

## ДОДАТОК А

Графічний матеріал кваліфікаційної роботи

## *Кваліфікаційна робота*

# Методи аналізу та класифікації зображень з використанням гібридного класифікатора

Виконав: ст. гр. СПм-22-6 Колісник Є.Б.

Керівник: проф. Можаяєв О.О.

## **Аналіз предметної області:**

2

**Метою кваліфікаційної роботи** є аналіз і дослідження методів маршрутизації для забезпечення якості трафіка.

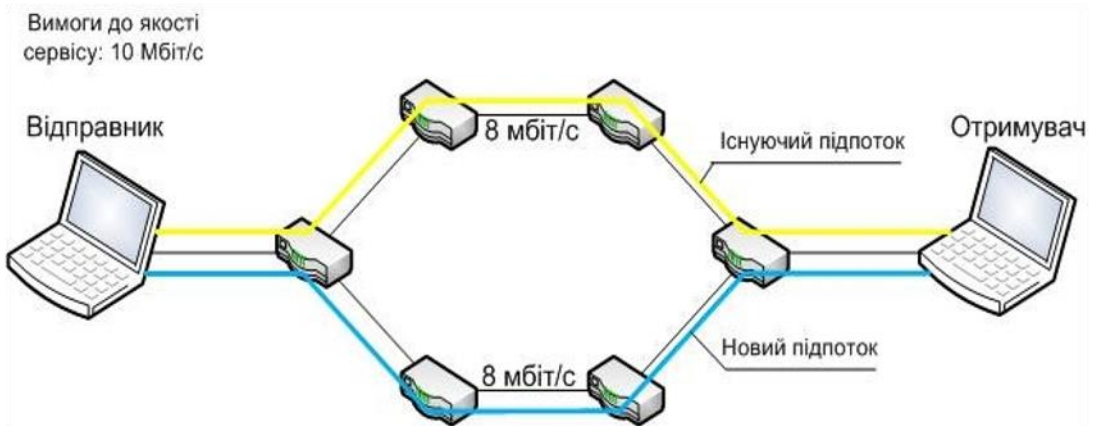
**Об'єкт дослідження:** методи багатопоточної маршрутизації

### **Завдання:**

- ❖ аналіз та дослідження робіт з ефективності багатопоточної маршрутизації;
- ❖ порівняльний аналіз ефективності статичного та динамічного методів багатопоточної маршрутизації;

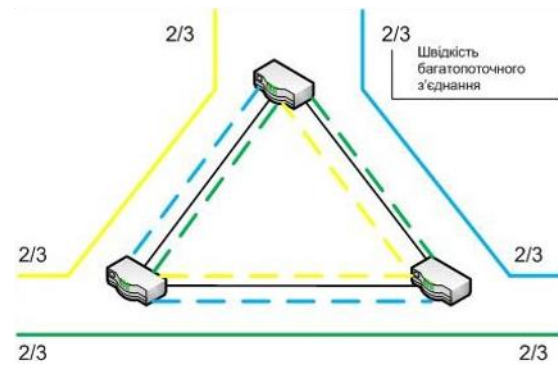
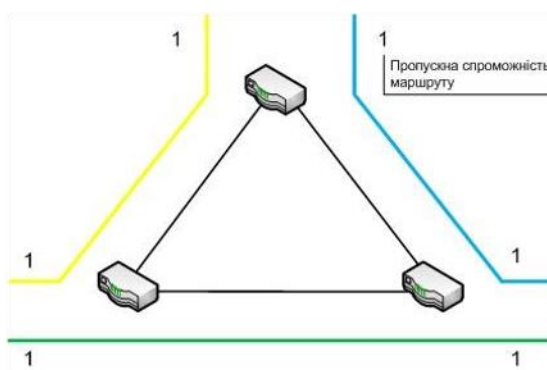
# Приклад роботи динамічного багатопоточного методу маршрутизації

3



4

## Топологія “трикутник”



методи багатопоточної маршрутизації не використовуються

використовується статичний метод багатопоточної маршрутизації

## Критерії вибору протоколів для порівняльного аналізу

5

*Клас методу.* Багатопотоковий транспортний протокол може відповідати статичному або динамічному методу

*Підтримка протоколу.* Для отримання актуальних достовірних результатів необхідно, щоб реалізація багатопоточного протоколу була відкритою та доступною, а також мала діючу підтримку, щоб відповідати поточним вимогам мереж

*Опорний транспортний протокол.* Багато протоколів використовують такі поширені транспортні протоколи, як TCP, UDP, для того щоб спростити впровадження цих протоколів в існуючу інфраструктуру мережі.

