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

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

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

Магістр
(назва освітнього ступеня)

на тему: Методи та засоби керування пріоритизацією трафіку в комп'ютерних мережах з використанням SDN

Виконав: студент(ка) 6 курсу, групи СІм-61
спеціальності 123 «Комп'ютерна інженерія»
(шифр і назва спеціальності)

(підпис) Дячук О. А. (прізвище та ініціали)

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

Нормоконтроль (підпис) Луцик Н.С. (прізвище та ініціали)

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

Рецензент (підпис) Стоянов Ю.М. (прізвище та ініціали)

Тернопіль
2023

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

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

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

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

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ

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

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

студенту *Дячуку Олегу Андрійовичу*
(прізвище, ім'я, по батькові)

1. Тема роботи *Методи та засоби керування пріоритизацією трафіку в комп'ютерних мережах з використанням SDN*

Керівник роботи *Жаровський Руслан Олегович, к.т.н.*
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «*01*» *грудня* *2023* року № *1132*

2. Термін подання студентом завершеної роботи *26.12.2023 р.*

3. Вихідні дані до роботи *Методика управління трафіком, технологія SDN, алгоритм HTB*

4. Зміст роботи (перелік питань, які потрібно розробити)
Вступ. 1. Огляд теоретичних основ в сфері керування пріоритизацією трафіку
2. Аналіз методів і алгоритмів керування трафіком в SDN мережах
3. Апробація запропонованих алгоритмів розподілу трафіку в SDN мережах
4. Охорона праці та безпека в надзвичайних ситуаціях
Висновки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)
1. Актуальність і мета дослідження.
2. Задачі дослідження, об'єкт і предмет, наукова новизна і практична цінність дослідження.
3. Методи організації пріоритизації трафіку в комп'ютерних мережах
4. Алгоритм планування Hierarchical Token Bucket
5. Модифікаований алгоритм планування Hierarchical Token Bucket
6. Перевірка ефективності модифікованого алгоритму HTB
7. Імітаційне дослідження алгоритму підтримки низькопріоритетних сервісів.
8. Експериментальне дослідження
9. Висновки

6. Консультанти розділів роботи

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

7. Дата видачі завдання 20.11.2023

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	*Аналіз сучасних методів і технологій в сфері пріоритизації і контролю трафіку*	*01.12.2023*	*виконано*
2	*Аналіз методів і алгоритмів керування трафіком в SDN мережах*	*03.12.2023*	*виконано*
3	*Розробка модифікованого алгоритму керування трафіком*	*10.12.2023*	*виконано*
4	*Апробація запропонованих алгоритмів розподілу трафіку в SDN мережах*	*16.12.2023*	*виконано*
5	*Охорона праці та безпека в надзвичайних ситуаціях*	*18.12.2023*	*виконано*
6	*Оформлення пояснювальної записки і графічного матеріалу*	*19.12.2023*	*виконано*
7	*Попередній захист кваліфікаційної роботи магістра*	*20.12.2023*	*виконано*
8	*Захист кваліфікаційної роботи магістра*	*26.12.2023*	

Студент _____ *Дячук О. А.*
(підпис) (прізвище та ініціали)

Керівник роботи _____ *Жаровський Р.О.*
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Методи та засоби керування пріоритизацією трафіку в комп’ютерних мережах з використанням SDN // Кваліфікаційна робота магістра // Дячук Олег Андрійович // ТНТУ, Комп’ютерна інженерія, група СІм-61 // Тернопіль, 2023 // с. – 84, рис. – 32, табл. – 11, бібліогр. – 28.

Ключові слова: SDN, HTB, пріоритет, планування черг.

У кваліфікаційній роботі магістра проведено огляд метрик що впливають на пріоритизацію трафіку в КМ, а також проведенений огляд методів і засобів керування пріоритизацією трафіку.

Розглянуто поняття доступності з точки зору комп’ютерних мереж. Наведені алгоритми пріоритизації і керування потоком. Розроблено модифікований алгоритм планування черг і алгоритм обробки низькопріоритетних сервісів.

Проведено імітаційне моделювання і експериментальне дослідження запропонованих алгоритмів.

ABSTRACT

Methods and tools for traffic prioritization control in computer networks using SDN // Master graduation thesis // Diachuk Oleh Andriiovych // TNTU, computer engineering, group СІм-61 // Ternopil, 2023 // p. – 84, fig. - 32, tab. - 11, bibliography. - 28.

Keywords: SDN, HTB, priority, queue planning.

In the master's thesis, the metrics affecting the traffic prioritization in KM were studied, as well as a review of the methods and means of traffic prioritization management was carried out.

The concept of availability from the point of view of computer networks is considered. Algorithms for prioritization and flow control are presented. A modified algorithm for queue planning and an algorithm for processing low-priority services have been developed.

Simulation modeling and experimental research of the proposed algorithms were carried out.

ЗМІСТ

ПЕРЕЛІК ОСНОВНИХ УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ І СКОРОЧЕНЬ

НТВ (англ. Hierarchical Token Bucket) – алгоритм «ієрархічне відро маркерів»

MTU (англ. Maximum Transmission Unit) – максимальна одиниця передачі

QoS (англ. Quality of Service) - якість обслуговування

SDN (англ. Software Defined Network) - програмно-визначена мережа

SLA (англ. Service Level Agreement) – угода про рівень обслуговування

КТМ – корпоративна телекомунікаційна мережа

КЗ – канал зв'язку

ПЗ – програмне забезпечення

МДЧ – максимальний директивний час

ПКТ – пріоритизація та контроль трафіку

ВСТУП

Актуальність теми. Корпоративні мережі дедалі частіше стають стримуючим чинником розвитку обчислювальної інфраструктури підприємства. Причинами є зростаючий обсяг трафіку та його різноманітність. У мережі передається різнорідний трафік, що включає як звичайні дані так і сигнали керування, голосові, відео дані, які потребують особливого першочергового пріоритету при їх передачі. Проблему підсилює перехід користувачів на дистанційну форму зайнятості та використання мобільних пристроїв у своїй роботі.

Тому актуальним завданням є пошук можливих засобів підвищення якості роботи мережі за рахунок внутрішніх механізмів розподілу трафіку. Одним із таких механізмів є оптимізація управління трафіком, яка може забезпечити ефективну роботу різнорідних систем. Ефективність використання ресурсів каналу пакетної передачі завжди була актуальною завданням, та її важливість зросла останніми роками у зв'язку з появою дедалі жорсткіших вимог до якості обслуговування різнорідного трафіку, особливо IP -телефонії і відеозв'язку.

При цьому, сучасні корпоративні телекомунікаційні мережі (КТМ) характеризуються використанням дедалі більше мережевих сервісів. Фундаментальною проблемою тут стає підвищення якості обслуговування за нових умов, які продиктовані потребами прикладного рівня. Критичною характеристикою багатьох важливих функціонування КТС мережевих сервісів є час відгуку, проте сучасні протоколи неспроможна забезпечити цю характеристику, оскільки містять даного критерію. Тому для того, щоб прикладний рівень ефективно використовував мережну інфраструктуру, потрібно впроваджувати додаткові критерії якості, такі як доступність.

Існують різні методи керування технологією QoS. У цій роботі будуть розглянуті такі методи, як ручне та автоматизоване налаштування, програмно-керовані мережі (англ. software-defined networking, SDN).

На сьогоднішній день, мережеві провайдери та численні IT організації використовують SDN-мережі для оптимізації управління мережевою

інфраструктурою та забезпечення високого рівня керованості, захищеності та надійності мережі. Основною концепцією SDN є розділення рівня управління мережею (control plane) та рівня передачі трафіку (forwarding plane). У звичайних мережах ці функції неможливо відокремити, оскільки вони реалізовані в одному пристрої на основі єдиного набору системної логіки.

Питанням організації, управління та масштабованості SDN присвячені роботи провідних зарубіжних вчених Bhandarkar S., Hu J., Oliveira AT, Mondal A., Tuncer D. Проблема забезпечення якості обслуговування в телекомунікаційних мережах досліджувалась у працях таких учених, як Devera M., Balan D., Domanska J., Stanwood KL, Keith S., C. Douligeris, Vegesna S., Ma Q.

В даній роботі буде проведено дослідження даної технології і шляхом аналізу результатів розробленого ПЗ планується виявити переваги та недоліки запропонованого методу автоматизації налаштування якості обслуговування корпоративної мережі для кращого розподілу трафіку.

Метою кваліфікаційної роботи полягає у підвищенні ефективності обслуговування трафіку корпоративних програмно-керованих телекомунікаційних мереж, що полягає у підвищенні показника доступності як компонентів, так і мережі загалом за рахунок розробки та впровадження нових алгоритмів управління SDN.

Задачі кваліфікаційної роботи:

1.Розкрити актуальність досліджуваної теми шляхом критичного аналізу існуючих систем керування трафіком;

2.Розглянути існуючі методи та моделі управління пріоритизацією трафіку мережі та виявити їх недоліки;

3.Запропонувати та описати новий метод управління пріоритизацією трафіку мережі;

4.Розробити алгоритм планування черг передачі даних що дозволить оптимізувати використання пропускної здатності каналу зв’язку та забезпечувати максимальну доступність підтримуваних сервісів

5.Провести тестування запропонованих методів.

Відповідно до цілей та завдань кваліфікаційної роботи визначено її об'єкт та

предмет.

Об’єкт дослідження: програмно – апаратні засоби управління пріоритизацією трафіку в комп’ютерних мережах.

Предмет дослідження: є моделі та алгоритми забезпечення якості обслуговування трафіку на основі доступності вузлів та каналів зв'язку в SDN мережах.

Методи дослідження: апарату теорії ймовірностей, теорії графів, теорії систем масового обслуговування, математичної статистики, математичного аналізу, методів комп'ютерного моделювання та технології об'єктно-орієнтованого програмування. Для практичної перевірки працездатності запропонованих алгоритмів використовувалося розроблене програмне забезпечення. Метод візуалізації даних, що дозволяє наочно представляти отримані результати дослідження.

Наукова новизна дослідження полягає у розробці методики керування пріоритизацією трафіку, яка ґрунтується на технології SDN, що дозволить підвищити ефективність передачі пріоритетного трафіку за рахунок планування черг передачі даних у програмно-конфігурованій телекомунікаційній мережі на основі модифікації підходу НТВ.

Теоретична значущість полягає у розробці вдосконаленого способу передачі пріоритетного трафіку.

Практичне значення результатів кваліфікаційної роботи полягає у можливості застосування розробленої методики для автоматизації та оптимізації процесу налаштування мережного обладнання.

Публікації. Результати дослідження апробовано на XI науково-технічній конференції Тернопільського національного технічного університету імені Івана Пулюя «Інформаційні моделі, системи та технології» (13-14 грудня 2023 року) у вигляді тез конференцій.

Дячук О.А., Жаровський Р.О. Використання SDN для оптимізації передачі даних в комп’ютерних мережах. Матеріали XI науково-технічної конференції Тернопільського національного технічного університету імені Івана Пулюя

«Інформаційні моделі системи та технології» (13-14 грудня 2023 року). Тернопіль: ТНТУ. 2023. С. 149-150.

Дячук О.А., Жаровський Р.О Управління потоком за критеріями доступності. Матеріали XI науково-технічної конференції Тернопільського національного технічного університету імені Івана Пулюя «Інформаційні моделі системи та технології» (13-14 грудня 2023 року). Тернопіль: ТНТУ. 2023. С. 151.

Структура роботи. До складу кваліфікаційної роботи магістра входить розрахунково-пояснювальна записка та графічний матеріал. Розрахунково-пояснювальна записка містить вступ, 4 розділи, загальні висновки, список використаної літератури і додатки. Обсяг роботи: розрахунково-пояснювальна записка – 84 арк. формату А4, графічна частина – 8 аркушів формату А1.

РОЗДІЛ 1
ОГЛЯД ТЕОРЕТИЧНИХ ОСНОВ В СФЕРІ КЕРУВАННЯ ПРІОРИТИЗАЦІЄЮ ТРАФІКУ

1.1. Метрики пріоритизації трафіку в комп’ютерних мережах

Корпоративній мережі, так само як і іншим типам комп'ютерних мереж, висуваються різноманітні вимоги. Однією з найважливіших є забезпечення користувачам швидкого та надійного доступу до ресурсів всіх комп'ютерів, що об'єднуються в мережу. Вирішення цього основного завдання передбачає врахування вимог до продуктивності, надійності, безпеки, керованості, сумісності та масштабованості.

Стандарт ITU-T Y.1540 визначає терміни та визначення для оцінки якості обслуговування (QoS) в мережах пакетної комутації, зокрема в Інтернеті [18]. Основні аспекти, що враховуються в Y.1540, включають параметри і метрики, що оцінюють якість обслуговування, такі як затримка, втрата пакетів і джиттер (різниця в часі між пакетами).

Цей стандарт визначає методології для вимірювання та оцінки різних аспектів QoS у мережах пакетної комутації, щоб забезпечити ефективну передачу різноманітних видів трафіку, таких як голос, відео та дані. Це допомагає операторам мережі та постачальникам послуг оцінювати та забезпечувати високий рівень обслуговування для своїх клієнтів. Розглянемо більш детально ці метрики.

Продуктивність мережі (throughput) [26] визначається кількістю інформації, що передається по мережі за одиницю часу. Важливо відзначити, що продуктивність мережі не тотожна максимальній пропускній здатності, відомій як ширина смуги пропускання (bandwidth).

Втрата пакетів вказує, скільки з відправлених джерелом пакетів дійшло адресата. Як правило, як справляється із втратами залежить від додатка. У випадку з веб-програмами, які використовують протокол TCP, пакет відправляється заново, а у разі телефонної розмови (протокол UDP) — пакет відкидається.

Коефіцієнт втрати пакетів (Packet Loss Ratio, PLR) є вимірюваною величиною, яка вказує на частку втрачених пакетів від загальної кількості переданих пакетів у мережі передачі даних. PLR виражається у відсотках або у десятковому вигляді від 0 до 1 [21].

Мережеві пакети є блоками даних, які передаються через комп'ютерні мережі. Втрата пакетів може виникнути з різних причин, таких як конгестія мережі, недостатня пропускна здатність, помилки у передачі даних або інші фактори.

PLR є важливим показником для оцінки якості мережі, особливо у випадку вимогливих застосувань, таких як інтернет-телефонія (VoIP) або відеоконференції. Великий коефіцієнт втрати пакетів може призводити до погіршення якості передачі даних, особливо у випадках, коли втрати стають помітними для користувача, наприклад, у вигляді збоїв в аудіо або відео.

Час передачі пакета (Packet Transfer Delay, PTD) - це час, який потрібно для того, щоб пакет даних пройшов від відправника до одержувача в комп'ютерних мережах. PTD включає в себе кілька компонентів:

–затримка передачі (Transmission Delay): Це час, який потрібен для передачі бітів пакета по мережі. Залежить від пропускної здатності каналу передачі даних;

–затримка обробки вузла (Node Processing Delay): Це час, який вузол мережі витрачає на обробку заголовку пакета та вирішення інших завдань;

–затримка в черзі (Queueing Delay): Це час, який пакет проводить у черзі на вузлі, якщо він не може бути відразу оброблений;

–затримка поширення (Propagation Delay): Це час, який потрібен сигналу (або пакету) для того, щоб пройти від відправника до одержувача через фізичну мережу. Залежить від швидкості поширення сигналу в середовищі передачі.

Отже, PTD представляє собою суму цих затримок і визначає загальний час, який необхідний для передачі пакета від відправника до одержувача. Поняття PTD є важливим для оцінки ефективності та якості мережі, особливо в тих випадках, коли важлива точність передачі даних, таких як у голосовому або відеозв'язку.

Найчастіше з метою оцінки зайнятості каналу використовують кругову затримку RTT (Round Trip Time) (рис. 1.1). Це інтервал часу між відправкою пакета

та закінченням його обробки на приймаючій стороні.

