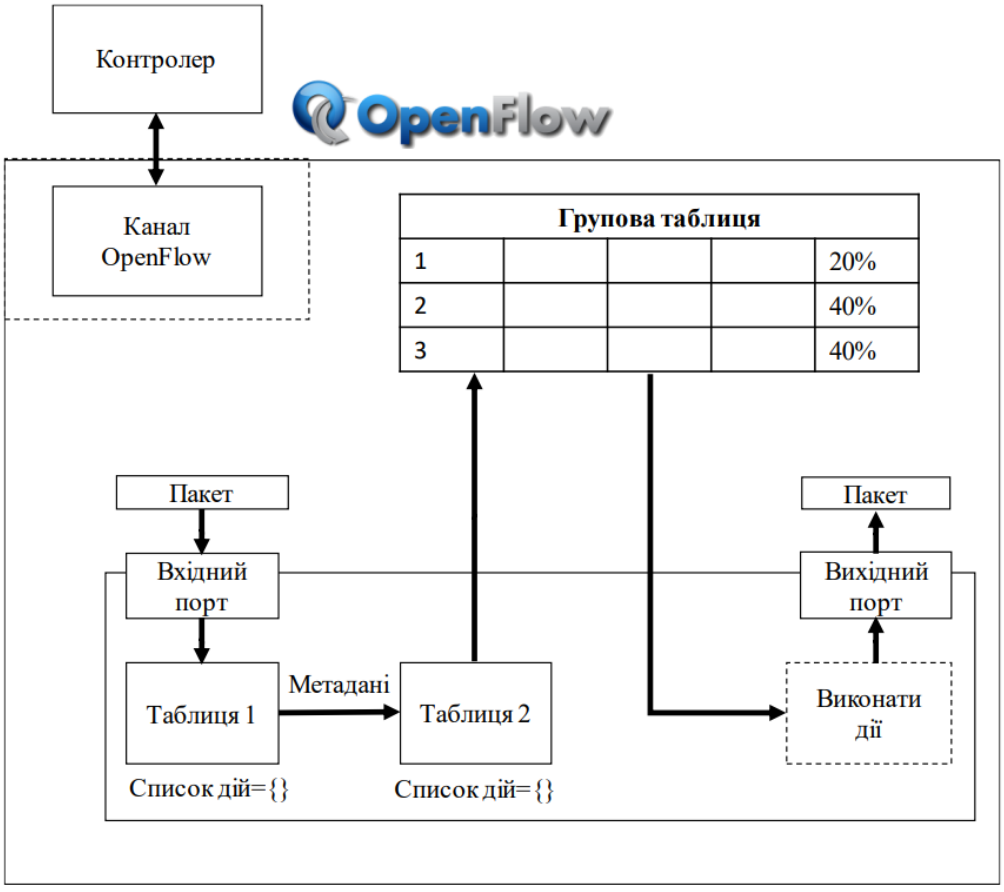


Рисунок 1.4 Модель комутації з використанням групової таблиці потоків та реалізація балансування навантаження на основі вагових коефіцієнтів шляхів

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. 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

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.

2. МОДЕЛІ 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, система моніторингу параметрів функціонування мережевих пристроїв та метод моніторингу з адаптивними параметрами опитування. 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.

2.1. 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.

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.



Рисунок 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.

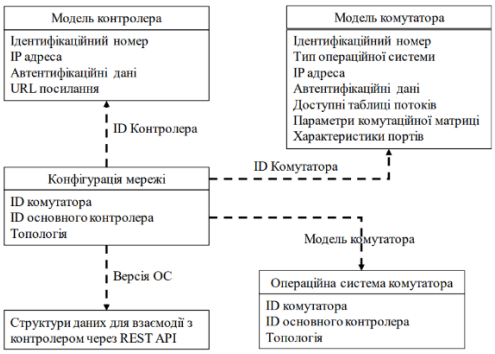


Рисунок 2.2. Модель даних для конфігурації системи моніторингу

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.

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 програмнокерованими комутатори конкретної моделі.

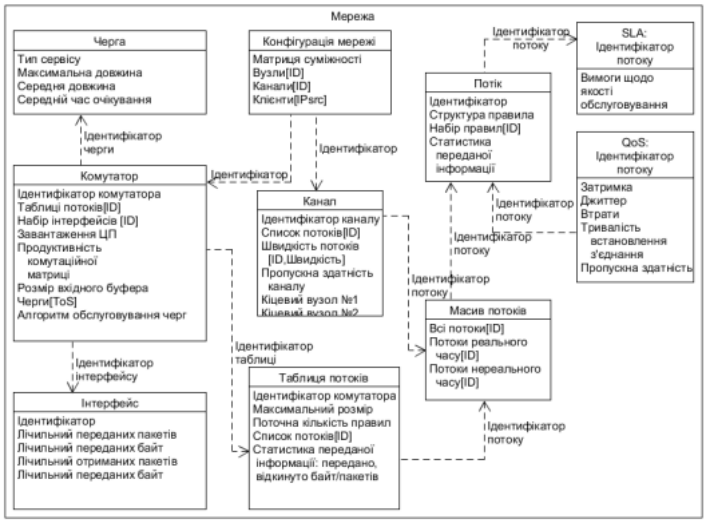


Рисунок 2.3. Модель даних для зберігання поточного стану елементів мережі

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.