*Можливість використання довільного алгоритму управління навантаженням.* Важливо, щоб багатопотоковий протокол міг використовувати сучасні алгоритми управління навантаженням у своїй роботі.

## Порівняння багатопотокових транспортних протоколів

6

Протокол	Клас	Підтримка	Опорний транспортний протокол	Алгоритм керування перенавантаженням
MP-TCP	Статичний	Так	TCP	Будь-який
SCTP-CMT	Статичний	Ні	SCTP	Будь-який
MPQUIC	Статичний	Ні	UDP	Будь-який
BNDEEI-PTCP	Динамічний	Ні	TCP	Loss-based

## Порівняння багатопотокових транспортних протоколів

7

Головний висновок із наведеного вище, полягає в тому, що відсутні роботи, присвячені порівнянню ефективності статичного та динамічного методів багатопотокової маршрутизації, тому для проведення такого порівняння було обрано протоколи МРТСП та FDMP відповідно.

## МРТСП

8

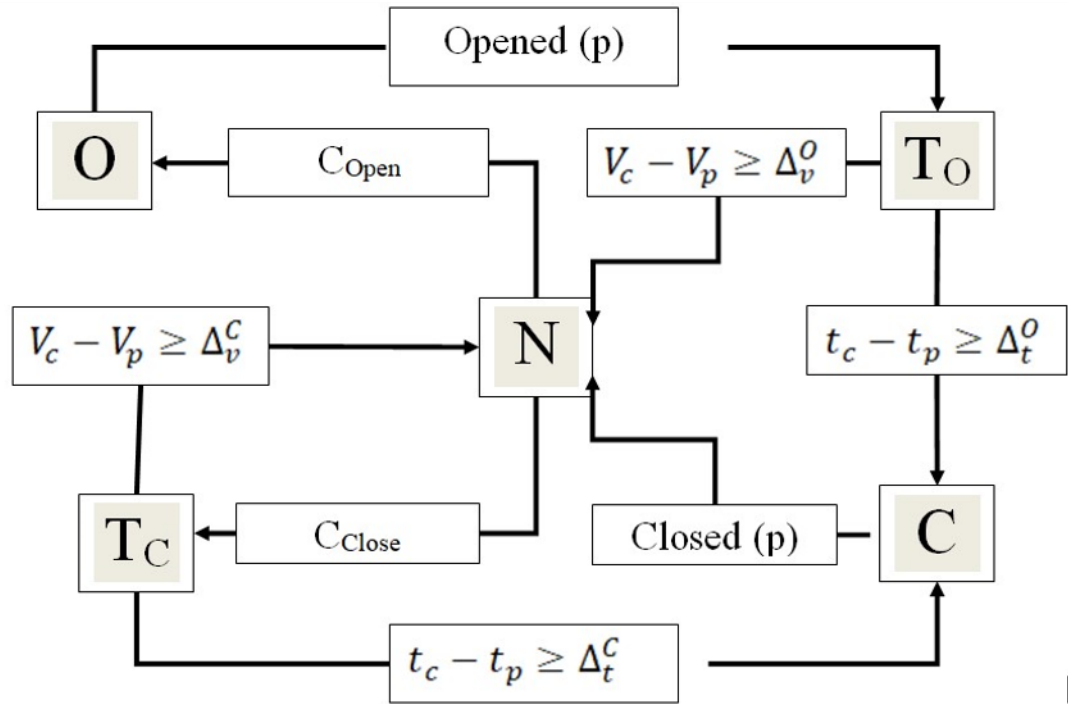
Multipath TCP (МРТСП) – це модифікація протоколу TCP, що підтримує статичний метод багатопотокової маршрутизації