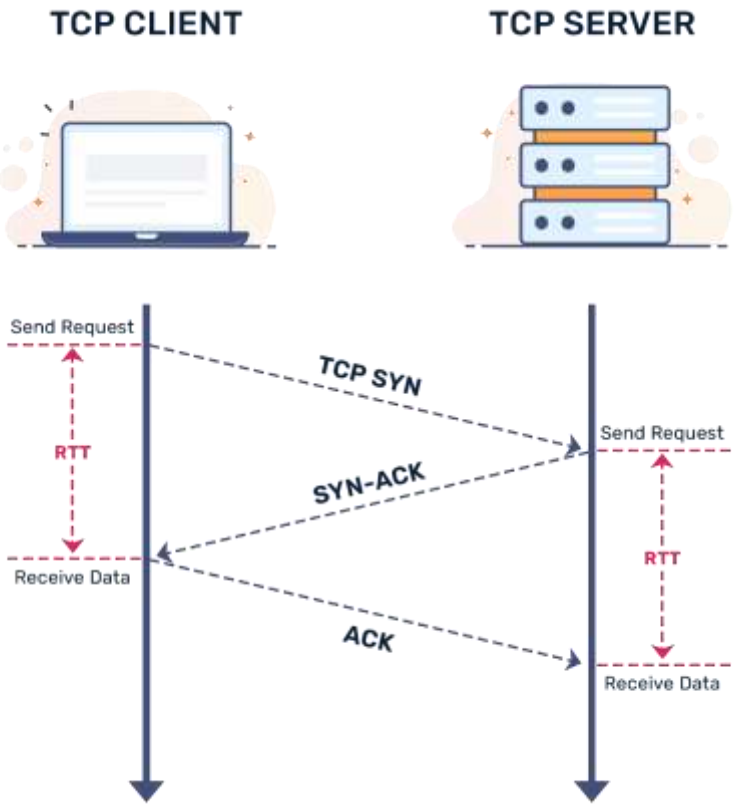


Рис.1.1. Виникнення затримки в мережі

RTT можна виміряти за допомогою будь-якого навіть самого простішого програмного засобу. Наприклад використавши команду ping: можна побачити, коли запускаємо ping test.b-cdn.net:

```
ping test.b-cdn.net -c 4
PING test.b-cdn.net (138.199.57.151): 56 data bytes
64 bytes from 138.199.57.151: icmp_seq=0 ttl=57 time=14.524 ms
64 bytes from 138.199.57.151: icmp_seq=1 ttl=57 time=15.309 ms
64 bytes from 138.199.57.151: icmp_seq=2 ttl=57 time=15.215 ms
64 bytes from 138.199.57.151: icmp_seq=3 ttl=57 time=15.992 ms

--- test.b-cdn.net ping statistics ---
4 packets transmitted, 4 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 14.524/15.260/15.992/0.520 ms
```

Рис.1.2. Приклад визначення RTT за допомогою команди ping

В останній стрічці (це може відрізнятися в різних версіях ping): відображена інформація стосується RTT. Час (тут у мілісекундах) – це загальний час, витрачений на «запит». Він включає в себе час, потрібний для SYNchronization, час необхідний для надсилання зазначеного пакету (в основному цю метрику відстежують за допомогою протоколу ICMP (Internet Control Message Protocol) і час потрібний для підтвердження та отримання відповіді. Коли виникає неоптимальна затримка, це може бути викликано різними проблемами, наприклад:

– депріоритизація ICMP. Маршрутизатори іноді налаштовані так, що вони не відповідають після X запитів ICMP; тобто обмеження швидкості;

– географічна відстань. Це означає, що вашому пакету ICMP може знадобитися перестрибнути, скажімо, 10 мереж, перш ніж досягти пункту призначення;

– погані умови мережі. Спроба перевірити продуктивність мережі на хості на повільному 3G-з’єднанні; або мережа, яка сильно перевантажена;

– проблеми з сервером. Сервери іноді можуть відчувати періоди надзвичайно високого навантаження; це означає, що залишилося небагато циклів ЦП для надсилання підтвердження для ваших ping запитів (це впливає як на затримку, так і на час зворотного зв’язку).

Варіація затримки пакета (packet delay variation, PDV) під час передачі послідовних пакетів називається джиттером. Знову ж таки в залежності від додатку визначається оптимальний рівень даної метрики. Більшості додатків достатньо, щоб пакет був доставлений і неважливо була там затримка. Але для тієї ж IP - телефонії це відіграє важливу роль.

Джиттер, або варіація затримки пакета, проявляється в нерегулярних інтервалах прибуття послідовних пакетів до одержувача. У випадку систем IP-телефонії це може призводити до спотворень звуку і зробити його нерозбірливим.

Джиттер призводить до специфічних порушень передачі звуку, що звучать як тріск і клацання [13]. Строго кажучи, це відхилення від справжньої періодичності імовірно періодичного сигналу (рис. 1.3). Очікується, що голосові пакети періодично надходять до місця призначення з постійною швидкістю. В ідеалі пакет

має надходити кожні X мілісекунд, і X має бути постійним протягом часу. Однак у реальних мережах X ніколи не є постійним. Незначна зміна X до +/- 30 мілісекунд є прийнятною та може бути виправлена за допомогою jitter buffer. Якщо X змінюється занадто сильно, jitter buffer не зможе компенсувати цю зміну, і пакети почнуть скидатися.

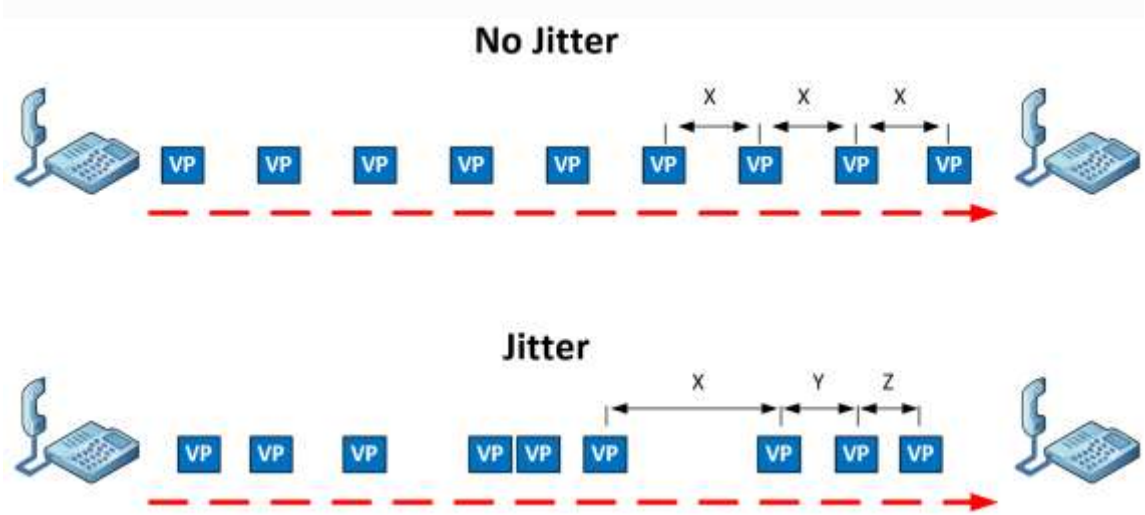


Рис. 1.3. Приклад виникнення джиттера при VoIP зв’язку

У будь-якій мережі завжди спостерігається джиттер. Однак зазвичай джиттер дуже часто є результатом перевантаженості мережі, яка затримує одні пакети більше, ніж інші. Якщо механізми пріоритизації трафіку не налаштовані належним чином на маршрутизаторах і комутаторах, будь-який рівень перевантаження мережі призведе до збільшення джиттера.

Частина мережевих сервісів в комп’ютерних мережах є критичними до затримок та/або інших показників якості обслуговування (Quality of Service, QoS). Серед чутливих до затримок сервісів можна назвати IP-телефонію, відеоконференцзв'язок, VDI (Virtual Desktop Infrastructure – інфраструктура віртуальних робочих столів), які набули ще більшого поширення у зв'язку з масовим переходом на віддалений режим роботи під час пандемії COVID-19.

Причому, сервіси, в яких критичним параметром є час відгуку, не обмежуються перерахованими вище: можна також виділити чат-боти для здійснення

торгівлі, в тому числі на біржових майданчиках, хмарні сервіси, серед яких набирає популярності TaaS (Testing-as- a-Service – тестування як сервіс), що працює в режимі «м'якого» реального часу, послуги АСУ та інші.

Функціонування організації залежить від доступності даних сервісів, так як вони забезпечують потреби бізнесу у комунікації, інструментарії на вирішення повсякденних завдань, автоматизації бізнеспроцесів тощо. Тому постачальники телекомунікаційних послуг повинні гарантувати, що надані ними послуги досягатимуть необхідного цільового показника доступності. Саме для цього застосовуються механізми забезпечення якості обслуговування, що відображено, в тому числі, у згаданій вище рекомендації ITU-T Y.1540, а також у рекомендації ITU-T Y.1541 [19], яка дає наступне обґрунтування необхідності застосування механізмів QoS: споживачі потребують таких рівнів мережевих показників якості, які у поєднанні з їх хостами, кінцевим обладнанням та іншими пристроями забезпечують задовільну підтримку їх додатків.

Тому методи забезпечення якості обслуговування і розподілу трафіку в корпоративних програмно-керованих мережах на основі доступності вузлів та каналів зв'язку становлять предмет даного дослідження

1.2. Методи організації пріоритизації трафіку в комп’ютерних мережах

Для підвищення продуктивності мережі без зростання пропускної здатності транспортної мережі було створено технологію «Якість обслуговування» (Quality of Service, QoS). Якість обслуговування використовує розподіл за категоріями та призначення пріоритетів трафікам, що дозволяє гарантувати даним з більшим пріоритетом кращі умови передачі через мережу, незалежно від вимог до пропускної здатності потоків менш важливих додатків [6].

Існує три моделі забезпечення QoS:

1) Best Effort: не гарантує якість передачі трафіку, всі потоки рівні.

2) IntServ: гарантує якість кожного потоку, заздалегідь резервує ресурси від джерела до одержувача.

3) DiffServ: за визначення якості відповідає кожен вузол, яким йде передача, відсутня резервування.

В даний час у мережах найчастіше використовуються механізми DiffServ. Оскільки механізми IntServ вимагають величезних ресурсів від обчислювальних процесорів видаючих пристроїв, у зв'язку з постійним завчасним резервуванням ресурсів.

Модель диференційованого обслуговування визначає забезпечення QoS на основі чітко визначених компонентів, що комбінуються з метою надання необхідних послуг. Архітектура DiffServ передбачає наявність класифікаторів та формувачів трафіку на кордоні мережі, а також підтримку функції розподілу ресурсів у ядрі мережі з метою забезпечення необхідної політики. Розділяє трафік на класи, вводячи кілька рівнів QoS [10].

Дана модель управління трафіком гарантує, що трафік більшу частину часу отримуватиме адекватне обслуговування, можливо, з деяким ступенем диференціації. У разі перевантаження потоки адаптуватимуть свій трафік до наявних ресурсів і продовжуватимуть обслуговуватися, хоча, можливо, з нижчою якістю. Перевагою даного рішення є більш висока загальна ефективність, так як воно дозволяє передавати більшу кількість потоків простіше, з мінімальною підтримкою сигналізації та за допомогою простих механізмів організації шляхів передачі даних [12].

1.3. Метод реалізації якості обслуговування шляхом ручного налаштування

Метод ручного налаштування якості обслуговування полягає в поетапному монотонному ручному налаштуванні адміністратором мережі правил обслуговування трафіку на кожному мережному пристрої.

Зазвичай мережеві адміністратори становлять у електронних таблицях правила QoS з урахуванням аналізу вимог якості обслуговування клієнтів. Потім основі цих правил створюють типові конфігурації для мережного устаткування. І нарешті, дані конфігурації вручну вносяться на мережне обладнання, що вимагає налаштування.

Це тривалий і неефективний метод, пов'язаний з помилками через людський фактор, який може призвести до деградації сервісів та додаткових фінансових витрат. Але він відносно дешевий, тому що не передбачає купівлю, використання та експлуатацію нового дорогого обладнання за допомогою SDN, або ПЗ для автоматизації процесу налаштування мережевого обладнання.

1.4. SDN метод

Головна ідея програмно-керованої мережі (SDN) полягає у розділенні функцій передачі трафіку і функцій керування [9].

Необхідний аналіз пакетів, зміни інформації в пакетах та правила пересилання покладаються на контролер. Самі пристрої дотримуються інструкцій контролера, навантаження на обчислювальні ресурси значно знижується. Досягається централізація логіки управління мережею, що дозволяє конфігурувати мережу як єдине ціле і значно полегшує експлуатацію мережі.

Відкрита архітектура SDN забезпечує сумісність із кількома постачальниками. API -інтерфейси підтримують широкий спектр програм, включаючи хмарне оркестрування та важливі для бізнесу мережеві програми. Крім того, інтелектуальне програмне забезпечення може керувати обладнанням різних виробників за допомогою відкритих програмних інтерфейсів, таких як OpenFlow. Нарешті, з SDK інтелектуальні мережеві служби та програми можуть працювати у загальному програмному середовищі [14].

Архітектура мереж SDN поділяється на три рівні:

1) Рівень додатків.
2) Рівень контролю.
3) Рівень передачі.

Ресурси рівня передачі виконують мережеві функції, такі як передача та обробка даних, і їх поведінка управляється рівнем контролю. Взаємодія контролера з рівнем передачі даних можлива за допомогою інтерфейсів та спеціалізованих протоколів, таких як OpenFlow, що забезпечують взаємодію з мережевими

пристроями. З іншого боку, контролер надає стандартизовані програмні інтерфейси API, які дозволяють створювати програми для управління мережею [2].

Такі програми можуть, наприклад, керувати пропускною здатністю, динамічно призначати пріоритети різних видів трафіку, контролювати доступ до мережних ресурсів.

Архітектура мережі SDN представлена рис. 1.4.

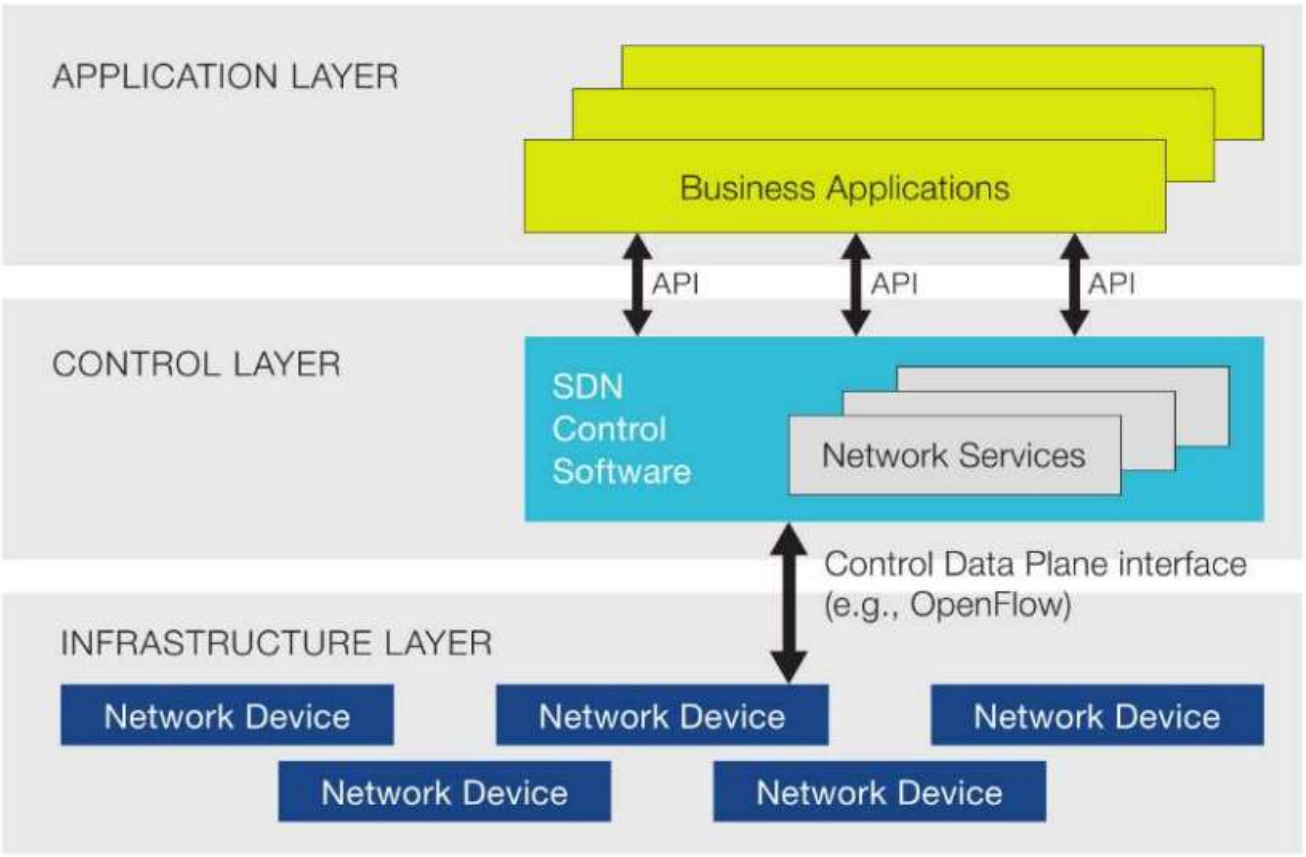


Рис.1.4. Архітектура SDN

До недоліків концепції належить висока вартість реалізації. Устаткування з підтримкою даної технології, що повністю керується з центрального контролера, може дозволити не кожна компанія. До того ж, зараз існує лише кілька виробників такого обладнання.

1.5. Автоматизовані методи керування якістю обслуговування

Автоматизовані системи управління якістю обслуговування – це програмно-апаратні комплекси, які централізовано розраховують топологію мережі, а потім автоматично підключаються до мережного обладнання та налаштовують його. Основними компонентами є:

–керуюча система - є ЕОМ зі спеціальним програмним забезпеченням, який автоматизовано налаштовує мережне обладнання;

–база даних, що зберігає інформацію про мережу (топологія, політика тощо);

–кероване мережеве обладнання, на якому необхідно налаштувати якість обслуговування.

Розглянемо автоматизовані системи управління якістю обслуговування з прикладу AutoQoS від компанії Cisco Systems.