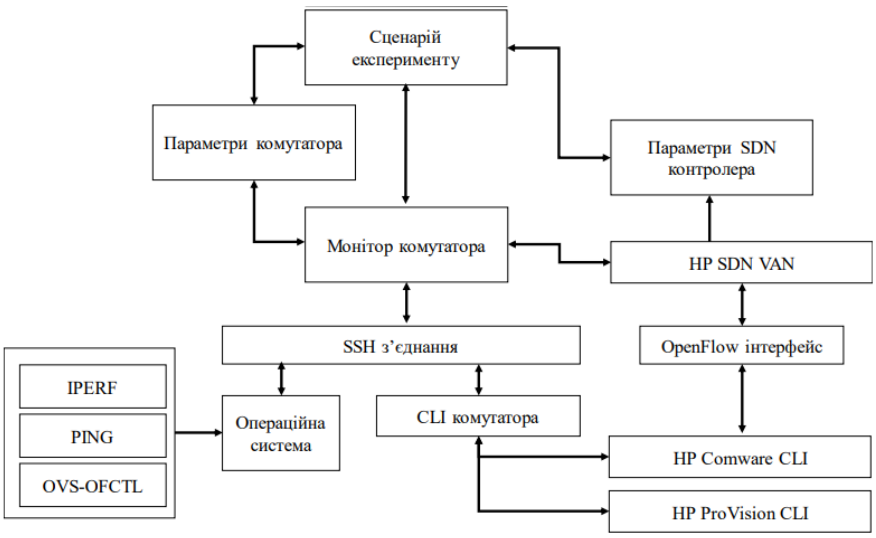


Рисунок 2.4. Архітектура центрального додатка моніторингу програмно-керованої мережі

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):

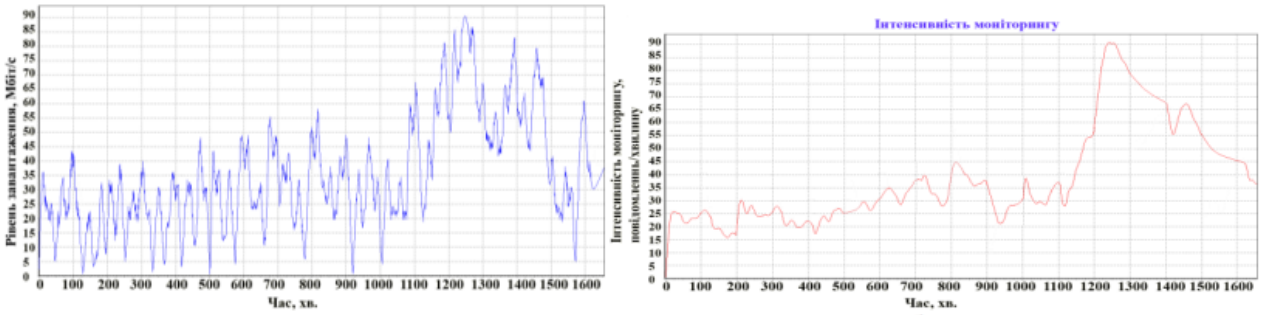
$$f(\rho_i)=\begin{cases}1-\frac{\rho_i}{2}, & 0<\rho_i\le 0.5;\\ \frac{0.1}{(\rho_i-0.367)}, & 0.5<\rho_i\le 1,\end{cases} \tag{2.1}$$

де f – функція завантаженості, що використовується для визначення інтервалу опитування комутатора; ρ – завантаження інтерфейсу комутатора, безрозмірний коефіцієнт; і – 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

Рисунок 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.

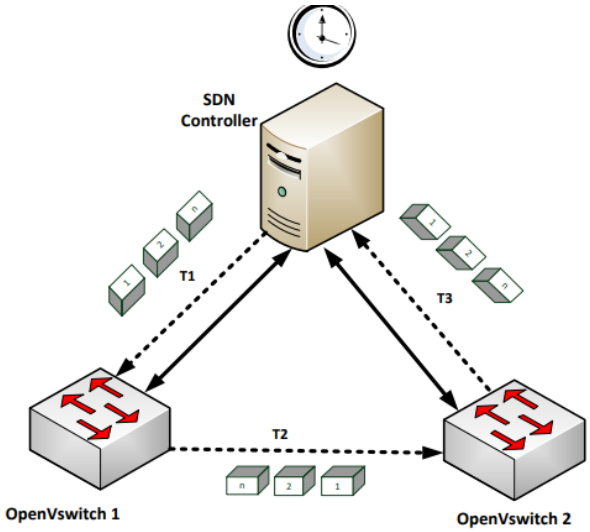


Рисунок 2.6. Схема вимірювання затримки окремого потоку в програмно-керованій мережі

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.