Принципи спрощення інтеграції МРТСП до сучасних мереж:

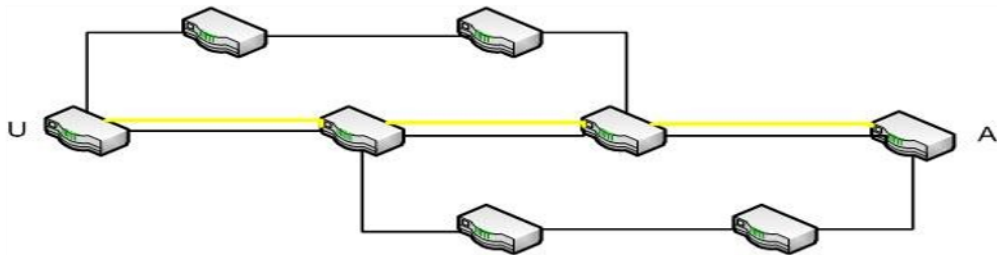
- Сегменти даних, що передаються протоколом, повинні відповідати специфікації TCP протоколу: структура заголовка сегмента даних, його максимальний розмір, а також процедура встановлення та розриву транспортного з'єднання, щоб знизити ризики несумісності з комутаційним обладнанням.
- Протокол повинен залежати від припущень відсутності фрагментації сегмента даних чи складу полів розширення заголовка сегмента (опціональних полів).
- Протокол повинен використовувати існуючий інтерфейс протоколу TCP для роботи з додатками
- Транспортний агент повинен мати можливість автоматично вибрати, за яким протоколом – TCP або МРТСП, необхідно працювати.

Протокол МРТСП досягає поставлених цілей за рахунок використання протоколу TCP з механізмом синхронізації управління вікном перевантаження для кожного зі своїх підтоків та передачі службової інформації в опціональній частині транспортного заголовка пакетів.

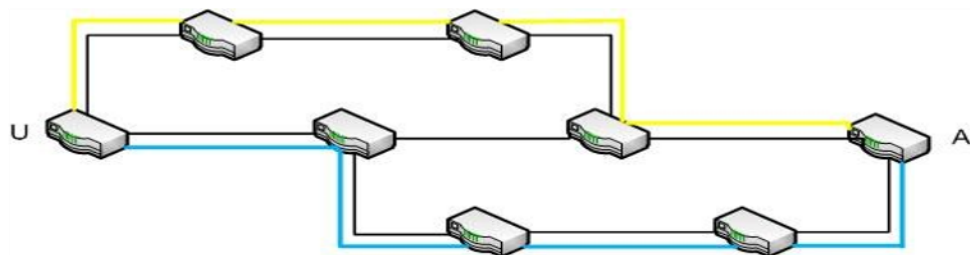
## Модель роботи транспортного агента FDMP 9