AutoQoS дозволяє автоматизувати налаштування функції QoS, тобто. пріорітизації при доставці пакетів у мережі. AutoQoS ввело цю функціональність в операційні системи IOS та Catalyst OS та спростило впровадження IP QoS у локальні та глобальні мережі. Це дало можливість автоматизувати налаштування інфраструктури IP при впровадженні технології «голос поверх IP» (VoIP) по всій мережі — від розподільної шафи до корпоративної IP -магістралі, а також спростити налаштування послуг, які надає сервіс-провайдер [10].

Перевагами автоматизованих систем управління якістю обслуговування є:

– автоматизація та оптимізація процесу конфігурування обчислювальної інфраструктури підприємства;

– автоматизація процесів класифікації трафіку, створення політик обробки трафіку, налаштування конфігурації та інших;

– підвищення надійності мережі шляхом зменшення (у деяких випадках навіть виключення) операторських та конфігураційних помилок;

– можливість централізованого управління для моніторингу трафіку та отримання більш детальної інформації про стан мережевих операцій;

– можливість економічно управляти функціями QoS у великих IP -мережах,

завдяки поглибленому аналізу, інтелектуальному дизайну QoS та масштабованому впровадженню.

Недоліками автоматизованих систем управління якістю обслуговування є неможливість переналаштування QoS при зміні топології мережі та додаткові фінансові витрати для компанії при купівлі, впровадженні та експлуатації цих систем.

Висновки до розділу 1:

У рамках комп’ютерних мереж функціонують різні послуги, доступ до яких здійснюється за допомогою телекомунікаційної мережі. Дані внутрішньокорпоративні послуги являють собою різні служби, на яких ґрунтується робота організації, до них можна віднести:

– корпоративний портал для забезпечення взаємодії працівників організації;

– сервіси VoIP для забезпечення зв'язку між співробітниками та сторонніми організаціями;

– доменні служби;

– послуги бізнес-логіки;

– послуги , що надаються для юридичних осіб, наприклад, послуги фінансової звітності.

Забезпечення доступності даних сервісів і безперебійна їх робота є однією з пріоритетних задач розробки методів і засобів контролю і розподілу мережевого трафіку. За результатами порівняльного аналізу та огляду існуючих методів управління якістю обслуговування корпоративних мереж можна зробити такі висновки:

– ручне налаштування якості обслуговування - це трудомісткий і неефективний метод, пов'язаний з помилками через людський фактор, який може призвести до деградації сервісів та додаткових фінансових витрат;

– недоліком SDN є необхідність заміни існуючого парку мережного обладнання на нове з підтримкою SDN, що призводить до значних фінансових витрат підприємства. Також при виході з ладу контролера мережі перестає функціонувати вся мережа повністю, що змушує резервувати контролер, а це

призводить до додаткових фінансових витрат;

– недоліками автоматизованих систем управління якістю обслуговування є неможливість переналаштування QoS при зміні топології мережі та додаткові фінансові витрати для компанії при купівлі, впровадженні та експлуатації цих систем;

– більшість авторів що досліджують методи оптимізації проходження трафіку телекомунікаційних мереж, не враховують критерій доступності мережі і, як і раніше, оцінюють цей показник через коефіцієнт готовності мережі;

– обмеження, що перешкоджають досягненню високої доступності в програмно-керованої телекомунікаційної мережі, пов'язані з обмеженнями обчислювальної потужності контролерів і пристроїв, що організують рівень передачі, а також обєму пам'яті і буфера; затримкою між типами пристроїв контролер-контролер та контролер-комутатор; зростання трафіку в каналі зв'язку.

Для підвищення доступності мережі пропонується підхід підвищення показника доступності каналів зв'язку за рахунок застосування нового алгоритму керування потоком.

Тому в даній роботі буде необхідно:

–здійснити дослідження нового методу управління якістю обслуговування корпоративних мереж;

–розробити новий алгоритм управління якістю обслуговування;

–створити програмне забезпечення, яке цей підхід дозволить реалізувати та отримати результати практичних тестів.

РОЗДІЛ 2
АНАЛІЗ МЕТОДІВ І АЛГОРИТМІВ КЕРУВАННЯ ТРАФІКОМ В SDN МЕРЕЖАХ

У цьому розділі пропонується спосіб підвищення доступності каналу зв'язку програмно-керованих мереж. Описується розроблений алгоритм планування черг передачі даних на основі пріоритету, який дозволяє оптимізувати використання пропускної здатності та забезпечувати мінімально можливу затримку для пріоритетних класів.

2.1. Поняття доступності мережі та методи її забезпечення

Поняття "мережевої доступності" (network availability) у літературі виділяють в окремий клас. Незважаючи на це, більшість авторів [16] трактують мережеву доступність як властивість зберігати працездатний стан протягом деякого напрацювання, розраховують як власну доступність системи:

$$A = MTBF / (MTBF + MTTR),$$

де $MTBF$ (Mean time between failures) - середній час між відмовами, $MTTR$ (Mean Time to Restoration) - середній час, що необхідний для відновлення нормальної роботи після виникнення відмови [25].

У роботі Y. Zhou [20] розглядаються два основні фактори, що визначають доступність мережі:

– перший чинник – це доступність окремих мережевих елементів. Протягом терміну служби мережевого елемента бувають періоди, коли він не працює через несправність, технічне обслуговування або ремонт;

– другий чинник – топологія мережі. Очевидно, що більша надмірність у мережі (наприклад, більша кількість каналів, що з'єднують мережні комутатори)

призведе до більш високої доступності, а також до більш високих витрат на підтримку та керування.

Доступність каналу зв’язку. Припустимо, що у кожного користувача є рівно один канал зв’язку, яким він може обмінюватися даними з іншим користувачем. Якщо цей канал не працює, зв'язок припиняється. Оскільки канал складається з послідовного з'єднання вузлів та ліній зв'язку, його доступність просто обчислюється як добуток доступності вузлів та зв'язків, що становлять цей канал. Потім доступність мережі може бути визначена як доступність мінімального шляху по всіх парах вузлів. Оскільки всі шляхи відомі (або можуть бути обчислені за допомогою простого алгоритму найкоротших шляхів) доступність мережі легко обчислюється. Нехай p_{ij} означає можливість доступності каналу зв'язку (i , j), який з'єднує вузли i і j. Щоб отримати максимально можливу доступність мережі, мінімальна доступність шляху по всіх парах вузлів повинна бути максимальною. Якщо привласнити вагу $-log\ (p_{ij})$ кожного зв'язку $(i\ ,j) \in E$, то мету буде досягнуто за рахунок використання найкоротших шляхів (з урахуванням нових ваг) між парами вихідних пунктів призначення (sd). Найкоротший шлях із найбільшою вагою має найменшу доступність шляху та визначає оптимальну доступність мережі, яку можна отримати.

Доступність незавантаженої мережі. Інший крайній випадок – зв'язок може відбуватися по всіх можливих шляхах. Це дуже схоже на маршрутизацію в Інтернеті, де в разі збою протоколи маршрутизації автоматично перенастроюють таблиці маршрутизації для направлення трафіку альтернативними робочими шляхами до місця призначення. Недоступність мережі в цьому випадку визначається ймовірністю того, що між конкретною парою джерело-пункт призначення недоступний шлях (тобто мережу вимкнено). Максимальна ймовірність у всіх парах sd визначає недоступність мережі та, отже, доступність мережі.

Розрахунок доступності незавантаженої мережі насправді занадто оптимістичний на практиці, оскільки він передбачає, що, коли шляхи виходять з ладу, альтернативні шляхи мають достатньо ресурсів, доступних для обробки

трафіку каналу зв’язку, що відмовив, тому також необхідно формалізувати поняття доступності завантаженої мережі.

M. Durvy та ін у статті [5] визначають доступність мережі як відсоток від загальної пропускної здатності мережі, доступної для маршрутизації трафіку. Автори відзначають, що усереднена за часом доступність мережі ефективно відображає частоту і вплив збоїв у мережі. Також було введено нову метрику SLA (Service Level Agreement) під назвою «доступність послуги», яку визначили як час, протягом якого послуга доступна клієнту. У середовищі, схильному до збоїв, втрати і затримки пакетів можуть досягати рівня, при якому більшість програм не можуть працювати належним чином. У таких випадках автори вважають, що послуга недоступна для клієнта.

Незважаючи на зростаючу популярність SDN, аналіз робіт з предмета дослідження дозволив зробити висновок, що питаннями доступності мереж займаються вкрай мало. Є ряд досліджень щодо затримок при передачі пакетів [23], обчислення маршруту та балансування потоків [17,9] та передачі керуючих повідомлень від комутаторів до контролерів [5], у яких не торкається питання оцінки та оптимізації доступності мережі. Так, наприклад, зазначено, що збільшення затримок SDN може бути пов'язане з розташуванням контролера в мережі, неефективному програмному забезпеченні, організації правил переадресації і в контролі навантаження.

Область, пов'язана з доступністю SDN, досліджена недостатньо. Також варто відзначити, що багато досліджень проводилися на застарілих версіях протоколу OpenFlow, що не дозволяє повною мірою спиратися на отримані в ході даних робіт результати.

Проте, аналіз робіт [15; 9] дозволив виділити такі важливі для даної роботи фактори, що впливають на масштабованість і продуктивність, а, отже, і можливості підвищення доступності мереж, що використовують архітектурний принцип SDN:

– обчислювальна потужність контролерів та пристроїв, що організують рівень передачі;

– ємність пам'яті та буфера;

– розташування контролера в мережі, що в деяких дослідженнях не впливало на загальну продуктивність, а в інших спостерігалося зростання затримки;

– затримка між типами пристроїв контролер-контролер та контролер-комутатор;

– зростання трафіку у каналі зв'язку.

На рівні інфраструктури мережі масштабованість головним чином залежить від обладнання, недостатність обчислювальної потужності якого або нестача пам'яті впливає на рівень якості обслуговування всієї мережі. Відповідно, в таких умовах необхідно адаптувати механізми QoS для програмно-керованих мереж, які в загальному випадку реалізують такі функції:

-підтримка гарантованої смуги пропускання;

-зменшення втрат;

-управління навантаженнями;

-формування (шейпінг) трафіку;

-налаштування пріоритетів (класифікація) трафіку в мережі.

Виділимо основні поняття, які використовуються у службах QoS:

-черга (queue) – буфер пристрою, реалізований віртуально або фізично та використовуваний для зберігання сукупності пакетів, які колективно очікують передачі мережевим пристроєм на підставі планувальника пакетів;

-планувальник пакетів (scheduler) – це механізм, який керує чергою розподілу мережних пакетів. З його допомогою можна відкидати пакети, якщо буфер переповнився, а також змінювати порядок надсилання пакетів.

У різних операційних системах використовують різні планувальники пакетів. Планувальники зазвичай намагаються досягти балансу між ефективністю використання ресурсів та часом запуску програми.

Шейпінг пакетів (packet shaping) – це механізм керування трафіком, який затримує деякі або всі пакети, щоб привести їх у відповідність до бажаного профілю трафіку. Існує 2 типи алгоритмів формування трафіку: Leaky Bucket та Token Bucket.

Найбільшою популярністю користується ієрархічний алгоритм відра маркерів (Hierarchical Token Bucket, HTB), що здійснює поділ смуги пропускання для певних

типів потоку в окремі класи, кожен з яких має власну смугу пропускання. HTB вибудовує класи як дерева: вони можуть розділятися на дочірні класи, кожен із яких ділить між собою смугу батьківського класу.

Багато авторів пропонували власні реалізації алгоритмів управління трафіком, серед них: S. Keith [11], C. Bastian [1], L. Guo [7], K. Stanwood [17], однак, у всіх цих роботах кількість класів трафіку, як і раніше, оцінюється виходячи з з інтенсивності потоку (throughput), а величина затримки доставки пакета клієнту не враховується.

Ряд робіт, наприклад, [6, 10, 12, 14] присвячені забезпеченню QoS у програмно-керованих мережах. У цих роботах в основному розглядалися питання адаптації алгоритмів забезпечення якості обслуговування для роботи з протоколом OpenFlow, розробки фреймворків для роботи алгоритмів QoS в SDN та впливу маршрутизації та положення контролера на характеристики якості обслуговування, серед яких переважно розглядали надійність, масштабованість та балансування навантаження.

2.2. Алгоритм пріоритизації та управління потоком HTB для підвищення доступності мережі

Одним із ефективних способів підвищення доступності в програмно керованих мережах є впровадження технології QoS. Використання алгоритмів пріоритизації та контролю трафіку (ПКТ) дозволяє відокремлювати трафік функціонуючих сервісів із загального потоку та забезпечувати гарантовану смугу пропускання для них. Конфігурування даних алгоритмів здійснюється на підставі виділення частини КС для різних типів трафіку. Однак, такий розподіл не враховує особливості трафіку, що проходить, і не гарантує забезпечення доступності трафіку, чутливого до затримок каналу. Тим самим, доступність сервісу знижується, що веде до потенційної загрози безпеці системи. Цю проблему можна вирішити шляхом розробки алгоритмів пріоритизації та контролю трафіку, що дозволяють оптимально розподіляти трафік, що проходить через КЗ, з метою підвищення доступності певних типів трафіку.

Під підвищенням доступності каналів зв'язку ми розумітимемо гарантоване попадання в задані часові інтервали (директивний час), що забезпечується за допомогою зменшення часу відгуку сервісу шляхом мінімізації часу обробки пакетів. Часові інтервали, що задаються, необхідні для кожного типу трафіку (сервісу), вказуються в SLA, далі називатимемо ці інтервали максимальним директивним часом (МДЧ).

Виділимо сутності:

–множина значень про рівень обслуговування з елементами $\Delta_{\text{сер}}\,(i)$ – заданий (максимально допустимий) час відгуку для i -го сервісу;

–множина відгуків сервісів з елементами $t(i)$. $t(i)$ залежить від затримки маршрутизатора $\Delta_{\text{МЗ}}(\,i)$, затримки ОС $\Delta_{\text{ОС}}\;(i)$, затримки ПЗ сервісу $\Delta_{\text{ПЗ}}\;(i)$.

Отже,

$$t(i)\;=\;\Delta_{\text{МЗ}}(\,i)\;+\;\Delta_{\text{ОС}}\;(i)\;+\Delta_{\text{ПЗ}}\;(i).$$

Введемо обмеження $Y = \{Y_1, Y_2, Y_3\}$ на:

–розмір буферної пам'яті мережевого обладнання Y_1,

–швидкість передачі пакетів мережевим обладнанням Y_2,

–роботи кількох сервісів у рівних пріоритетах Y_3.

Тоді функція доступності i -го сервісу виглядає такою чином:

$$t(i,Y)\;\leq\;\Delta\,\text{сер}\,(i)\,.$$

Дослідження показують пряму залежність відгуку сервісу від використовуваного алгоритму планування черг. Найчастіше для класифікації трафіку використовується алгоритм планування Hierarchical Token Bucket. Розглянемо принцип роботи цього алгоритму [4, 3].

Класова дисципліна HTB призначена для поділу смуги пропускання між різними типами трафіку, кожен з яких може отримати частку гарантованої смуги пропускання. Алгоритм передбачає класифікацію трафіку за певними ознаками,

такими як: IP-адреса призначення чи джерела, порт призначення чи джерела, протокол передачі і т.д. Кожен клас відповідає певному типу трафіку та має свій пріоритет відповідно до SLA. Кожен клас має своє чергу накопичення пакетів, алгоритм HTB вибудовує класи як дерева.

Клас служби визначає параметри контролю, такі як максимальна пропускна здатність (max-limit або max-rate) або максимальний розмір пакета та використовує дисципліну черги для забезпечення дотримання цих правил. Планувальник та клас пов'язані один з одним, а правила, визначені класом, мають бути пов'язані з визначеною чергою. Найчастіше кожен клас має однієї дисципліною черги, але й можливо, що кілька класів разом застосовують одну й ту саму чергу. Найчастіше при постановці пакетів у чергу компоненти контролю певного класу відкидають пакети, перевищують певну вхідну інтенсивність.

HTB здійснює керування потоком мережевих пакетів шляхом виділення токенів на їх передачу відповідно до пріоритетів.

Алгоритм Token Bucket (рис.2.1.) базується на аналогії з відром, де маркери, представлені в байтах, додаються з певною швидкістю. Саме відро має заданий об’єм. Якщо відро заповнюється, нові токени викидаються.

Перш ніж дозволити будь-якому пакету пройти через чергу, відро черги перевіряється, щоб побачити, чи воно вже містить достатньо маркерів на цей момент. Якщо так, відповідна кількість токенів видаляється ("перераховується"), і пакету дозволяється пройти через чергу. Якщо ні, пакети залишаються на початку черги очікування пакетів, доки не буде доступна відповідна кількість токенів.