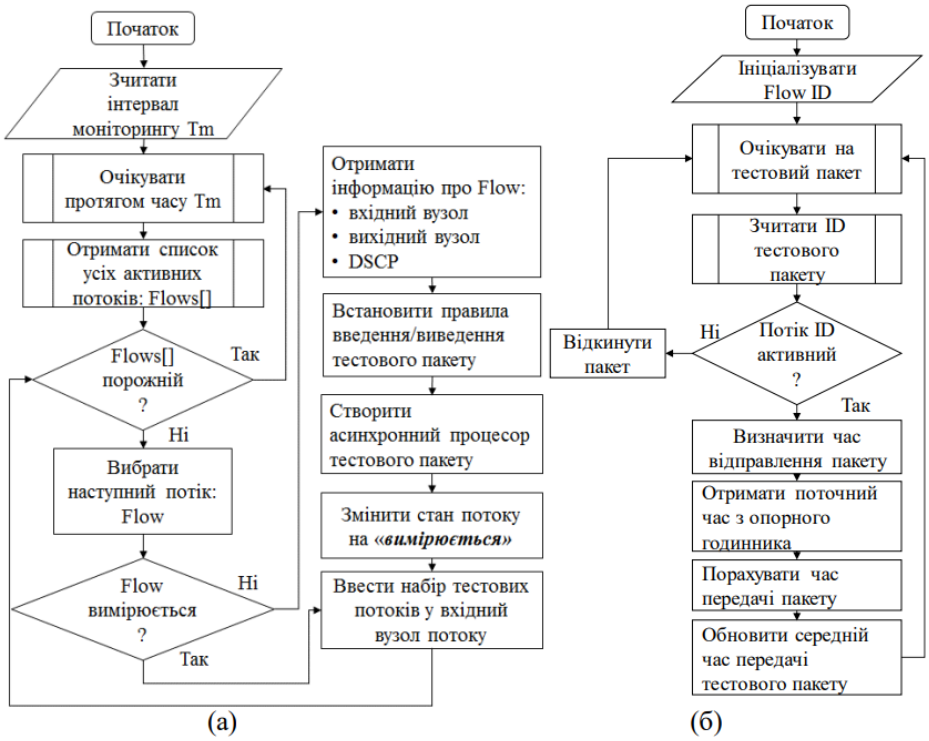


Рисунок 2.7. Алгоритми відправки (а) та очікування (б) пакету для вимірювання затримки потоку

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.

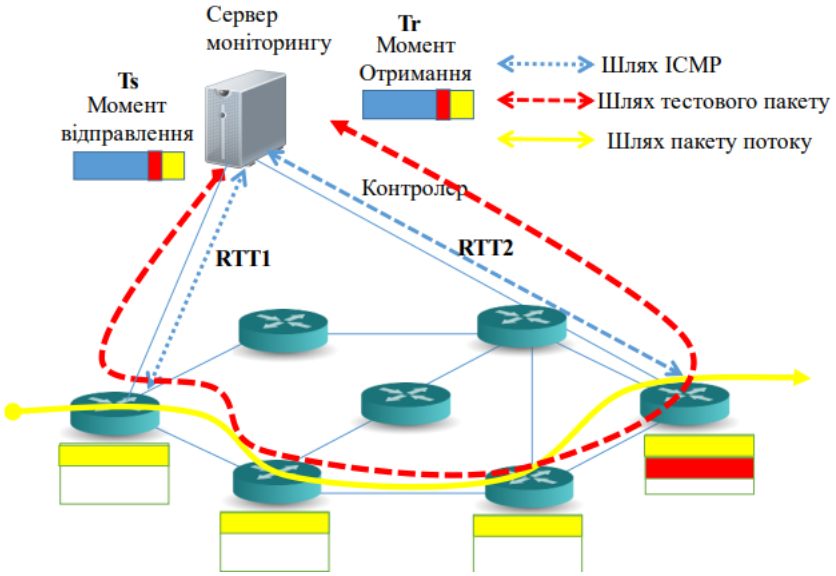


Рисунок 2.8. Схема вимірювання затримки передачі окремого потоку в мережі

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;

s t – момент відправлення тестового пакету з контролера;

r t – момент отримання тестового пакету контролером;

RTT1 та RTT2 – тривалості відгуку першого й останнього комутатора шляху на ICMP запит від контролера.

Джиттер потоку вимірюється як різниця зазначених затримок.

2.4. 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, 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 (рис. 2.9). 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.

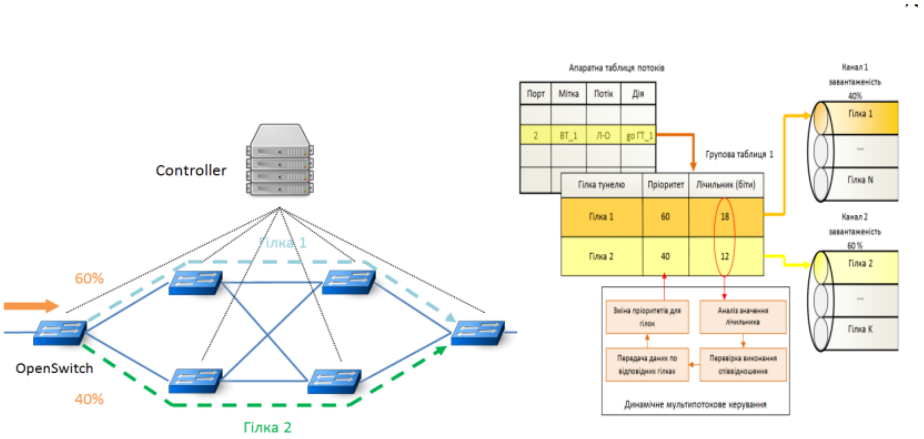


Рисунок 2.9. Принцип балансування трафіку між гілками одного віртуального тунелю на основі зміни пріоритету гілок

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

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.