## Приклад неоптимальної роботи алгоритму GSPF 10

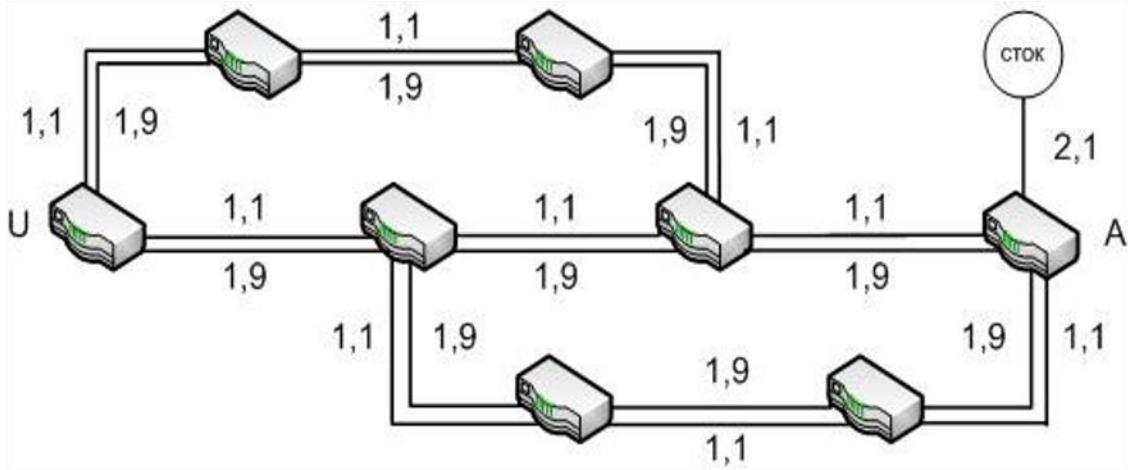


## Приклад маршрутів, що не перетинаються, не знайдених алгоритмом GSPF



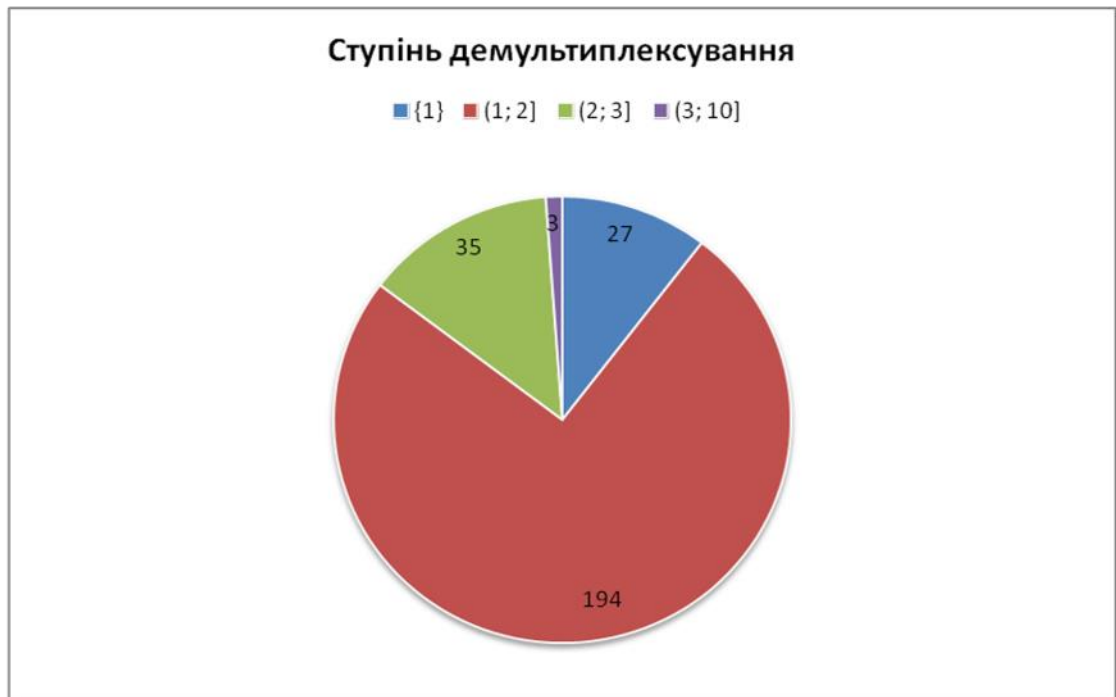
## Приклад перетвореного графа алгоритмі MCMF

11



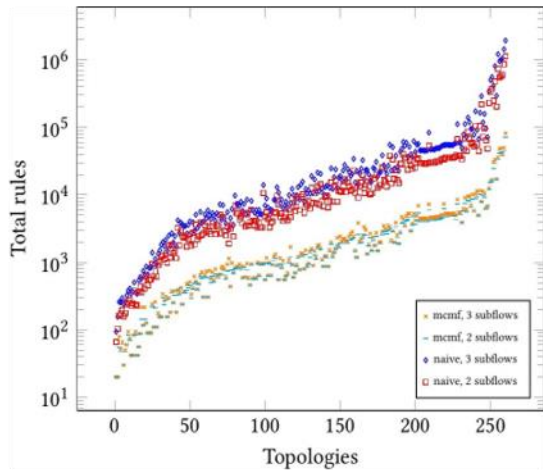
## Співвідношення кількості топологій з різним ступенем демультимплексування