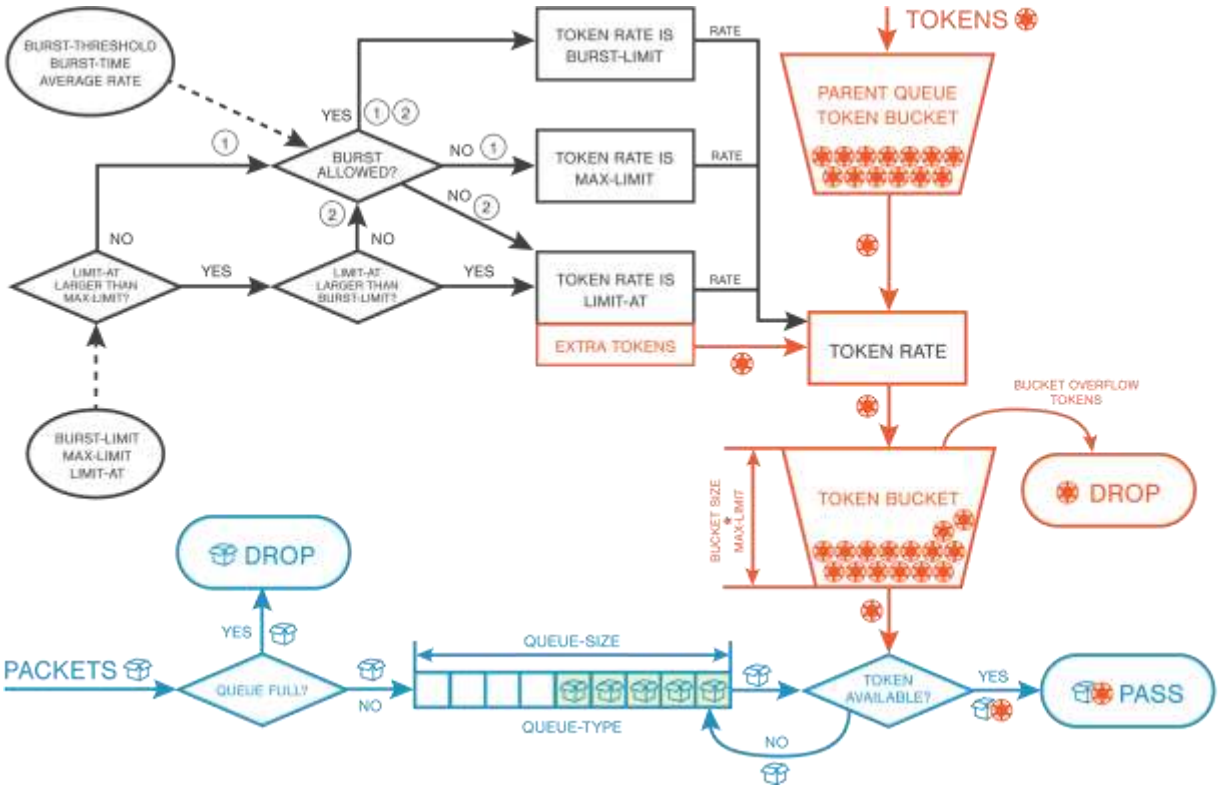


Рис. 2.1. Виділення токенів в HTB

Будь-який дочірній (листовий) клас (рис.2.2), який хоче запозичувати токен, буде забирати його у свого батьківського класу, який, у свою чергу, може запозичувати у свого батьківського класу, поки токен не буде знайдений або кореневий клас не буде досягнутий. Шейпінг відбувається лише у листових класах.

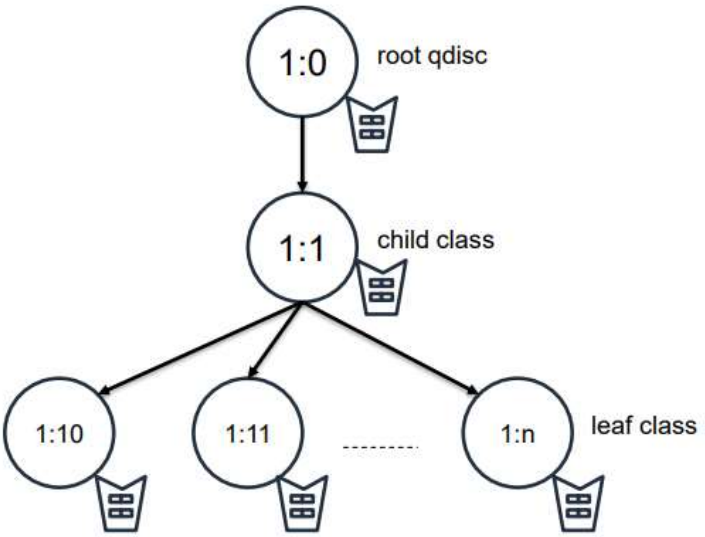


Рис. 2.2. Структура класів Hierarchy Token Bucket (HTB)

Кожна черга в HTB має два обмеження швидкості:

–CIR (Committed Information Rate) – (limit-at у RouterOS) найгірший сценарій, канал отримає таку кількість трафіку, незважаючи ні на що (припускаючи, що ми дійсно можемо надіслати стільки даних);

–MIR (максимальна швидкість інформації) – (max-limit у RouterOS) найкращий сценарій, швидкість, яку може досягти канал передачі, якщо батьківська частина черги має вільну пропускну здатність.

Іншими словами, спочатку limit-at (CIR) усіх черг буде задоволено, лише тоді дочірні черги намагатимуться запозичити необхідну швидкість передачі даних у батьківських, щоб досягти свого max-limit (MIR).

Ось чому, щоб забезпечити оптимальне (як розроблено) використання функції подвійного обмеження, ми пропонуємо дотримуватися цих правил:

1) Сума встановлених значень усіх дочірніх має бути меншою або дорівнювати кількості трафіку, доступного батьківському класу

$$CIR(parent) \geq CIR(child1) + \ldots + CIR(childN).$$

2) Максимальне значення для будь-якого дочірнього має бути меншим або дорівнювати максимальному значенню батьківського класу

$$MIR\ (parent) \geq MIR(child1)\ \&$$
$$MIR\ (parent) \geq MIR(child2)\ \& \ldots \&\ MIR\ (parent) \geq MIR(childN)\ .$$

Пріоритет відповідає за розподіл трафіку батьківських черг, що залишився, до дочірніх черг, щоб вони могли досягти max-limit.

Черга з вищим пріоритетом досягне свого max-limit раніше черги з нижчим пріоритетом. 8 – найнижчий пріоритет, 1 – найвищий. Пріоритет працює лише для leaf черги - пріоритет у внутрішній черзі немає значення якщо вказано max-limit (а не 0)

Розглянемо HTB в дії. Для цього ми візьмемо одну структуру HTB і спробуємо охопити деякі можливі ситуації та функції, змінивши обсяг вхідного трафіку, який HTB має передати змінивши деяких параметрів.

Структура HTB складатиметься з 5 черг:

–Queue01 внутрішня черга з двома дочірніми елементами - Queue02 та Черга03;

–Queue02 внутрішня черга з двома дочірніми елементами - Queue04 та Черга05;

–Queue03 кінцева черга;

–Queue04 кінцева черга;

–Queue05 кінцева черга.

Queue03, Queue04 та Queue05 — це клієнти, яким постійно потрібна швидкість 10 Мбіт/с. Вихідний інтерфейс здатний обробляти трафік 10 Мбіт/с.

При звичайному розподілі HTB було створено таким чином (рис. 2.2), що, задовольняючи всі обмеження, головна черга більше не мала вільної пропускної здатності для розподілу, значення max-limit=10 Мбіт/с для всіх черг:

–Черга 01 limit-at=0 Мбіт/с,;

–Черга 02 limit-at=4 Мбіт/с;

–Черга 03 limit-at=6 Мбіт/с, пріоритет =1;

–Черга 04 limit-at=2 Мбіт/с, пріоритет =3;

–Черга 05 limit-at=2 Мбіт/с, пріоритет =5.

Для такого прикладу отримаємо наступний вихідний розподіл трафіку:

–Черга 03 отримає 6 Мбіт/с;

–Черга 04 отримає 2 Мбіт/с;

–Черга 05 отримає 2 Мбіт/с.

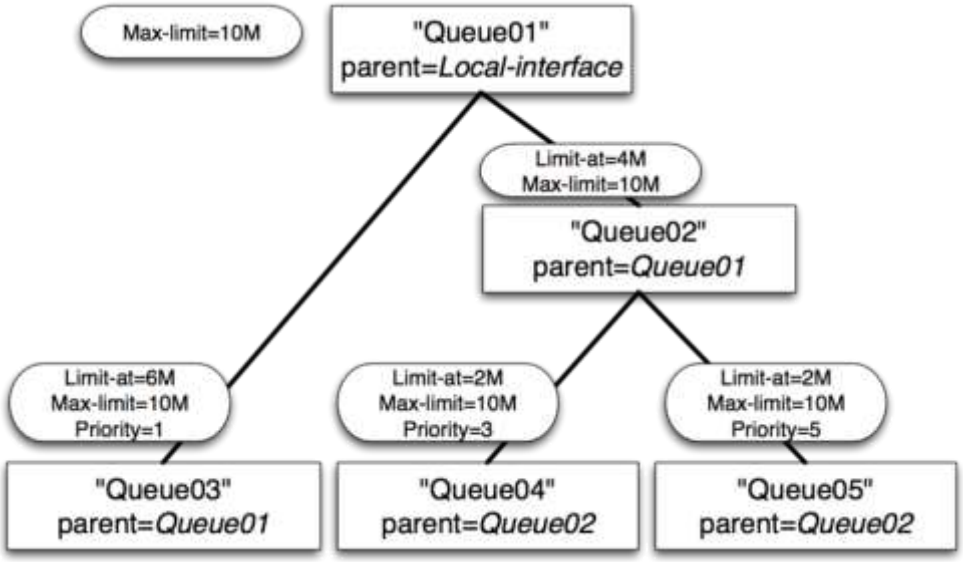


Рис.2.2. Звичайний випадок розподілу трафіку

Модифікуємо звичайний випадок із завданням максимальних лімітів (рис2.3.) значення max-limit=10 Мбіт/с для всіх черг:

–Черга 01 limit-at=0 Мбіт/с;

–Черга 02 limit-at=4 Мбіт/с;

–Черга 03 limit-at=2 Мбіт/с, пріоритет =3;

–Черга 04 limit-at=2 Мбіт/с, пріоритет =1;

–Черга 05 limit-at=2 Мбіт/с, пріоритет =5.

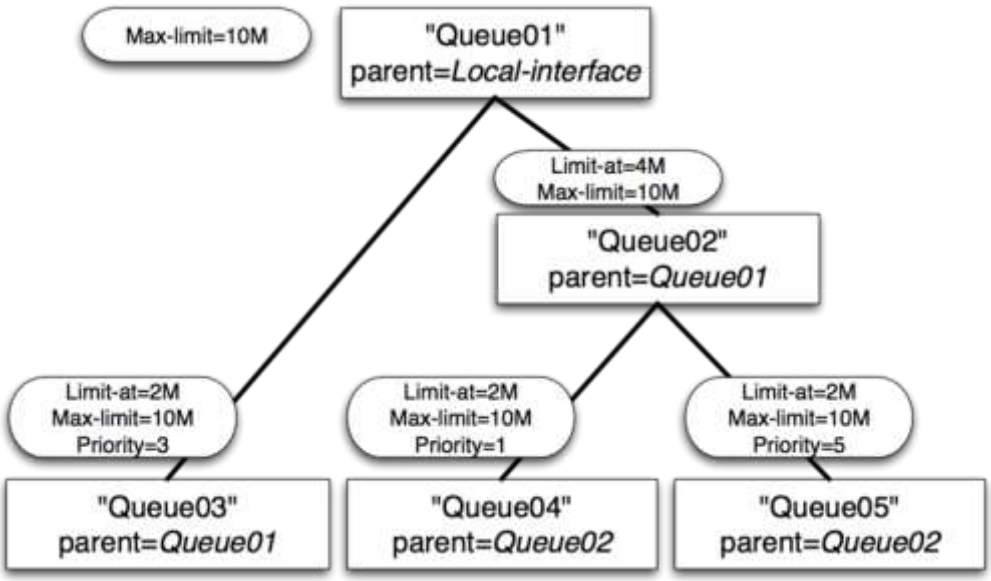


Рис.2.2. Звичайний випадок із визначеним max-limit

Для такого прикладу отримаємо наступний вихідний розподіл трафіку:

–Черга 03 отримає 2 Мбіт/с;

–Черга 04 отримає 6 Мбіт/с;

–Черга 05 отримає 2 Мбіт/с.

Тобто після задоволення всіх limit-at HTB надасть пропускну здатність черзі з найвищим пріоритетом (Queue04).

Розглянемо наступний випадок з конфігуруванням внутрішньої черги limit-at. значення max-limit=10 Мбіт/с для всіх черг:

–Черга 01 limit-at=0 Мбіт/с,;

–Черга 02 limit-at=8 Мбіт/с;

–Черга 03 limit-at=2 Мбіт/с, пріоритет =1;

–Черга 04 limit-at=2 Мбіт/с, пріоритет =3;

–Черга 05 limit-at=2 Мбіт/с, пріоритет =5.

Тут внутрішня черга Queue02 мала limit-at, щоб він зарезервував 8 Мбіт/с пропускної здатності для черг Queue04 та Queue05 та має найвищий пріоритет (рис. 2.3).

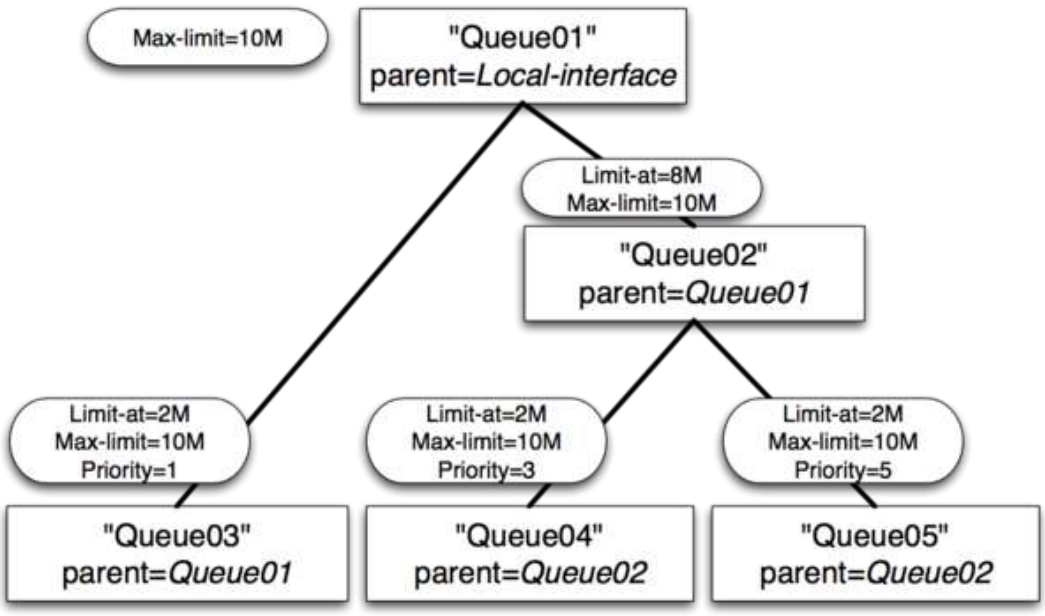


Рис.2.3. Приклад конфігурування limit-at на внутрішній черзі Queue02

В результаті отримаємо наступний розподіл:

–Черга 03 отримає 2 Мбіт/с;

–Черга 04 отримає 6 Мбіт/с;

–Черга 05 отримає 2 Мбіт/с.

Розглянемо наступний випадок з конфігуруванням кінцевої черги limit-at (рис. 2.4.):

–Черга 01 limit-at=0 Мбіт/с, max-limit=10 Мбіт/с;

–Черга 02 limit-at=4 Мбіт/с, max-limit=10 Мбіт/с;

–Черга 03 limit-at=6 Мбіт/с, max-limit=10 Мбіт/с, пріоритет =1;

–Черга 04 limit-at=2 Мбіт/с, max-limit=10 Мбіт/с, пріоритет =3;

–Черга 05 limit-at=12 Мбіт/с, max-limit=15 Мбіт/с, пріоритет =5.

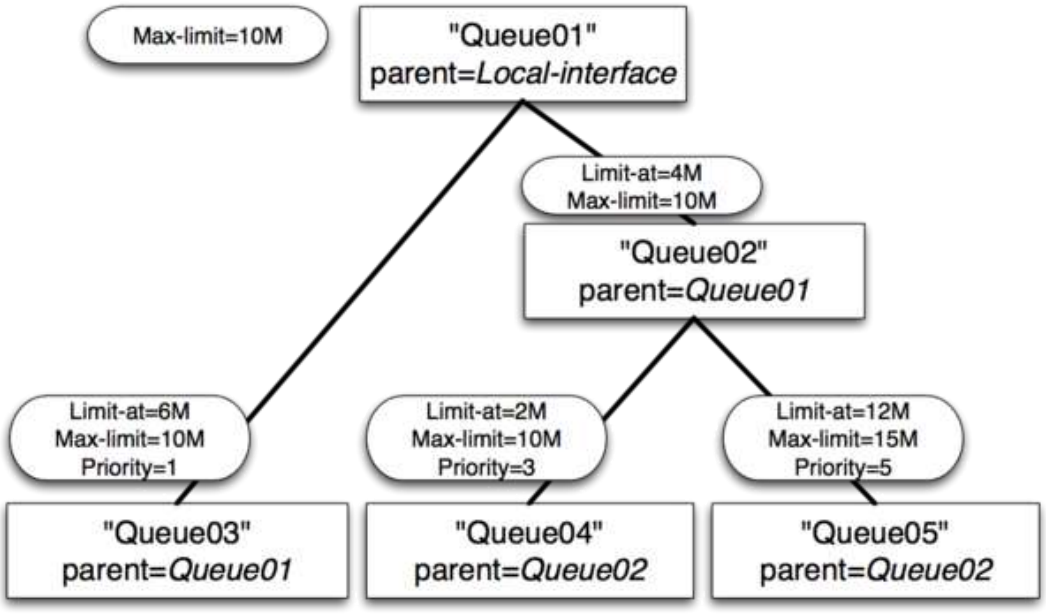


Рис.2.4. Приклад конфігурування limit-at на кінцевій черзі

В результаті отримаємо наступні значення:

–Черга 03 отримуватиме ~3 Мбіт/с;

–Черга 04 отримає ~1 Мбіт/с;

–Черга 05 отримає ~6 Мбіт/с.