Спосіб ідентифікації потоку конкретного окремого клієнта на основі відносного пріоритету, розрахованого з допомогою удосконаленого методу призначення пріоритетів трафіку. Представлено 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.

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 метод моніторингу з адаптивними параметрами опитування, який підлаштовує параметри системи моніторингу відповідно до характеру/інтенсивності мережевих процесів, наприклад, завантаження каналу чи вузла. 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.

Потокову модель маршрутизації в програмно-керованих мережах для підвищення ефективності використання мережевих ресурсів та забезпечення необхідної якості обслуговування всім складовим мультисервісного потоку. Модель балансування навантаження в програмно-керованих мережах для керування параметрами потоку та передачі даних в умовах перевантажених каналів.

3. ДОСЛІДЖЕННЯ ПРОЦЕСІВ ОБСЛУГОВУВАННЯ НАВАНТАЖЕННЯ ТА МОНІТОРИНГУ МЕРЕЖЕВИХ РЕСУРСІВ

У розділі проаналізовано 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 моделі балансування навантаження.

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.

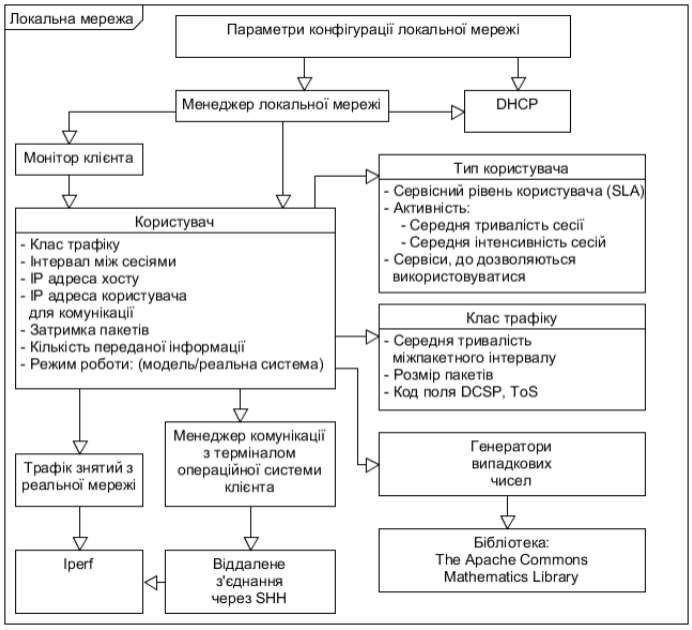


Рисунок 3.1. Структурна схема моделі системи генерації мультисервісного трафіку

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 [17].

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++, OpNet, Mininet) 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.

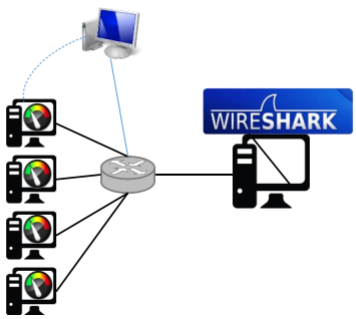


Рисунок 3.2. Схема тестового середовища для дослідження характеристик мультисервісного трафіку та оцінки якості розробленої системи генерації мультисервісного трафіку

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.

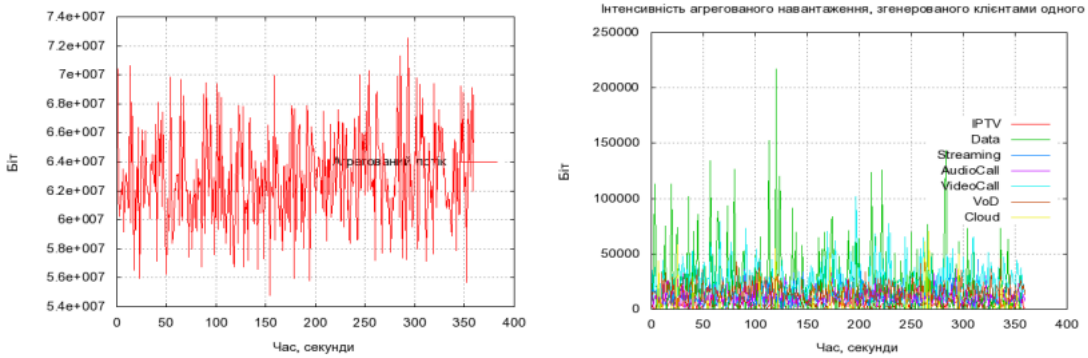


Рисунок 3.3. Мультисервісний агрегований трафік, що генерується десятьма клієнтами з різних фізичних машин за допомогою розробленої системи

На графіку справа (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 [10], який базується на реальному програмному комутаторі OpenVSwitch [21]. 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.