12

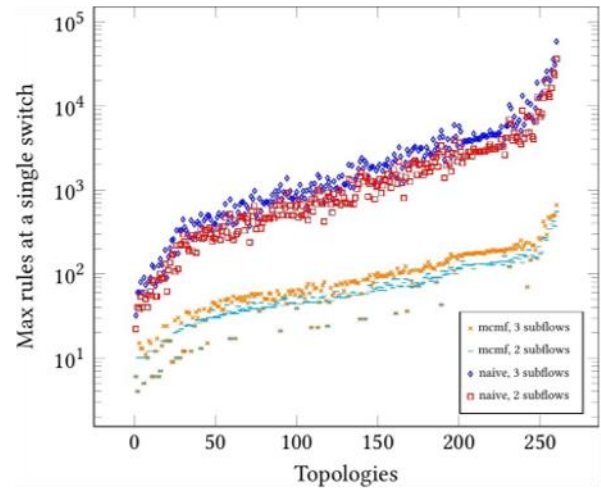


# Результати проведення оцінки кількості правил OpenFlow

13



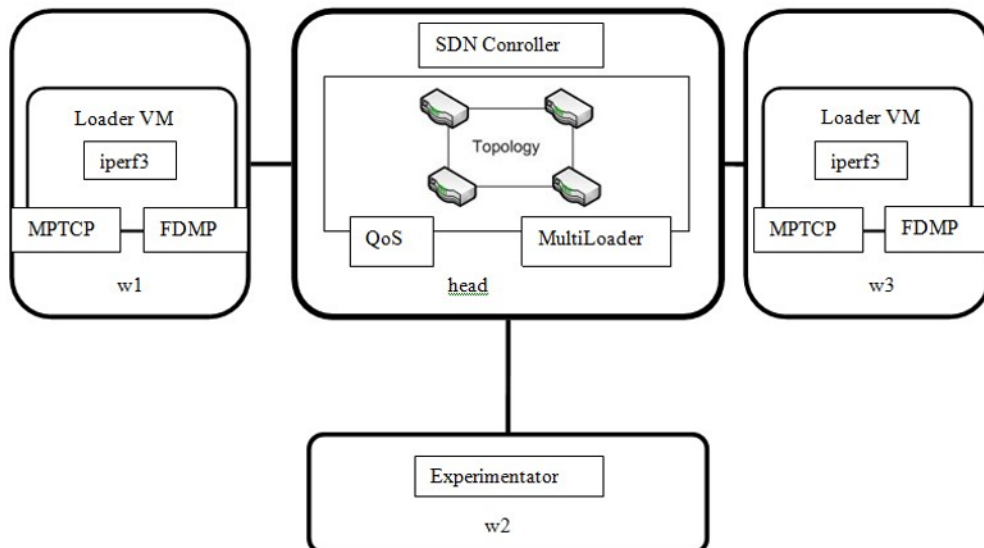
Кількість OpenFlow правил, достатня для обробки багатопоточних з'єднань між кожною парою вузлів для різних топологій



Оцінка мінімального розміру таблиці потоків протоколу OpenFlow, необхідна для встановлення правил