В цьому випадку задовільнивши всі limit-at, HTB був змушений виділити 20 Мбіт/с – 6 Мбіт/с для черги 03, 2 Мбіт/с для черги 04 та 12 Мбіт/с до черги 05, але вихідний інтерфейс може обробляти лише 10 Мбіт/с. Оскільки черга вихідного

інтерфейсу зазвичай є FIFO, розподіл пропускної здатності зберігатиме співвідношення 6:2:12 або 3:1:6.

2.3. Алгоритм планування черг передачі даних

У цій роботі традиційний алгоритм HTB буде змінено наступним чином: при виділенні токенів на передачу трафіку алгоритм спиратиметься не на визначені йому параметри смуги пропускання, а на дотриманні входження затримки класу трафіку директивному інтервалу часу.

Відповідно для кожного класу трафіку визначається директивний час затримки пакета в черзі на очікування передачі мережевим обладнанням. Кожен клас має своє чергу накопичення пакетів. Випуск пакетів у передаючу чергу здійснюється зі швидкістю, розрахованою на підставі директивного часу та алгоритму планування використання пропускної здатності.

Підхід передбачає класифікацію мережевого трафіку за певними ознаками, такими як:

–IP-адреса вузла призначення, вузла джерела;

–порт вузла призначення, вузла джерела;

–реалізований протокол передачі даних.

Кожному класу трафіку визначається пріоритет відповідно до угоди про якість обслуговування (SLA). Для кожного класу трафіку визначається директивний час затримки пакета у черзі на очікування передачі мережевим обладнанням. Кожен клас має своє чергу накопичення пакетів. Випуск пакетів у вихідну чергу здійснюється зі швидкістю, розрахованою на підставі директивного часу та алгоритму планування використання пропускної здатності.

Математично це можна описати:

–множина класів трафіку:

$$Kl = \{Kl_1, \dots, Kl_b\},$$

де b – кількість класів. Також задається множина пріоритетів класу:

$$P_r = \{Pr_0, \dots, Pr_{b-1}\}.$$

Відповідно менше значення Pr_{i}, означає вищий пріоритет класу;

–множина швидкостей потоку пакетів для кожного класу:

$$Sp = \{Sp_1, \dots, Sp_b\}, \sum_{i=1}^{b} Sp_i \leq max_{rate},$$

де max_{rate} - максимальна пропускна здатність інтерфейсу;

– множина директивного часу обробки пакетів для кожного класу:

$$Tобр = \{Tобр_1, \dots, Tобр_b\},$$

відповідно інтенсивності обслуговування пакетів визначаються $\mu_i = 1/Tобр_i$;

–множина допустимих затримок у чергах для класів

$$Del = \{Del_1, \dots, Del_b\};$$

–множина значень інтенсивності пакетів вхідного трафіку:

$$\lambda = \{\lambda_1, \dots, \lambda_b\}.$$

Алгоритм визначає одну чергу передачі пакетів різних класів. З черг класів пакети виходять з інтенсивністю меншою або рівною μ_i. Пакети потрапляють у чергу послідовно відповідно до пріоритетів Pr_{i}.

Формалізуємо основні кроки алгоритму планування черг передачі даних:

Крок 1. Ввід вхідних даних:

$$Kl, P_r, T\text{обр}, max_{rate}, min_{delay},\ MTU, Sp\ , \lambda.$$

Крок 2. Поставити у відповідність класам їх пріоритет:

$$Kl_0 = Pr_0, \dots , Kl_{b-1} = Pr_{b-1}\ .$$

Крок 3. Якщо умова $\sum_{i=1}^{b-1} Sp_i \leq max_{rate}$ виконується, то перехід до кроку 4, інакше встановлюємо поточне значення часу обробки пакета для кожного класу мінімально допустимим:

$$T_i^* = min_{delay}, i = 0, \dots , (b - 1).$$

Крок 4. Визначаємо сумарну кількість токенів, яку надалі розподілятимемо за класами

$$Tok = max_{rate}/(\ MTU \times 8)\ .$$

Далі проводиться розподіл токенів за класами (смуги пропускання вихідної черги) таким чином, що більш пріоритетний клас забирає 2/3 доступних токенів (смуги пропускання), кожен наступний за пріоритетом клас займає 2/3 смуги, що залишилася після попереднього за пріоритетом класу, і так далі до останнього класу, який займає смугу, що залишилася $i = 0$.

Крок 5. Кількість токенів для i - го класу, $Ent(.)$ – ціла частина.

$$Tok_i = \begin{Bmatrix} Ent\left(\frac{2}{3}Tok\right), \text{якщо } \left(\frac{2}{3}Tok\right) > Ent\left(\frac{2}{3}Tok\right) \\ \frac{2}{3}Tok, \text{якщо } Ent\left(\frac{2}{3}Tok\right) = \frac{2}{3}Tok \end{Bmatrix}.$$

Крок 6. Час обробки пакетів i -го класу

$$T_i^* = (8 \times MTU) / (max_{rate} \times Tok_i).$$

Крок 7. Якщо $T_i^* \leq T\text{обр}_i$ (менше директивного), то $\mu_i^* = 1/T_i^*$; $i = i + 1$.

Якщо $i < b$ (залишилися класи), то перехід до кроку 5.

Інакше якщо $Tok = 0$ (не залишилися токени), то кінець алгоритму. Інакше кінець алгоритму.

Інакше якщо $T_i^* \geq$ min_delay, то $Tok_i = Tok_i + 1$, $Tok = Tok - 1$, якщо $Tok = 0$ (не залишилися токени), то кінець алгоритму, Інакше перехід до кроку 6.

Інакше токенів для i -го класу занадто багато $Tok_i = Tok_i - 1$, $Tok = Tok + 1$, перехід до кроку 6;

Якщо інтенсивність вхідного потоку в більш пріоритетному класі нижче вихідної граничної інтенсивності, то наступний за пріоритетом клас може зробити підвищення граничної вихідної інтенсивності (при збереженні $\sum_{i=1}^{b} Sp_i \leq max_{rate}$) . Право підвищення граничної вихідної інтенсивності переходить низькопріоритетним класам.

Якщо інтенсивність потоку пакетів більше граничної інтенсивності потоку, то пакети починають відкидатися для вирівнювання пікової інтенсивності потоку до граничної інтенсивності потоку. Підвищення чи зниження інтенсивності пакетів визначається результаті роботи алгоритму планування використання пропускної здатності у вигляді перерозподілу токенів НТВ на передачу.

В додатку Б представлено блок-схему розробленого алгоритму.

2.4. Алгоритм підтримки низькопріоритетних сервісів

Тестування розробленого алгоритму на устаткуванні у вигляді програмної реалізації та за допомогою імітаційної моделі виявило проблему нерівномірності втрат пакетів у чергах.

Для вирішення цієї проблеми було запропоновано наступне рішення: спробуємо знизити втрати пакетів у низькопріоритетних чергах за рахунок запозичення невеликої частини смуги пропускання у високопріоритетних черг, що

допоможе підвищити продуктивність мережі в цілому. Для цього розподіляти між дочірніми класами (чергами) надлишки токенів батьківських класів.

У розробленій моделі, як і в оригінальному НТВ, надлишок токенів віддавався дочірньому класу з найвищим пріоритетом у разі конкуренції за токени. Під надлишком токенів розуміється різниця між гарантованими токенами батьківського класу та суми гарантованих токенів його дочірніх класів.

У новій модифікації моделі надлишок токенів батьківського класу ділитимемо між його дочірніми класами.

Формалізуємо основні кроки алгоритму підтримки низькопріоритетних сервісів.

Крок 1. Ввід : $Kl, P_r, T\text{обр}, max_{rate}, min_{delay}, \ MTU, Sp\ , \lambda$.

Крок 2. $Kl_0 = \ Pr_0, \dots\ , Kl_{b-1} = \ Pr_{b-1}$.

Крок 3. Якщо $\sum_{i=1}^{b-1} Sp_i \leq max_{rate}$, то перехід до кроку 4, інакше $T_i^* = min_{delay}, i \ = \ 0, \dots\ , (b \ - \ 1)$.

Крок 4 $Tok \ = \ max_{rate}/(\ MTU \ \times \ 8)$.

Крок 5. $Tok_i \ = \ Tok \ \times \ 2/3$.

Крок 6. $T_i^* = \ (8 \times MTU)\ /(\ max_{rate} \times \ Tok_i)$.

Крок 7. Якщо $T_i^* \leq T\text{обр}_i$, то $\mu_i^* = 1/T_i^*$; $i \ = \ i \ + \ 1$; якщо $i < b$, то до кроку 5, інакше якщо $Tok = 0$, то кінець алгоритму; інакше i = 0 , перехід до кроку 8; інакше якщо $T_{i^*} \geq min\_delay$, то $Tok_i = Tok_i + 1$, $Tok = Tok - 1$, якщо Tok = 0, то кінець алгоритму; інакше перехід до кроку 6; інакше $Tok_i = Tok_i - 1 \ \ , Tok = Tok + 1$, перехід до кроку 6;

Крок 8. Якщо $\lambda_i > \mu_i^*$ (Перевірка відповідності інтенсивності обслуговування інтенсивності вхідного потоку пакетів, тобто можливості збільшити смугу передачі для i -го класу), то $Tok_i = Tok_i + 1 \ , Tok = Tok - 1$, перехід до кроку 6; інакше i = $i \ + \ 1$; якщо $i < b$ (залишилися класи), то перехід до кроку 8, інакше кінець алгоритму.

Висновки до другого розділу:

Розроблено алгоритм планування черг передачі на основі модифікації відомого підходу «ієрархічне відро маркерів» (HTB). Алгоритм дозволяє забезпечувати мінімально можливу затримку для пріоритетних класів сервісів, що підтримуються, оптимізуючи використання пропускної здатності.

Розроблено алгоритм підтримки низькопріоритетних сервісів в умовах сильного домінування високопріоритетних сервісів, що ґрунтується на перерозподілі токенів управління потоком, що дозволяє забезпечити принцип справедливості щодо всіх сервісів, що працюють у КПТС

РОЗДІЛ 3
АПРОБАЦІЯ ЗАПРОПОНОВАНИХ АЛГОРИТМІВ РОЗПОДІЛУ ТРАФІКУ В SDN МЕРЕЖАХ

3.1. Дослідження роботи алгоритму пріоритизації HTB

Для виявлення вузьких місць у функціонуванні алгоритму HTB в умовах порушення мережевої доступності на експериментальній установці (рис. 3.1) було проведено дослідження його типової конфігурації у різних режимах навантаження. Метою експерименту було визначення ключових аспектів, що впливають на забезпечення низької затримки передачі пакетів при різній інтенсивності вхідного потоку пакетів. Елементи експериментальної установки представлені у таблиці 3.1. Установку було розгорнуто на віртуальній машині VMWare.

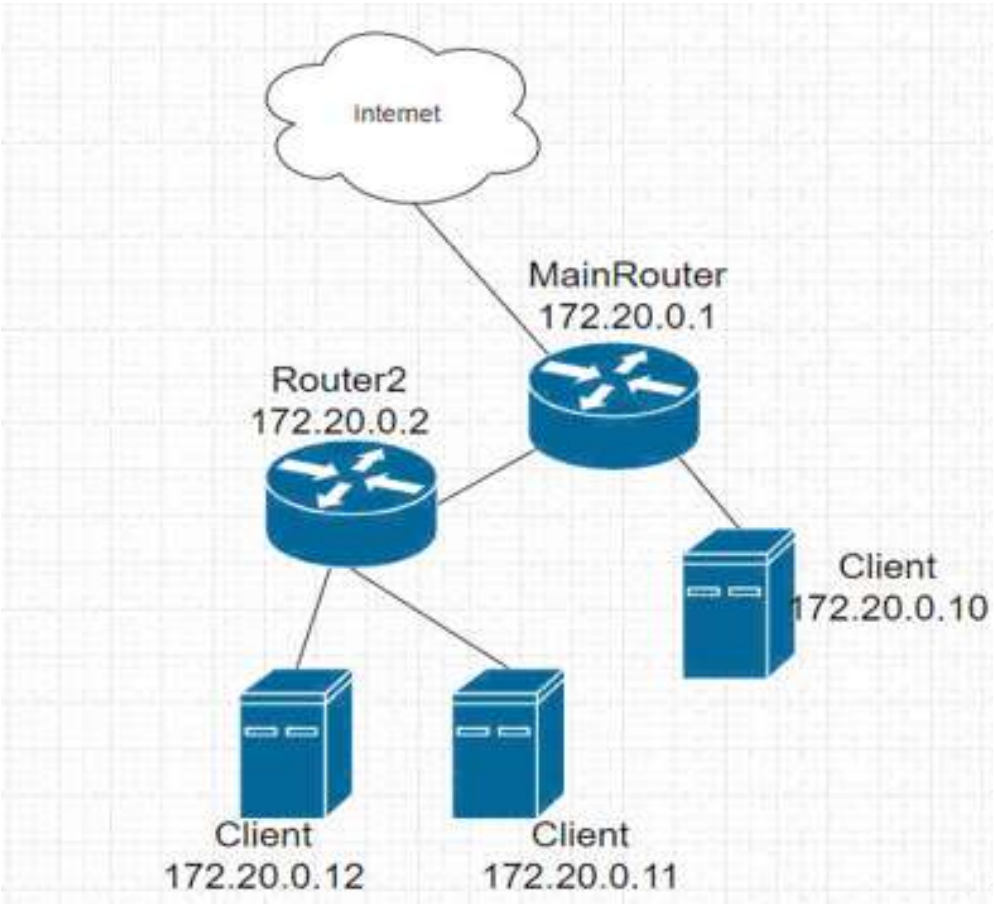


Рис. 3.1. Схема тестованої програмно-керованої мережі

Таблиця 3.1

Склад експериментальної установки

Найменування елемента	Опис елемента
MainRouter	Програмний маршрутизатор інтернет-провайдера на ОС xUbuntu, що забезпечує надання послуги з надання доступу до мережевих сервісів
Client-1, Client-2, Client-3	Робочі станції клієнтів на ОС xUbuntu
Router-2	Програмний маршрутизатор призначений для розподілу передачі пакетів вузлів мережі (Client-2, Client-3)

Конфігурація алгоритму планування черг HTB представлено в таблиці 3.2.

Таблиця 3.2

Конфігурація HTB

Пристрій	Батьківський клас (пристрій)	Пріоритет	Швидкість передачі класу (Mbit/s)
MainRouter	-	0	10
Router-2	MainRouter	1	5
Client-1	MainRouter	0	5
Client-2	Router-2	0	3
Client-3	Router-2	1	2

У межах проведення експерименту було визначено 5 режимів тестування алгоритму, подані у таблиці 3.3.

Таблиця 3.3

Показник значень інтенсивності в заданий інтервал часу

Режим тестування	Значення інтенсивності першої черги (Пріоритетна)	Значення інтенсивності другої черги	Значення інтенсивності третьої черги
1	Початкове	Підвищується	Підвищується
2	Початкове	Початкове	Початкове
3	Початкове	Підвищується	Початкове
4	Початкове	Початкове	Початкове
5	Підвищується	Підвищується	Підвищується

Значення інтенсивності вхідного потоку пакетів, визначені для кожного навантаженого класу, представлені в таблиці 3.4. Зміна інтенсивності здійснювалась скриптом за допомогою утиліти hping3. Розмір пакета складає 10000 байт.

Таблиця 3.4

Інтенсивності надходження пакетів вхідного потоку

Час, с	Значення інтенсивності (інтервал у секундах між відправкою пакетів)
1-15	0,5 (початкова)
15-55	0,1
55-75	0,075
75-100	0,05

На даний момент існує три методи, що дозволяють викликати модуль керування трафіком HTB в операційній системі:

–традиційний командний рядок UNIX, що використовує утиліту traffic control із програмного пакету iproute2;

–HTB-інструменти, запропоновані Spirlea, Subredu та Stanimir [8] для спрощення процесу розподілу пропускної здатності;

–набір інтерфейсів WEB-інструментів: WebHTB та T-HTB, що використовуються для формування пакетів.

У цій роботі було обрано перший варіант, тому трафік контролюватимемо через утиліту tc з програмного модуля iproute. Розглянемо роботу алгоритму HTB в Linux, використовуючи утиліту hping3 для визначення затримок у класах пріоритетів. Результати порівняння представлені нижче.

Під час тестування першого режиму (рис. 3.2) ми бачимо, що при підвищенні інтенсивності другої та третьої черги підвищується затримка у всіх чергах. Перша, найбільш пріоритетна черга має найнижчу затримку з усіх, оскільки має найбільшу пропускну здатність. Так як пріоритет другої черги вищий, ніж у третьої, пропускна здатність, що залишилася, ділиться між ними на користь другої. При підвищенні значення інтенсивності втричі бачимо, що виникає переповнення буфера і алгоритм скидає передачу пакетів.