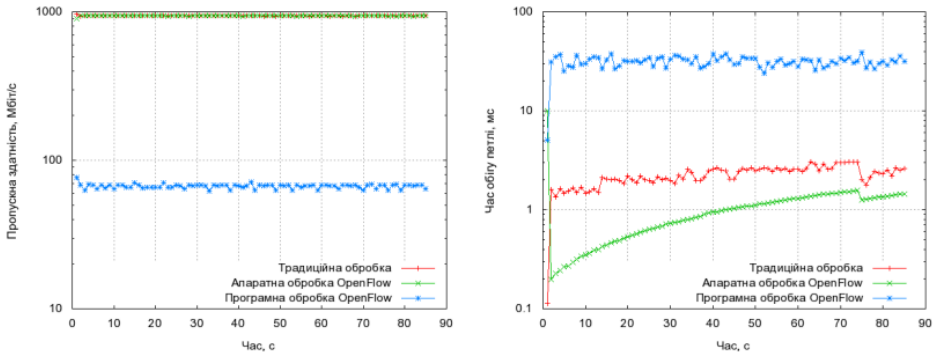


Рисунок 3.4. Порівняння пропускної здатності (а) та тривалості обігу петлі (б) програмно-керованого комутатора в трьох режимах роботи

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.

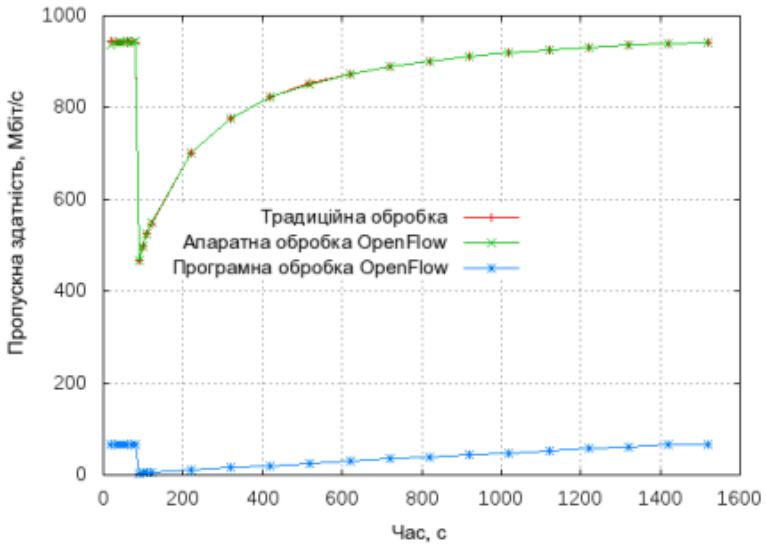


Рисунок 3.5. Дослідження залежності продуктивності програмно-керованого комутатора від розміру пакетів для трьох режимів роботи

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.

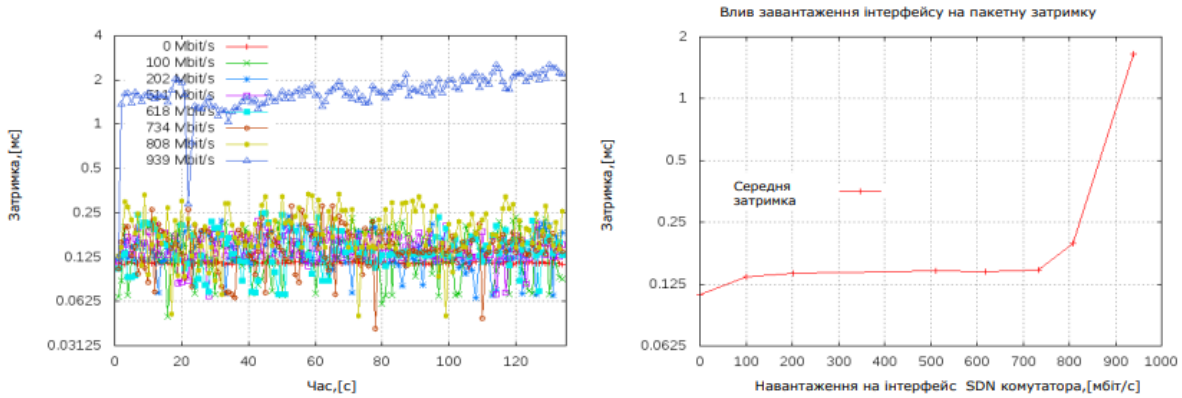


Рисунок 3.6. Характеристика затримки пакетів залежно від завантаження інтерфейсу для апаратної комутації в режимі OpenFlow

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.