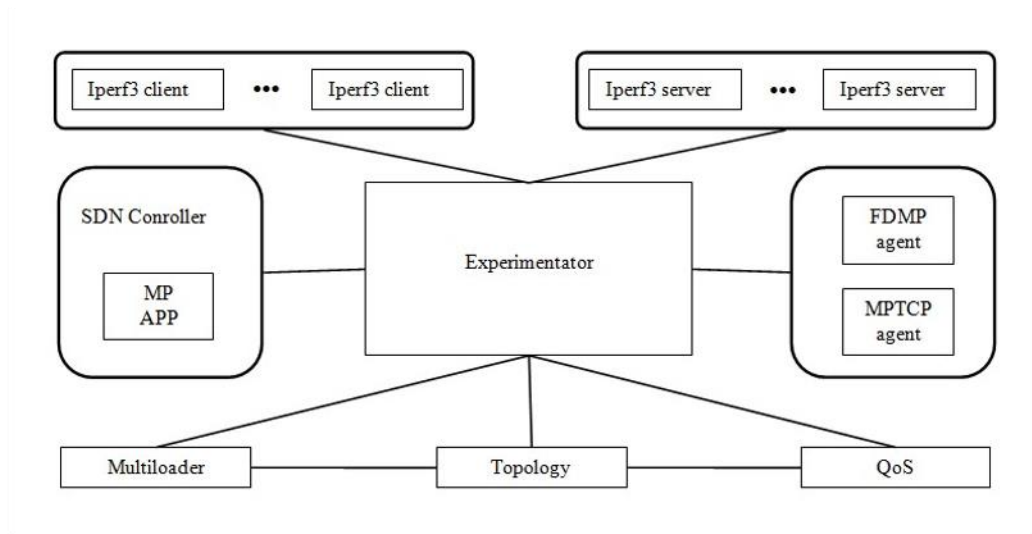
## Структура Стенду

14



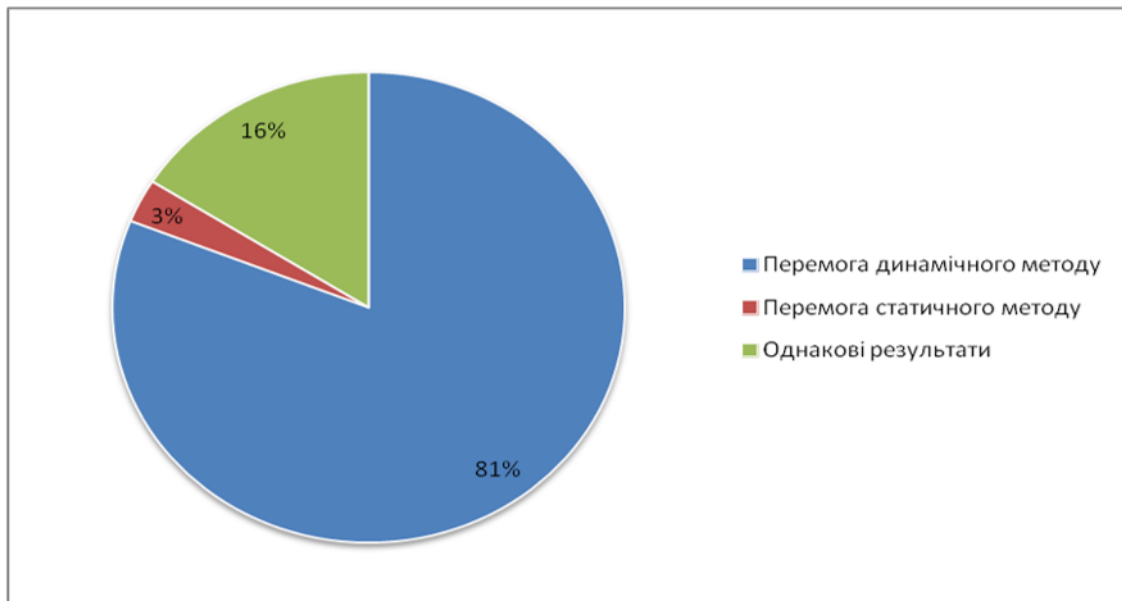
## Архітектура програмної частини експериментального стенду.