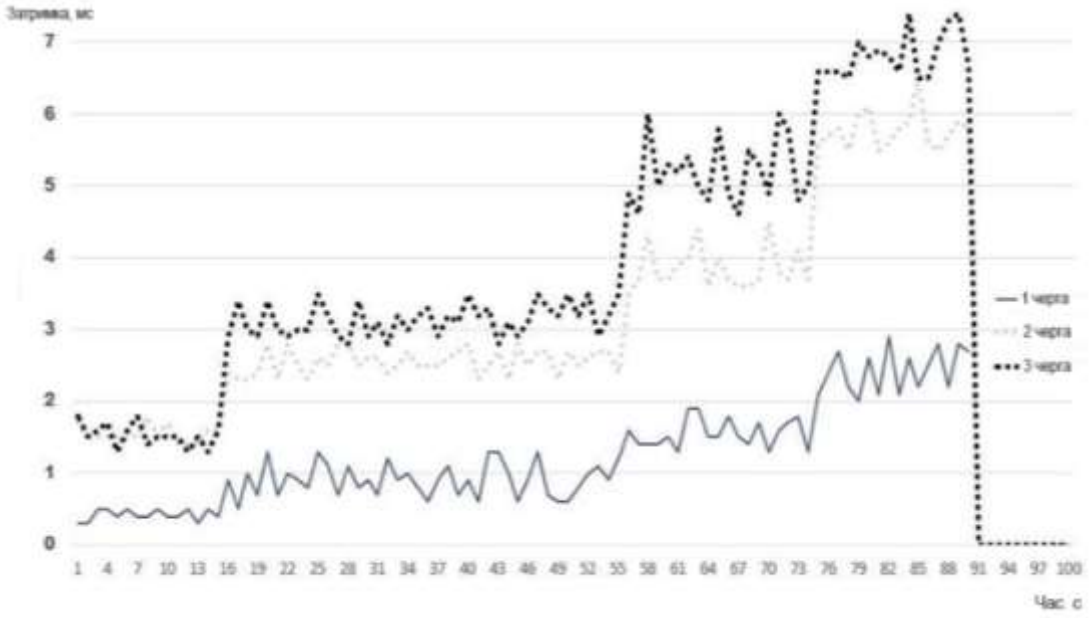


Рис. 3.2. Тестування НТВ у першому режимі

При тестуванні другого режиму (рис. 3.3) з підвищенням інтенсивності потоку у першій черзі спостерігається зростання затримки у всіх класах внаслідок підвищення інтенсивності потоку у пріоритетному класі. Розподіл пропускної здатності сприяє тому, що затримка найменш пріоритетної черги є найвищою, оскільки жертвує своєю пропускною здатністю на користь пріоритетної.

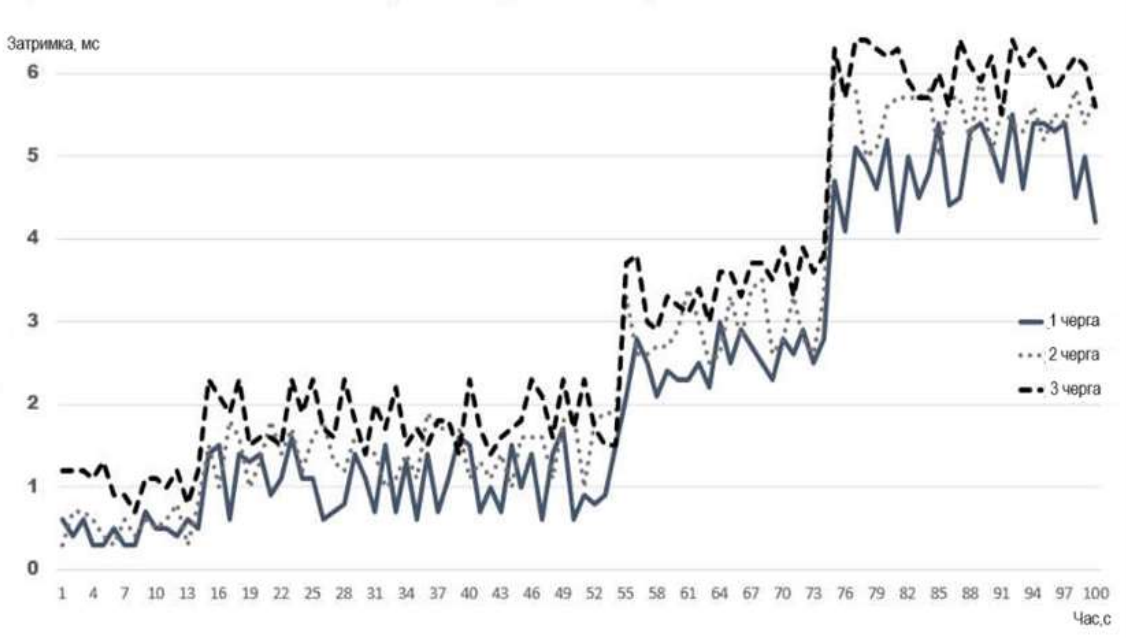


Рис. 3.3. Тестування НТВ у другому режимі

Третій режим (рис. 3.4) з підвищенням інтенсивності другого класу та фіксуванні значення інтенсивності інших черг показує, що затримка черги другого класу формується рахунок третьої черги. Перша черга має невелику затримку, оскільки має найвищий пріоритет, але при третьому підвищенні інтенсивності ми бачимо підвищення затримки за рахунок обмеження загальної пропускної здатності.

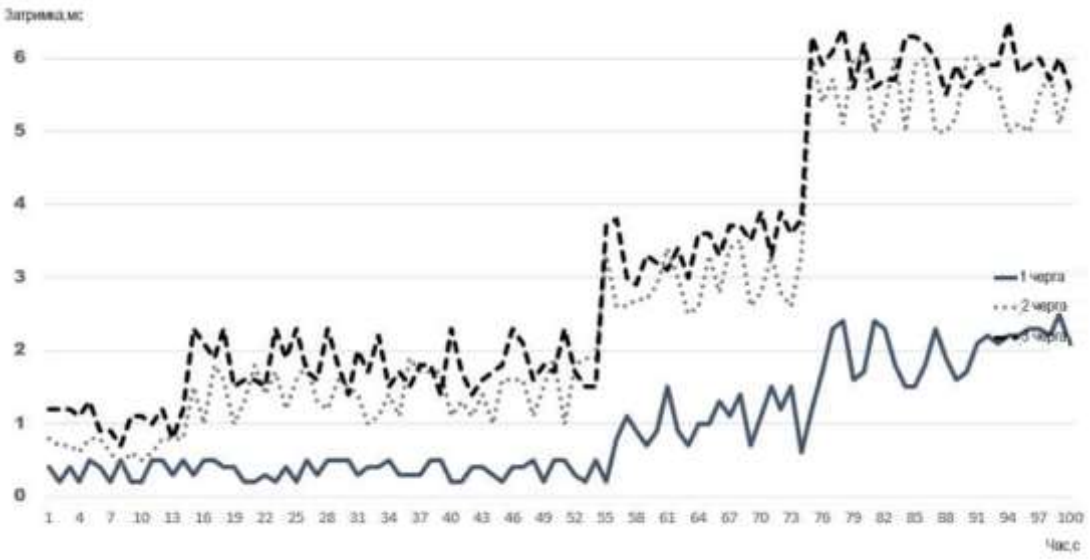


Рис. 3.4. Тестування НТВ у третьому режимі

При низькій інтенсивності вхідного потоку пакетів у режимі 4 (рис. 3.5) забезпечується низька затримка на обробку. Однак для низькопріоритетної черги затримка вища, ніж у пріоритетного класу, що обумовлено послідовністю права передачі пакетів на підставі пріоритету.

При підвищенні показників інтенсивності потоку всіх черг (5 режим, Рис. 3.6), спостерігається висока затримка передачі пакетів. На графіку бачимо розподіл затримок кожної черги залежно від їх пріоритету. При останньому підвищенні інтенсивності спостерігаються втрати пакетів.

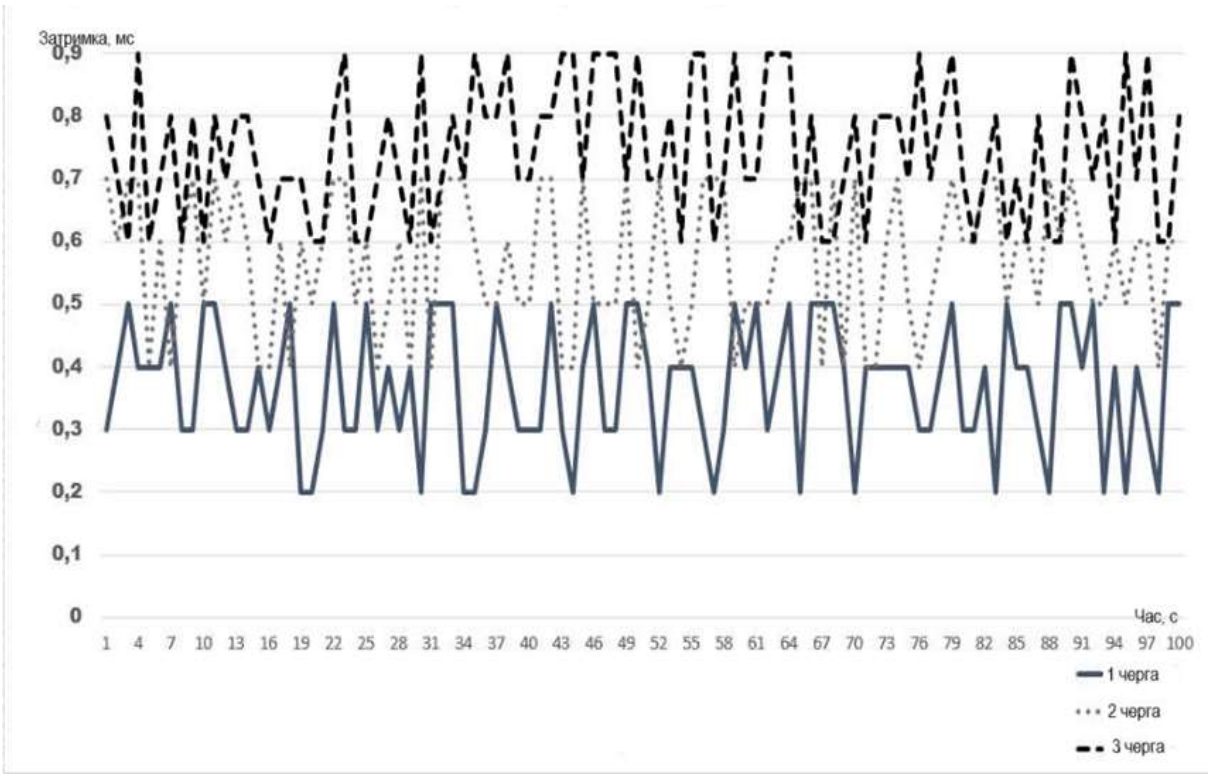


Рис. 3.5. Тестування НТВ у четвертому режимі

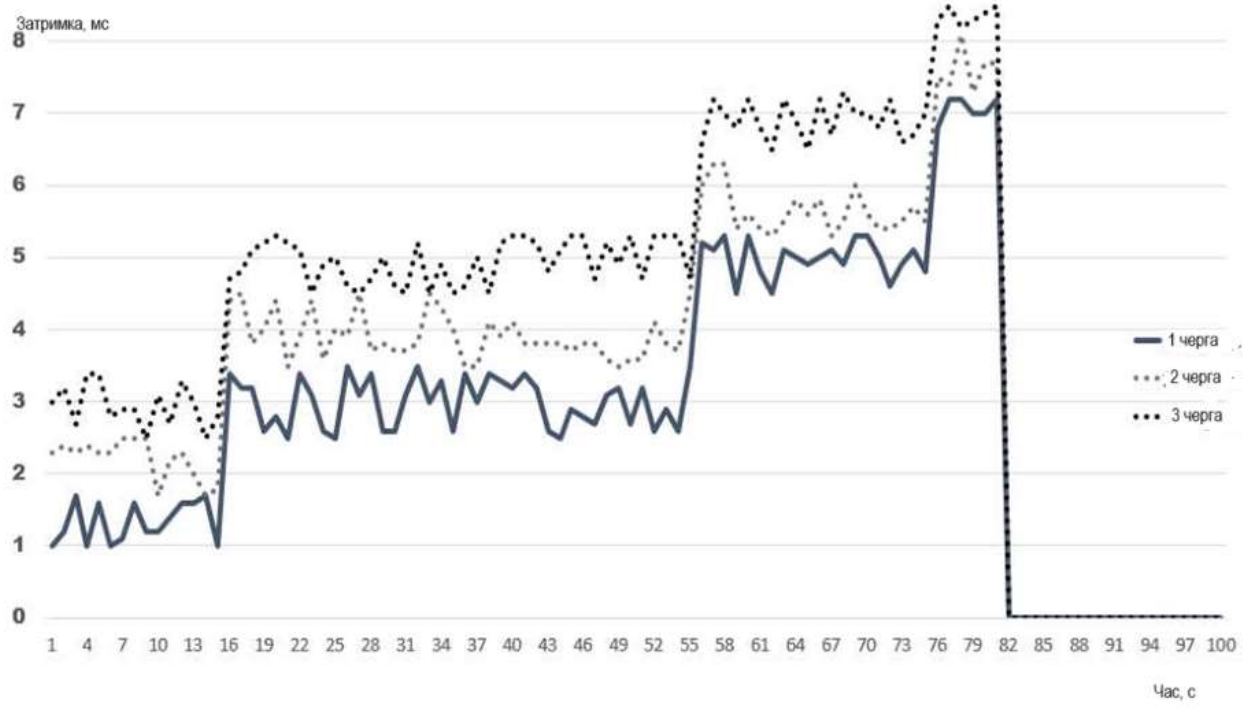


Рис. 3.6. Тестування НТВ 5 режим

Аналіз отриманих в результаті експерименту даних показав наступні можливі напрями оптимізації параметрів, що впливають на затримку передачі пакетів:

контроль інтенсивності вхідного потоку пакетів, динамічна зміна пропускної здатності каналу відносно вхідної інтенсивності, оптимізація надання смуги класів трафіку відносно вхідної інтенсивності. Виявлено необхідність контролю розміру буфера і захисту його від переповнення, також необхідно, щоб пріоритетні черги безпосередньо не залежали від заданого їм параметра пропускної здатності, а мали гарантовану затримку.

3.2.Перевірка ефективності модифікованого алгоритму HTB

Для перевірки ефективності розробленого алгоритму було проведено імітаційне моделювання його роботи в середовищі AnyLogic у порівнянні з роботою алгоритму HTB типової конфігурації.

Система, що моделюється, складалася з наступних елементів:

− черги передачі пакетів із пріоритетом 0;

− черги передачі пакетів із пріоритетом 1;

− черги передачі пакетів у середу передачі сигналу;

− планувальника розподілу використання черги передачі пакетів у середу для черг із пріоритетами.

Кожна модельована черга мала наступні параметри:

− інтенсивність потоку пакетів λ_i ;

− час обробки пакетів у черзі χ_i;

− гарантована затримка передачі пакетів черги, d_i;

Для черг задавалися такі обмеження:

− інтенсивність пакетів у черзі передачі пакетів в середовищі передачі не може перевищувати значення max_rate ;

− час обробки пакетів у черзі передачі пакетів у середовищі передачі завжди становить min_delay ;

− буфер черг із пріоритетами дорівнює 100 пакетів.

На рис. 3.7 представлена схема імітаційної моделі AnyLogic.