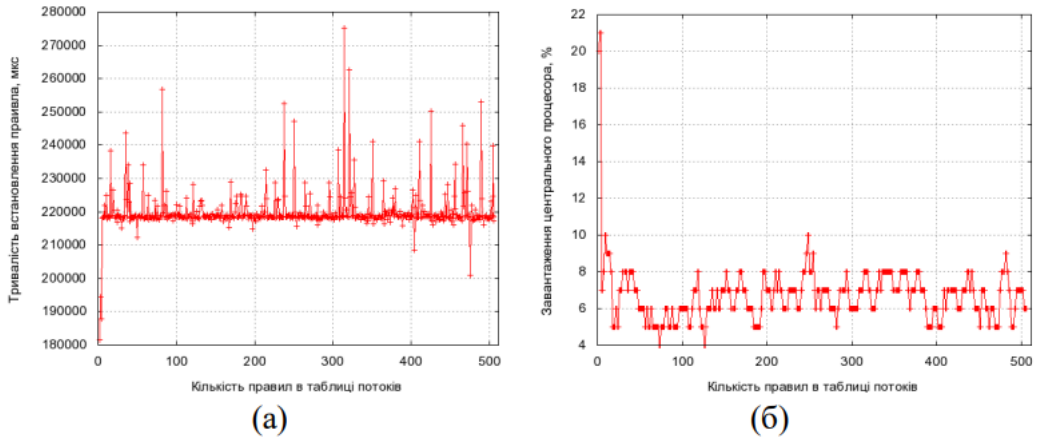


Рисунок 3.7. Характеристика завантаження центрального процесора (а) та тривалості встановлення нових правил (б) залежно від кількості правил у таблиці потоків при максимальному завантаженні інтерфейсу

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.

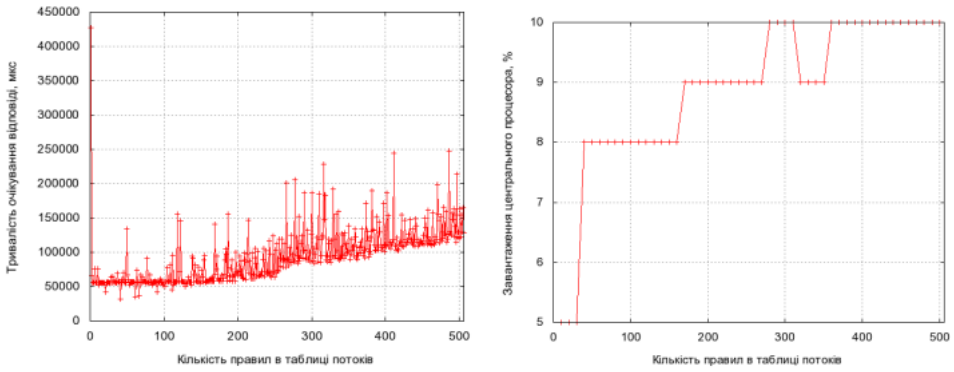


Рисунок 3.8. Залежність тривалості очікування відповіді про стан таблиці потоків залежно від її розмірів (а) та завантаження центрального процесора (б)

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).

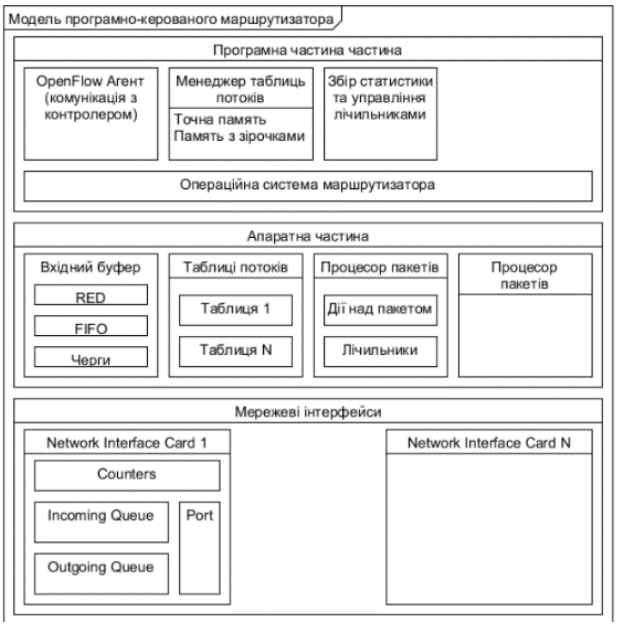


Рисунок 3.9. Структурна схема розробленої моделі програмно-керованого апаратного комутатора HP3500yl

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.

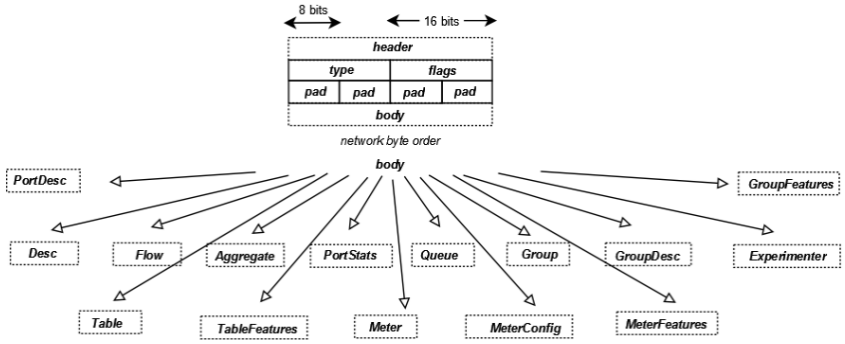


Рисунок 3.10. Класифікація повідомлень для здійснення моніторингу параметрів комутатора та його елементів відповідно

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 HP3500ylможе становити 1 секунду.