15

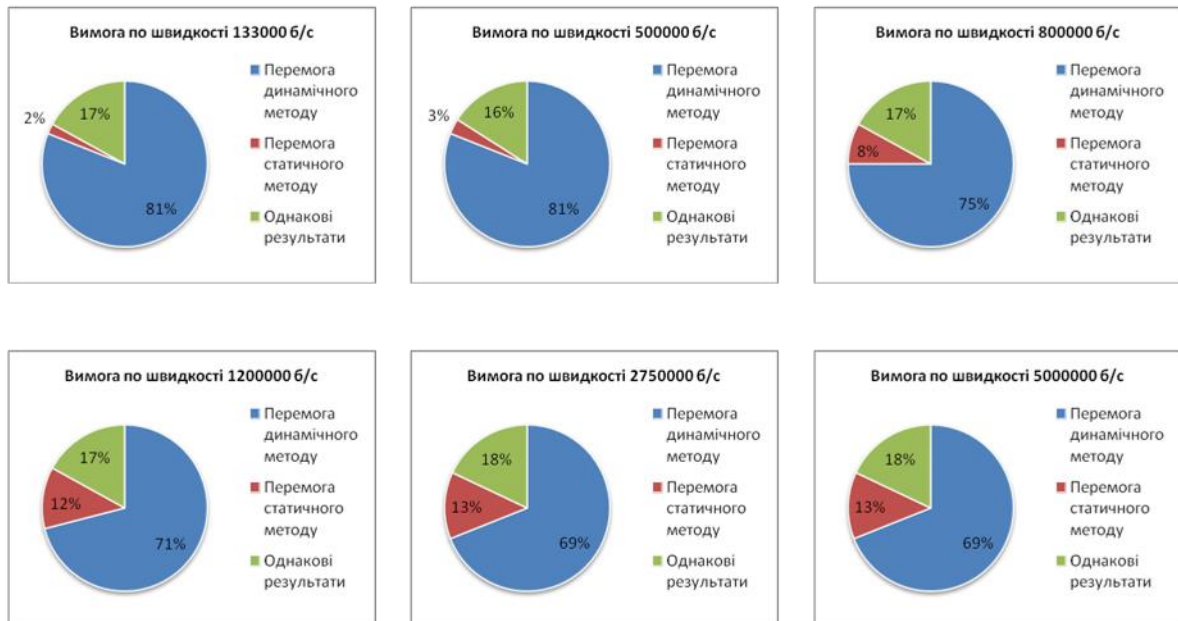


## Порівняння ефективності статичного та динамічного методів багатопотокової маршрутизації

16



## Порівняння ефективності методів багатопоточної маршрутизації для різних вимог<sup>17</sup> щодо швидкості



## Висновки

18

Існуючі роботи з аналізу ефективності багатопоточної маршрутизації переважно аналізують лише ефективність статичного методу, а аналіз динамічного методу не проводився для мереж із фіксованим зв'язком Інтернет-провайдерів. Для проведення порівняльного аналізу обрані протоколи МРТСР та FDMP як представники статичного та динамічного методів. Хоча вибрані протоколи надають ТСП інтерфейс для рівня програми, запропонована методика та результати порівняльного аналізу статичного та динамічного методів можуть стати в нагоді і для нових протоколів, заснованих на UDP протоколі, наприклад, QUIC. Так, для QUIC вже розроблена модифікація MP-QUIC, що підтримує багатопоточну маршрутизацію аналогічно МРТСР, тобто, зі статичним способом.

Для реалізації багатопоточної маршрутизації в ПКМ пропонується використовувати комбінацію реактивної та проактивної стратегій, а маршрути споріднених підтоків будувати згідно із запропонованим алгоритмом MCMF. Перелічені підходи були використані в експериментальному стенді для порівняльного аналізу. Порівняльний аналіз статичного та динамічного методів багатопоточної маршрутизації показав перевагу динамічного методу, коли кількість задоволених потоків не менша, ніж для статичного методу, з ймовірністю не менше 95%, при цьому статичний метод добре працює тільки на невеликих топологіях з кількістю вузлів менше 36 і невеликою часткою полкусів

## ДОДАТОК Б

### Реалізація стенду

Деталі реалізації підсистем Стенду, описаного в розділі 2.2:

Підсистема Topology реалізована мовою Python3 з використанням програмних OpenFlow комутаторів OpenVSwitch та віртуальних ліній зв'язку VETH, що підтримуються операційною системою на базі ядра Linux.