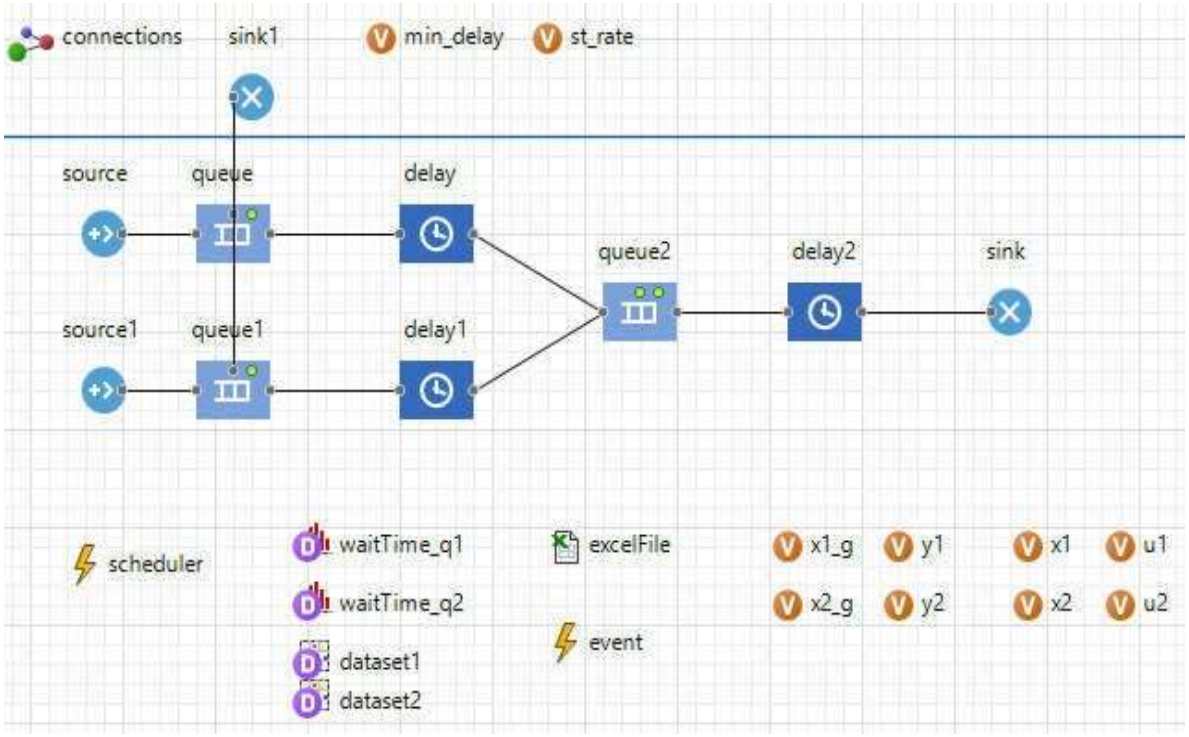


Рис. 3.7. Схема імітаційної моделі

У рамках тестування розробленого алгоритму було зроблено порівняння затримок передачі пакетів кожної черги за однакових навантажень на канал черги класів. Дослідження передбачало проведення 5-ти тестових вимірювань затримки передачі пакетів. Режими тестування представлені у таблиці 3.5. Результати проведеного експерименту подано у вигляді графіків (рис. 3.8–3.12).

Таблиця 3.5

Режими тестування моделей алгоритмів

Режим тестування	Інтенсивність черги 1 (пакет/мс)	Інтенсивність черги 2 (пакет/мс)	Кількість вимірювань
Режим 1	0,48	0,857	180
Режим 2	100	0,857	180
Режим 3	0,48	100	180
Режим 4	20,825	62,4	180
Режим 5	100	100	180

На рис. 3.8 зображено графік роботи алгоритмів у режимі 1. З графіка видно, що при низькій інтенсивності вхідного потоку обидва алгоритми забезпечують

низьку затримку обробки пакетів. Однак для низькопріоритетної черги НТВ затримка вище за пріоритетний клас, це обумовлено послідовністю права передачі пакетів на підставі пріоритету.

На рис. 3.9 представлений графік роботи алгоритмів у режимі 2. Графік показує, що НТВ зробив виділення смуги для пріоритетного класу і цим зайняв більшу частину ресурсів мережі, тому затримка у другому класі почала зростати. Розроблений алгоритм визначив, що інтенсивність потоку низькопріоритетного класу низька, тим самим можна забезпечувати низьку затримку для обох класів, залишаючись у директивному часі.

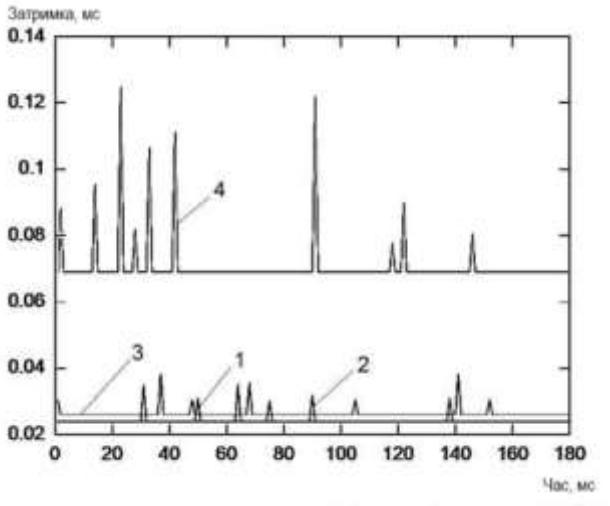


Рис. 3.8. Результат тестування моделей у режимі 1
1 - черга з пріоритетом 0 (модифікація НТВ), 2- черга з пріоритетом 1 (модифікація НТВ), 3- черга з пріоритетом 0 (НТВ), 4- черга з пріоритетом 1 (НТВ)

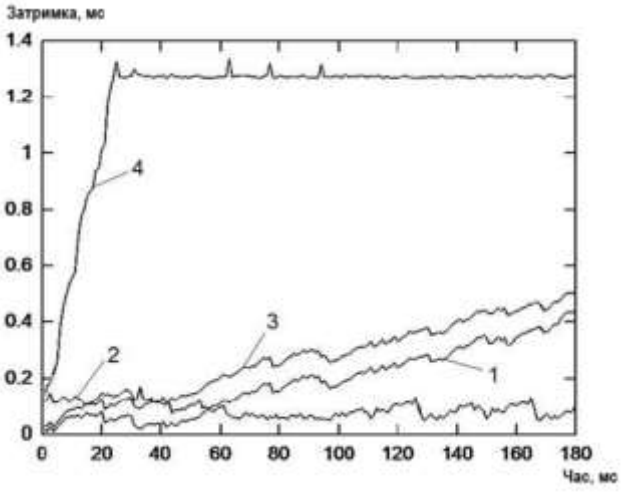


Рис. 3.9. Результат тестування моделей у режимі 2
1 - черга з пріоритетом 0 (модифікація НТВ), 2- черга з пріоритетом 1 (модифікація НТВ), 3- черга з пріоритетом 0 (НТВ), 4- черга з пріоритетом 1 (НТВ)

На рис. 3.10 представлений графік роботи алгоритмів у режимі 3. На графіку видно, що НТВ визначив низьку активність високопріоритетного класу і дозволяє низькопріоритетному класу використовувати більше смуги. Розроблений алгоритм, у свою чергу, визначив необхідну кількість смуги для високопріоритетного класу,

щоб видавати пакети з мінімальною затримкою, а частину, що залишилася, визначив для низькопріоритетного.

На рис. 3.11 наведено графік роботи алгоритмів у режимі 4. Навантаження на систему дорівнює 1, тобто межа обробки пакетів без втрат. З графіка слід, що НТВ зберігає задану смугу пропускання для високопріоритетного класу, не знижуючи його затримки, причому затримка низькопріоритетного класу сильно зросла.

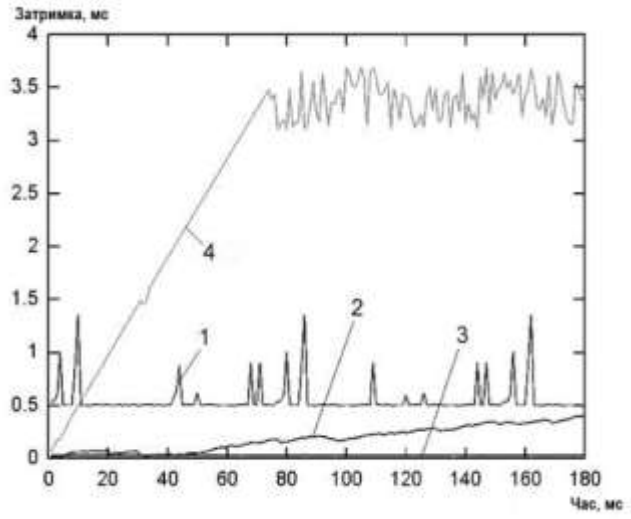


Рис. 3.10. Результати тестування моделей у режимі 3
1 - черга з пріоритетом 0 (модифікація НТВ), 2- черга з пріоритетом 1 (модифікація НТВ), 3- черга з пріоритетом 0 (НТВ), 4- черга з пріоритетом 1 (НТВ)

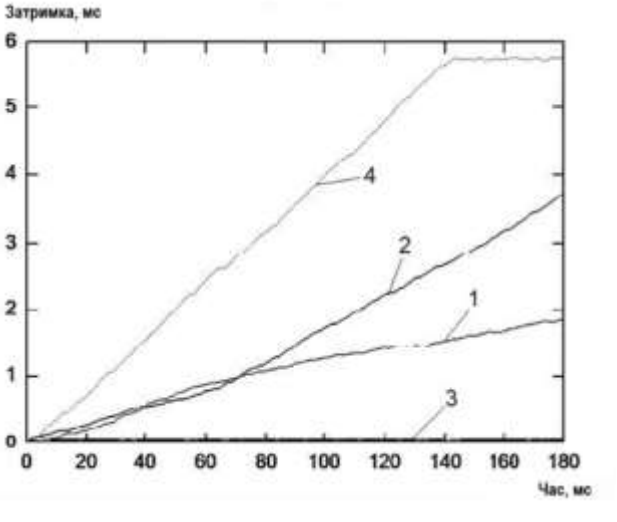


Рис. 3.11. Результати тестування моделей у режимі 4
1 - черга з пріоритетом 0 (модифікація НТВ), 2- черга з пріоритетом 1 (модифікація НТВ), 3- черга з пріоритетом 0 (НТВ), 4- черга з пріоритетом 1 (НТВ)

Розроблений алгоритм здійснює розрахунок оптимального розподілу смуги в межах навантаження на систему таким чином, щоб забезпечити середню мінімальну затримку для обох класів. Але оскільки класи мають різні пріоритети, затримка для високопріоритетного класу визначається нижче. До того ж, інтенсивність потоку заявок високопріоритетного класу нижча, ніж у низькопріоритетного, і відповідно до роботи алгоритму, пріоритетний клас отримує 100% необхідної смуги, а

низькопріоритетний лише ту частину, що залишилася. Тому ми спостерігаємо зростання затримки для низькопріоритетного класу.

На рис. 3.12 продемонстровано графік роботи алгоритмів у режимі 5. Інтенсивність вхідного потоку пакетів обох класів трафіку значно перевищує значення, можливе для обробки. Отже, відбувається накопичення пакетів у буфері черги, всі пакети, що виходять за межі буфера, починають відкидатися.

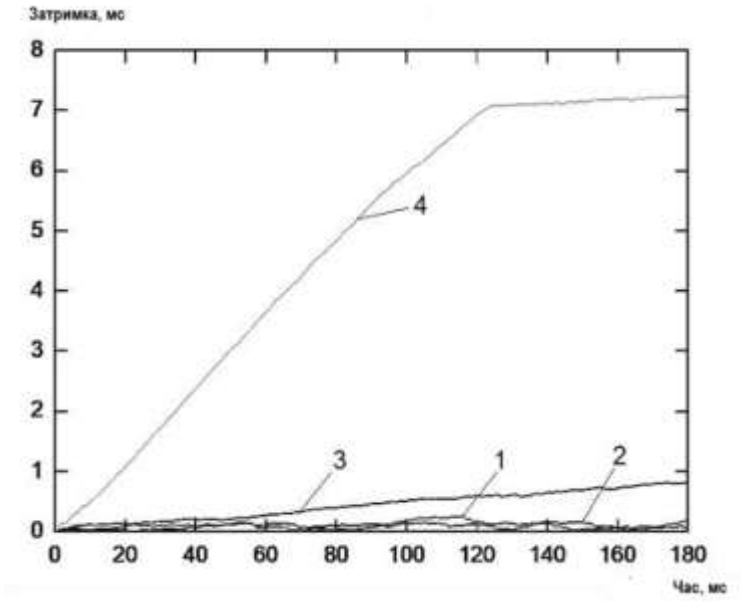


Рис. 3.12. Результати тестування моделей у режимі 5

1 - черга з пріоритетом 0 (модифікація НТВ), 2- черга з пріоритетом 1 (модифікація НТВ), 3- черга з пріоритетом 0 (НТВ), 4- черга з пріоритетом 1 (НТВ)

На графіку ми отримуємо значення затримки пакетів, передача яких була здійснена. І ця ситуація аналогічна режиму 4, коли система здатна обробити вхідний потік заявок, але так як інтенсивність обох класів однакова, то для низькопріоритетного класу залишається менше допустимої смуги і затримка низькопріоритетного класу починає зростати. НТВ робить обробку аналогічним чином, і розподіл смуги пропускання відбувається відповідно до визначеного конфігурацією значення без можливості розширення смуги для різних класів.

Результат порівняльного тестування алгоритмів дозволяє зробити висновок, що розроблений алгоритм показує нижчі сумарні значення затримки для різних

класів трафіку, забезпечуючи тим самим доступність сервісів в обох класах. Відповідно, запропонований алгоритм планування черг передачі даних дозволяє оптимізувати використання пропускної здатності та забезпечувати мінімально можливу затримку для пріоритетних класів. Однак у разі перевантаження в мережі можуть виникнути втрати пакетів у класах низької пріоритетності.

3.3. Імітаційне дослідження алгоритму підтримки низькопріоритетних сервісів

У тестуванні використано дві моделі відповідно для алгоритму планування черг передачі даних та алгоритму підтримки низькопріоритетних сервісів. Тестування проводилося у середовищі AnyLogic на основі моделі, описаній у розділі 3.2 та скоригованої щодо нових припущень. Схема уточненої імітаційної моделі представлена рис. 3.13. Кожен тест проводився протягом 2 хвилин і визначалася затримка кожного пакета, опис режимів тестування наведено у таблиці 3.6.

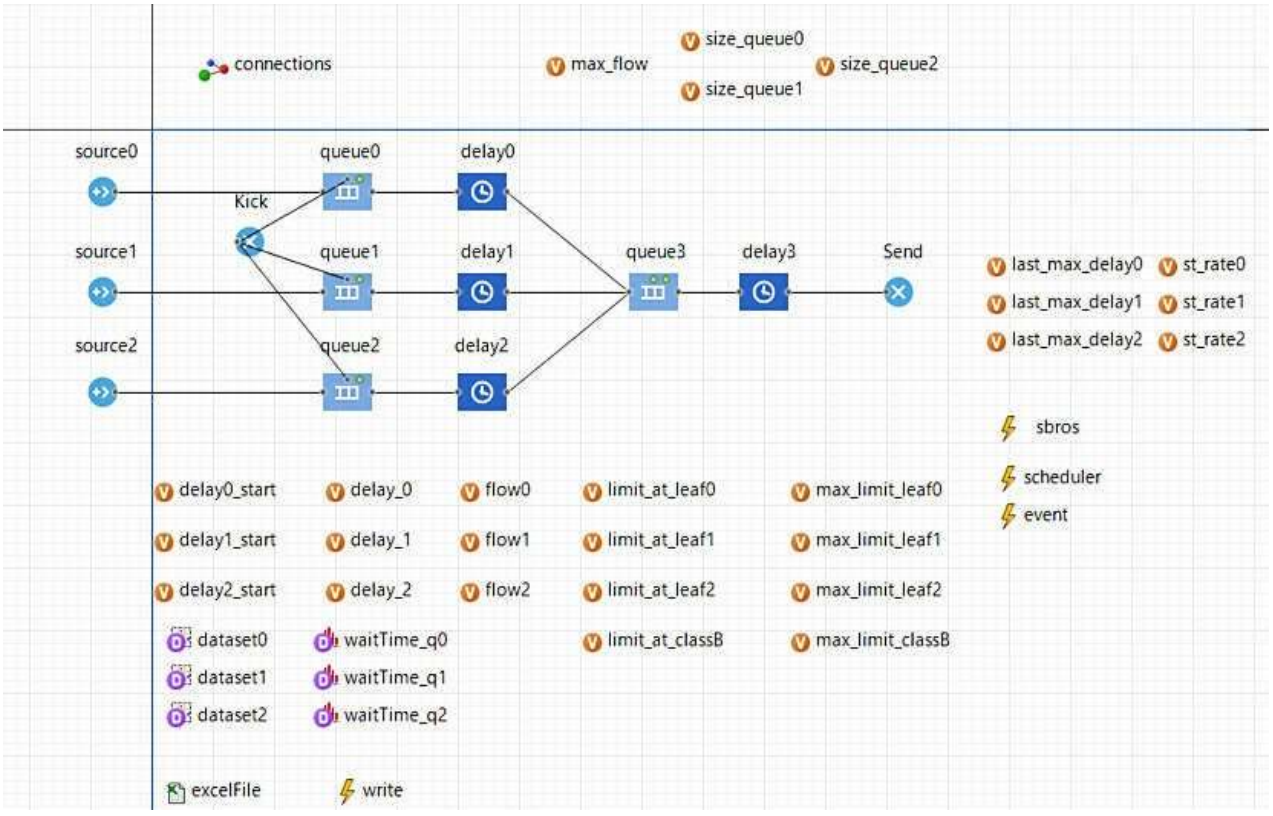


Рис. 3.13. Схема оновленої імітаційної моделі

Таблиця 3.6

Режими тестування

Режим тестування	Інтенсивність черги 1 (пакет/мс)	Інтенсивність черги 2 (пакет/мс)	Інтенсивність черги 3 (пакет/мс)
Режим 1	5	5	50
Режим 2	50	50	5
Режим 3	50	5	50
Режим 4	5	50	50
Режим 5	50	50	50
Режим 6	40	40	40

Встановимо для першої черги максимальний директивний час 2 мс, другої – 5 мс, для третьої - 10 мс (рис. 3.14).