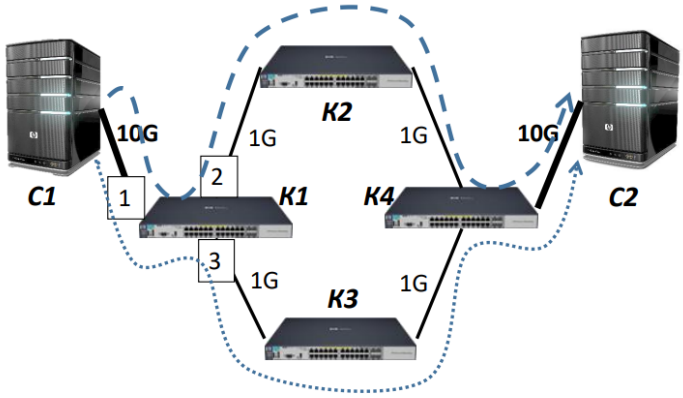


Рисунок 3.11. Конфігурація тестового середовища для дослідження впливу інтенсивності моніторингу на пропускну здатність апаратного комутатора HP3500yl

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.



Рисунок 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.

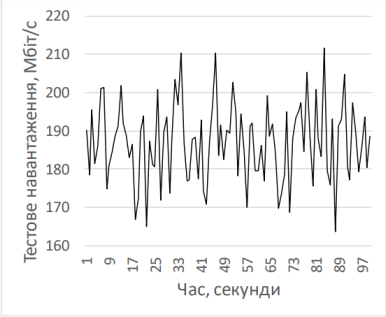


Рисунок 3.13. Згенерований мультисервісний агрегований потік на інтерфейс №1 комутатора К1

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

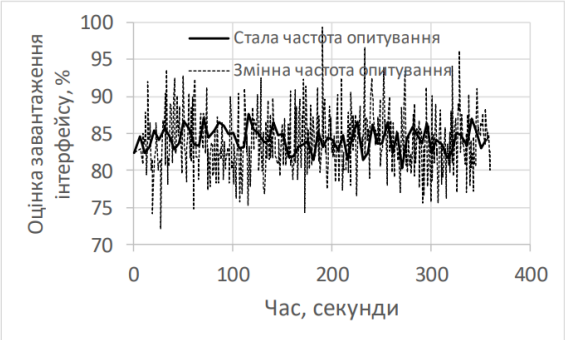


Рисунок 3.14. Завантаження інтерфейсу, отримана в результаті використання методів моніторингу зі статичним та змінним інтервалом опитування

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.

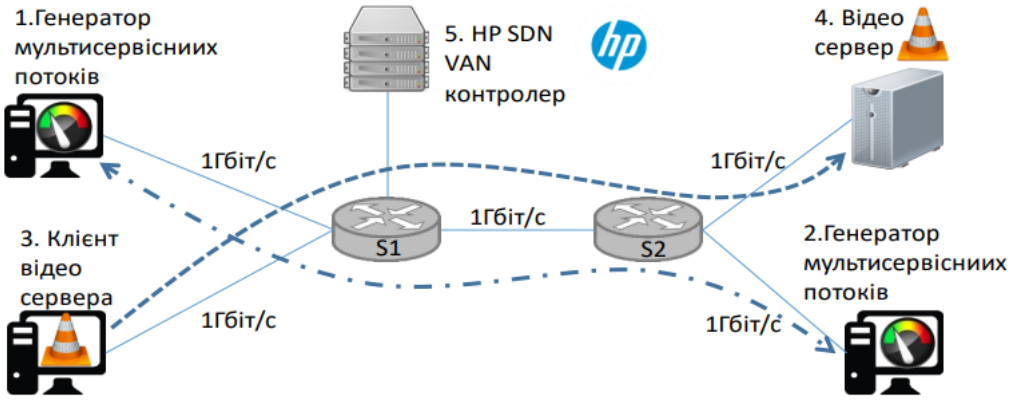


Рисунок 3.15. Схема тестового середовища для тестування впливу завантаження каналу на якість обслуговування потоків реального часу

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.

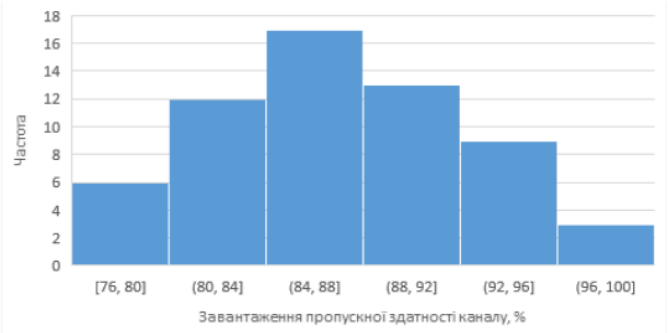


Рисунок 3.16. Розподіл інтенсивності вхідного навантаження на канал між комутаторами 1 та 2

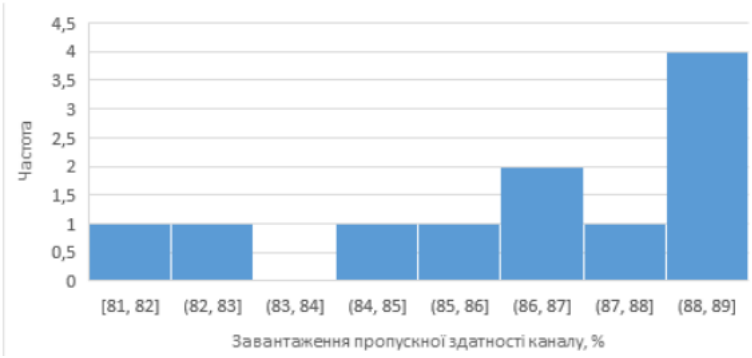


Рисунок 3.17. Розподіл завантаження каналу між комутаторами 1 та 2, отриманий в результаті використання моделі моніторингу зі сталою частотою опитування



Рисунок 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).