Підсистема QoS реалізована мовою Python3 з використанням зовнішньої утиліти tc (traffic control), що дозволяє налаштовувати параметри віртуальних ліній зв'язку VETH.

Підсистема Multiloader реалізована мовою Python3 за допомогою програмного OpenFlow комутатора та віртуальних ліній зв'язку VETH. Параметри якості сервісу не налаштовуються на лініях зв'язку, що ведуть до імітаційної моделі мережі, що дозволяє передати дані до імітаційної моделі з максимальною швидкістю для операційної системи, щоб дослідити ефективність багатопоточних протоколів усередині мережі Інтернет-провайдера.

Підсистема Iperf3 реалізована мовою C і заснована на однойменній утиліті до якої було додано функціональність старту потоків згідно з заданим розкладом.

Підсистема FDMP agent реалізована мовою C як частина ядра операційної системи на базі Linux версії 4.14.91.

Реалізацію підсистеми MPTCP agent взято з відкритого репозиторію.

Підсистема MP App реалізована мовою C++ як додаток користувача вітчизняного контролера ПКМ Runos ,що підтримує роботу за протоколом OpenFlow версії 1.3.

Підсистема експериментатора реалізована мовою Python3. Оскільки інші підсистеми можуть бути на окремих серверах/віртуальних машинах, то звернення однієї підсистеми до іншої використовується бібліотека fabric для

віддаленого виклику методів, які реалізують інтерфейси вищеописаних підсистем.

Апаратна частина Стенду представлена Рисунку А.1. Стенд складається із 4 серверів: w1, w2, w3, head. Сервери w1, w2, w3 мають процесор Intel Xeon E5-2667 v4 (8 ядер, базова частота 3,20 GHz), 32 ГБ оперативної пам'яті. Сервер head має два процесори Intel Xeon E5-2650 v4 (загалом 24 ядра, базова частота 2,20 GHz), 64 ГБ оперативної пам'яті. Сервер head з'єднаний з серверами w1 і w3 чотирма каналами з пропускною здатністю 10Гбіт/с, а з сервером w2 одним каналом з пропускною здатністю 10Гбіт/с.

На серверах w1, w3 розташовані по 4 віртуальні машини з FDMP агентом/МРТСП агентом та Iperf3. Ці сервери відповідають за генерацію потоків даних, який відправляється в мережу згідно з багатопоточним протоколом FDMP або МРТСП. Трафік від різних віртуальних машин прямує до сервера head через різні лінії зв'язку.

На сервері head працюють підсистеми Topology, QoS, Multiloader, а також контролер ПКС Runos із програмою MP App. Сервер head відповідає за імітаційне моделювання Інтернет-провайдера.

На сервері w2 працює підсистема Experimentator, яка відповідає за запуск експериментів та збирання результатів.

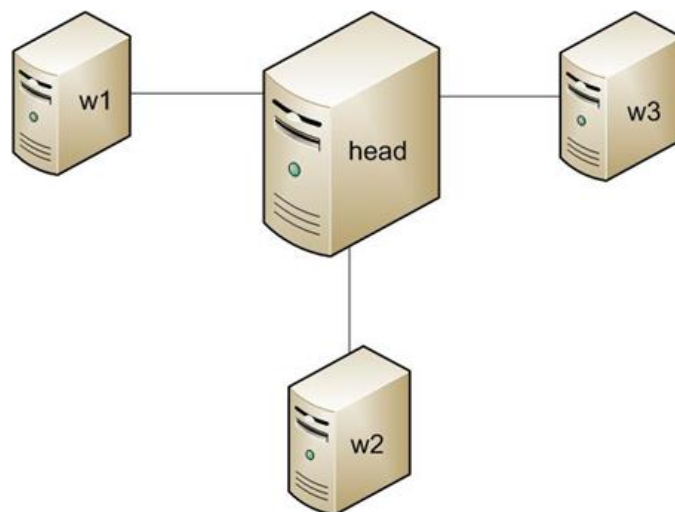


Рисунок Б.1 – Схема апаратної частини експериментального стенду.