Рисунок 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.

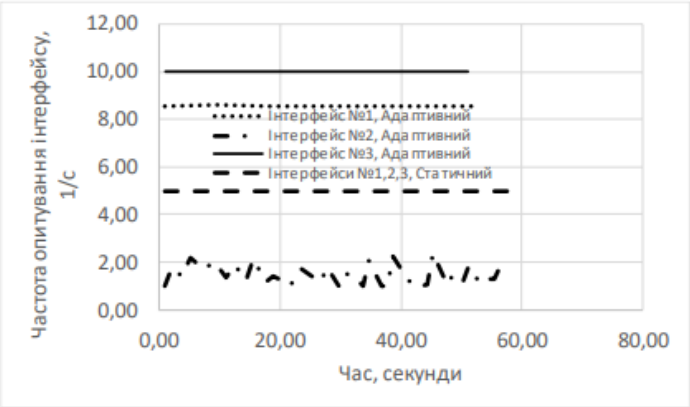
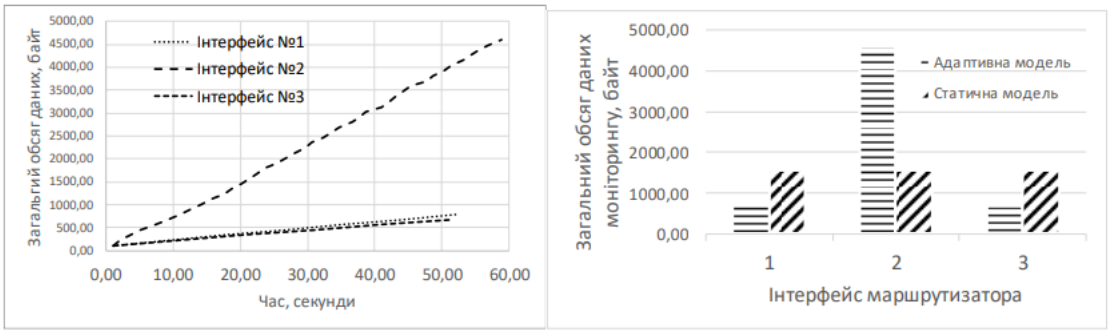


Рисунок 3.20. Інтенсивність моніторингу трьох інтерфейсів

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



(а) (б)

Рисунок 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.

Висновки

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.

ВИСНОВКИ

У роботі представленно 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 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.

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.

Спосіб ідентифікації потоку конкретного окремого клієнта на основі відносного пріоритету, розрахованого з допомогою удосконаленого методу призначення пріоритетів трафіку. Представлено 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.

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 метод моніторингу з адаптивними параметрами опитування, який підлаштовує параметри системи моніторингу відповідно до характеру/інтенсивності мережевих процесів, наприклад, завантаження каналу чи вузла. 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.

Потокову модель маршрутизації в програмно-керованих мережах для підвищення ефективності використання мережевих ресурсів та забезпечення необхідної якості обслуговування всім складовим мультисервісного потоку. Модель балансування навантаження в програмно-керованих мережах для керування параметрами потоку та передачі даних в умовах перевантажених каналів.

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.

Представлений 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.

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/whitepapers/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. 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.

6. 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/

7. 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_l
atency_with_OpenFlow.

8. 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.

9. 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.

10. 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.

11. 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.

12. 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.

13. 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.

14. 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).

15. 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.

16. Бешлей М.І. Розробка та впровадження нового алгоритму планування черг у мережах з диференціацією сервісів / М.І.Бешлей, М.О.Селюченко, Р.С.Колодій 150 // 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.

17. 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).

18. 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.

19. 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).

ДЕМОНСТРАТИВНІ МАТЕРІАЛИ

Протокол аналізу звіту подібності науковим керівником

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Андрій Краснюк

Назва: Моделі підвищення якості обслуговування у телекомунікаційних програмно-конфігурованих мережах

Координатор: Лариса Дакова

Підрозділ: ННІТ

Коефіцієнт подібності 1:17.2

Коефіцієнт подібності 2:23.9

Тривога: 17

Після аналізу Звіту подібності констатую наступне:

☐ виявлені в роботі запозичення є сумлінними і не мають ознак плагіату. Тому робота визнається самостійною і допускається до захисту;

☐ виявлені в роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і самостійності її автора. Роботу направити на доопрацювання;

☐ виявлені в роботі запозичення є недобросовісними і мають ознаки плагіату або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень. У зв'язку з чим, робота не допускається до захисту.

…… ……

Дата *Підпис Наукового керівника*