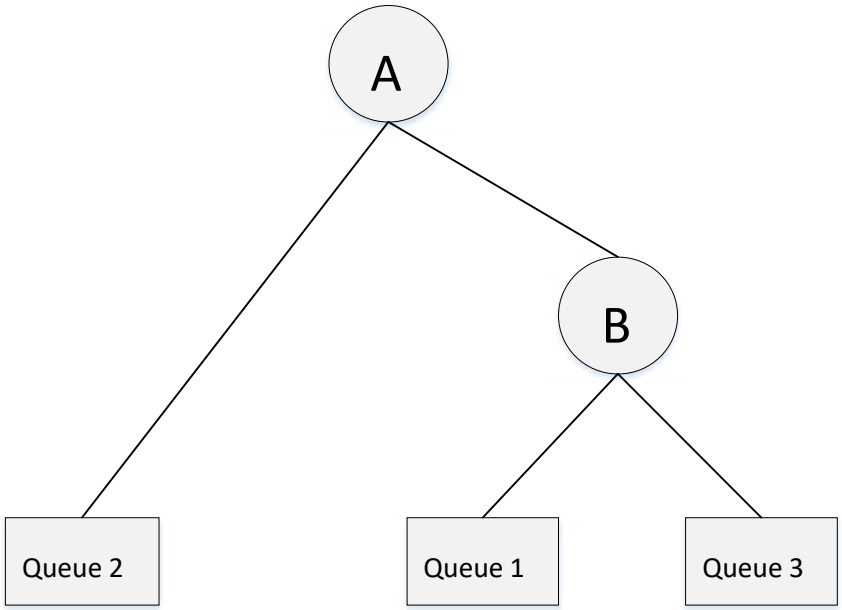


Рис. 3.14. Схематичне представлення моделей, що тестуються

Режим 1. Результати роботи (рис. 3.15 та 3.16) моделі показали, що в тому випадку, якщо навантажується лише один клас, різниця у роботі відсутня.

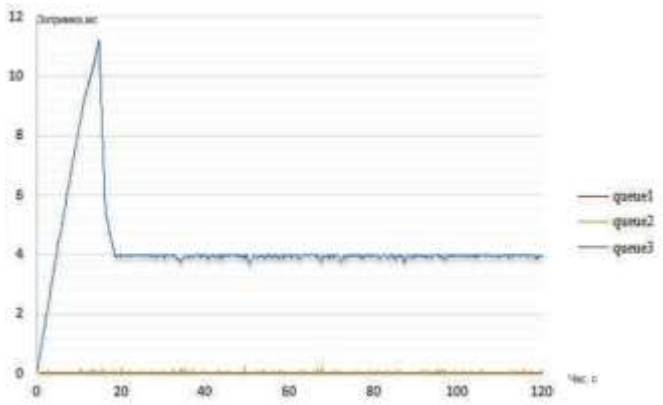


Рис. 3.15. Модель 1 Режим 1

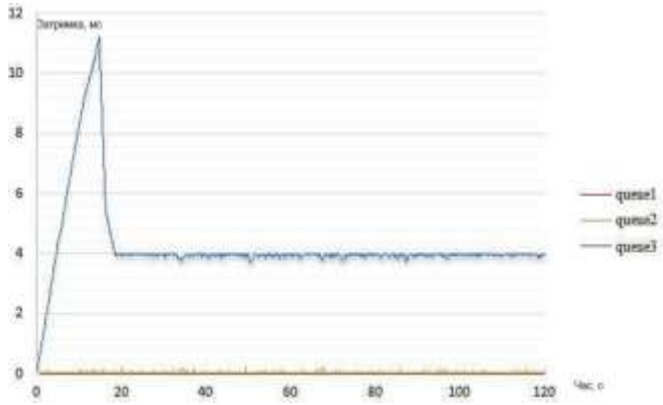


Рис. 3.16. Модель 2 Режим 1

Режим 2. У даному режимі тестування помітно (рис. 3.17 і 3.18), що друга модель поводиться гірше, частка пакетів, що прийшли за директивний час, у другій черзі зменшився вдвічі.

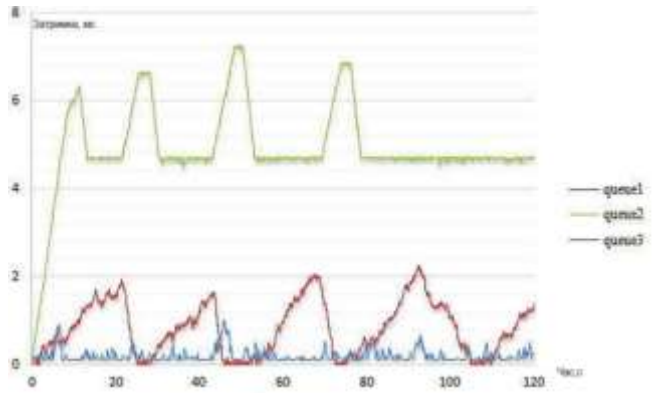


Рис. 3.17. Модель 1 Режим 2

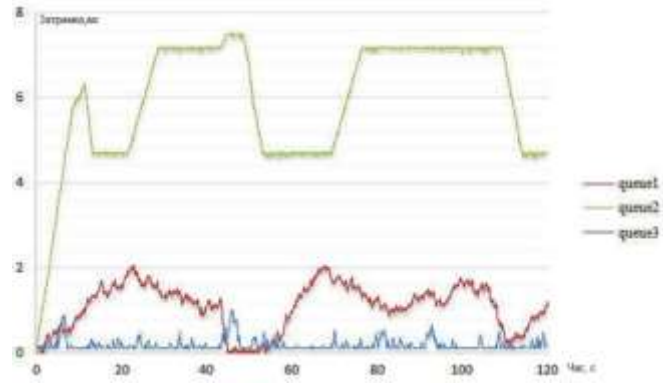


Рис. 3.18. Модель 2 Режим 2

Це пов’язано з тим, що для забезпечення затримки другої черги необхідний весь надлишок класу А, але оскільки перша черга майже завжди використовує свою частину надлишку А, то в другому класі спостерігаються зайві втрати, які відсутні у першої моделі (таблиця 3.7).

Режим 3. У цьому режимі тестування друга модель поводиться краще, оскільки для третьої черги досить зарезервованої нею частини надлишку токенів батьківських класів (таблиця 3.8).

Таблиця 3.7

Режим 2 . Частка пакетів, що прийшли за МДЧ.

	Модель 1	Модель 2
Queue1	97,6%	99,1%
Queue2	77,5%	39,8%
Queue3	100%	100%

Таблиця 3.8

Режим 3 . Частка пакетів, що прийшли за МДЧ.

	Модель 1	Модель 2
Queue1	94,1%	97,5%
Queue2	100%	100%
Queue3	72,2%	95,1%

Режим 4. У цьому режимі тестування найкращі результати показує перша модель. Таблиця 3.9 демонструє, що друга модель має такі самі проблеми, як і в другому режимі тестування: другій черзі недостатньо зарезервованих токенів.

Таблиця 3.9

Режим 4 . Частка пакетів, що прийшли за МДЧ.

	Модель 1	Модель 2
Queue1	100%	100%
Queue2	90,4%	41,2%
Queue3	90,5%	94,5%

Режим 5. У п'ятому режимі тестування моделі показали різні результати. Видно, що у другій моделі скоротилися втрати для третьої черги порівняно з першою, однак було втрачено майже всі пакети другої черги (табл. 3.10).

Таблиця 3.10

Режим 5 . Частка пакетів, що прийшли за МДЧ

	Модель 1	Модель 2
Queue1	100%	98,2%
Queue2	55,2%	4,8%
Queue3	65,6%	79,2%

Режим 6. При невеликому зниженні інтенсивності у кожній черзі все одно спостерігаються втрати у другій моделі у другому класі (таблиця 3.11).

Таблиця 3.11

Режим 6. Частка пакетів, що прийшли за МДЧ

	Модель 1	Модель 2
Queue1	100%	100%
Queue2	90,5%	56,8%
Queue3	89,6%	96%

На основі отриманих результатів можна зробити висновок, що розподіл надлишку токенів батьківського класу між дочірніми класами є менш ефективним з точки зору продуктивності мережі, ніж виділення надлишку токенів високопріоритетному класу в конкурентній боротьбі за токени між класами.

Алгоритм підтримки низькопріоритетних класів погано виявляє себе, коли класу необхідна кількість токенів, які набагато перевищують кількість гарантованих токенів для класу. Однак у тому випадку, коли кількість необхідних токенів класу трохи перевищує кількість гарантованих токенів або клас має багато батьків, у яких зарезервовані для нього токени, даний алгоритм показує стабільнішу затримку.

3.4. Експериментальне дослідження

Впровадження та тестування розробленого алгоритму в реальних умовах вимагало переконфігурації операційної системи на рівні ядра. У коді утиліти алгоритму HTB в операційній системі Linux, за допомогою команди insmod, був встановлений модуль, відповідальний за ефективний розподіл затримок між класами в залежності від їхнього пріоритету (Додаток В).

Код модуля ядра був написаний мовою C і скомпільований компілятором gcc+. Класифікація та маркування досліджуваних областей мережі проводилася за допомогою утиліти tc .

Експериментальне дослідження ефективності розробленого алгоритму ПКТ у порівнянні з типовим HTB складалося з двох експериментів:

Експеримент 1 . Тестування алгоритму HTB у різних режимах навантаження.

Експеримент 2 . Тестування модифікованого алгоритму.

Дослідження проводилося в рівних умовах послідовно через відсутність можливості запускати в операційній системі обидві версії алгоритму управління потоком одночасно. Програма експерименту та склад експериментальної установки описані в розділі 3.1, додатково на клієнтських пристроях контролювалася затримка:

–Client-1 з IP-адресою 172.20.0.10 має найвищий пріоритет і забезпечується гарантованою затримкою в 0,2 мс;

–Client-2 наступний рівень пріоритету, задана затримка 0,5 мс, але тільки в тих випадках, коли дотримуються умови, задана Client-1;

–Client-3 є пристроєм, що має найнижчий пріоритет і отримує затримку, виходячи з параметрів пропускної здатності мережі та завантаженості інших клієнтів.

У середовищі, що тестується, була встановлена максимальна пропускна здатність 10 Мбіт/с.

Для запуску модифікованого алгоритму необхідно викликати змінений модуль із ядра. Скрипт маршрутизатора RouterMain для виклику модуля sch_htb представлений на рис. 3.19. Цей скрипт використовує утиліти, що дають можливість класифікувати пріоритетні вузли, що входять до контрольованої мережі.

```
#!/bin/bash make
insmod./sch_htb.ko
tc qdisc add dev ens38 root handle 1: htb
tc class add dev ens38 parent 1: classid 1:10 htb rate 10Mbit
tc filter add dev ens38 parent 1: protocol ip prio 1 u32 match ip
dst 172.20.0.2/24 classid 1:10
tc filter add dev ens38 parent 1: protocol ip prio 2 u32 match ip
dst 172.20.0.10/24 classid 1:10
```

Рис. 3.19. Лістинг виклику модуля sch_htb

RouterMain являє собою програмний маршрутизатор, в який додається алгоритм пріоритизації, аналогічно налаштований і другий маршрутизатор, що забезпечує функціонування окремих сегментів мережі.

Порівняємо результати роботи розробленого модуля ядра зі стандартним алгоритмом HTB, встановленим у Linux.

З модифікованим HTB затримка пакетів першої черги є гарантованою, затримки другої та третьої черги зростають, оскільки одночасно підвищується інтенсивність потоку. На відміну від алгоритму HTB у найбільш пріоритетної черги спостерігається гарантована затримка, однак підвищується значення затримок інших черг. При останньому підвищенні інтенсивності потоку в обох випадках починаються втрати пакетів, однак модифікований алгоритм вирішує цю проблему шляхом розвантаження найменш пріоритетних черг (рис. 3.20).

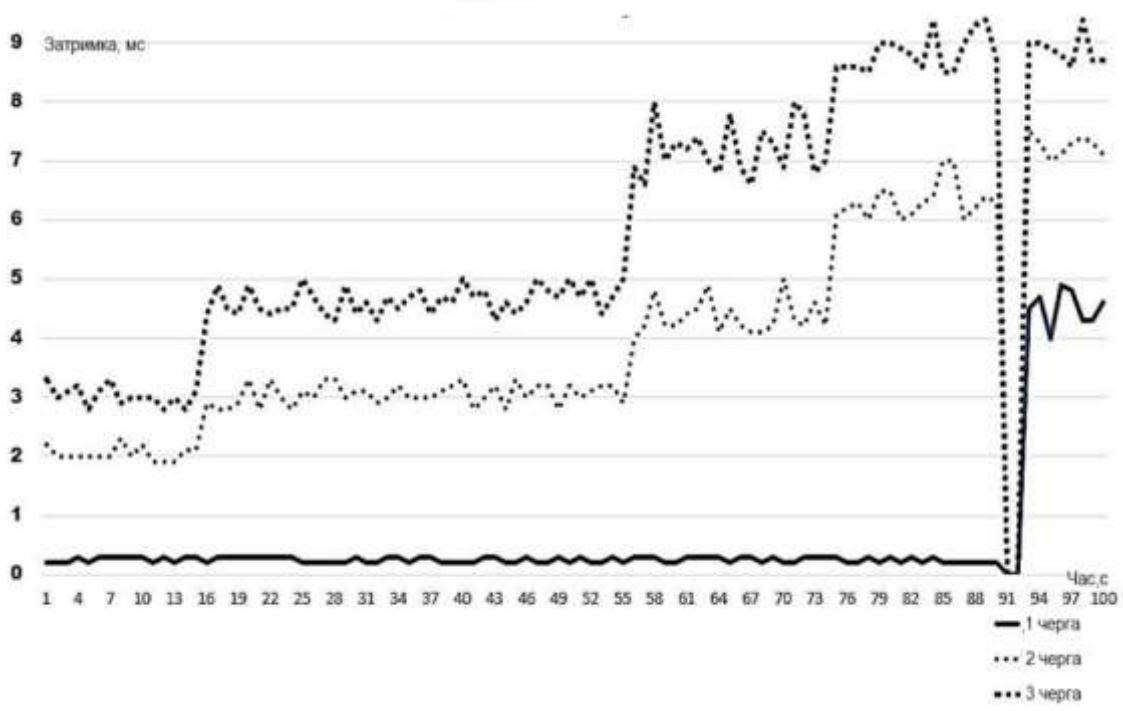


Рис. 3.20. Тестування 1 режиму модифікованого HTB

Цей спосіб дозволяє зберегти доступність мережі навіть за умов високої інтенсивності передачі пакетів.

У другому режимі при стандартному НТВ значення затримки рівномірно розподіляється між другою та третьою чергою на будь-якому кроці підвищення інтенсивності, а затримка першої черги зростає з підвищенням інтенсивності потоку. Завдяки модифікованому алгоритму НТВ затримка пакетів першого класу не відхиляється від заданого значення, затримки другої та третьої черги формуються за рахунок забезпечення гарантованої затримки першої черги. На графіку (рис. 3.21) видно, що відбувається їх рівномірний розподіл і втрат пакетів немає.

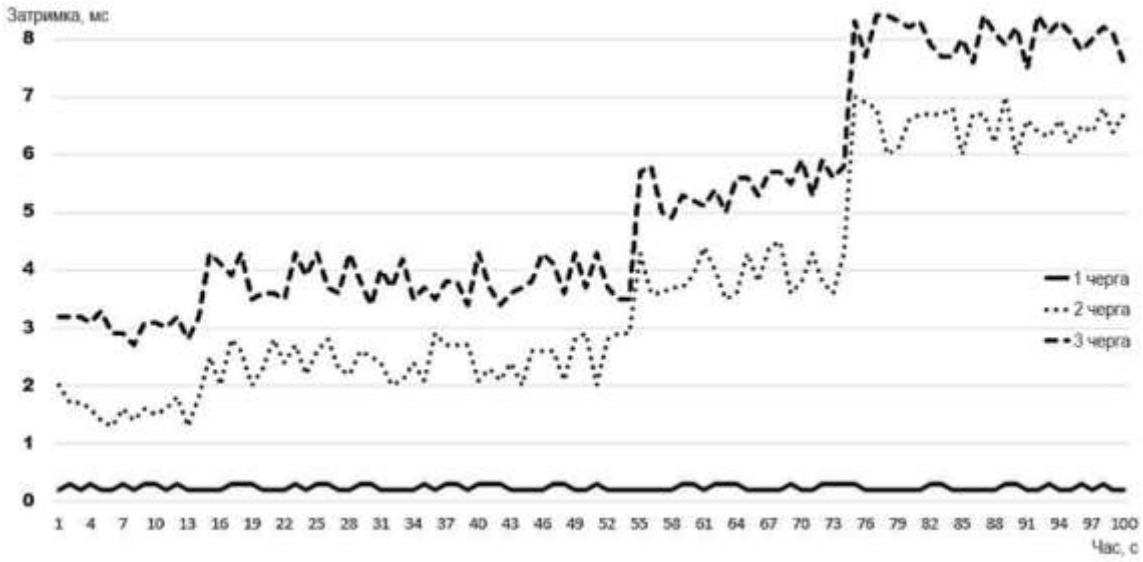


Рис. 3.21. Тестування 2 режиму при модифікованому НТВ

При третьому режимі тестування модифікованого алгоритму НТВ забезпечується гарантована затримка першої черги і зростають значення другої та третьої черги за рахунок підвищення потоку інтенсивності другої черги (рис. 3.22).

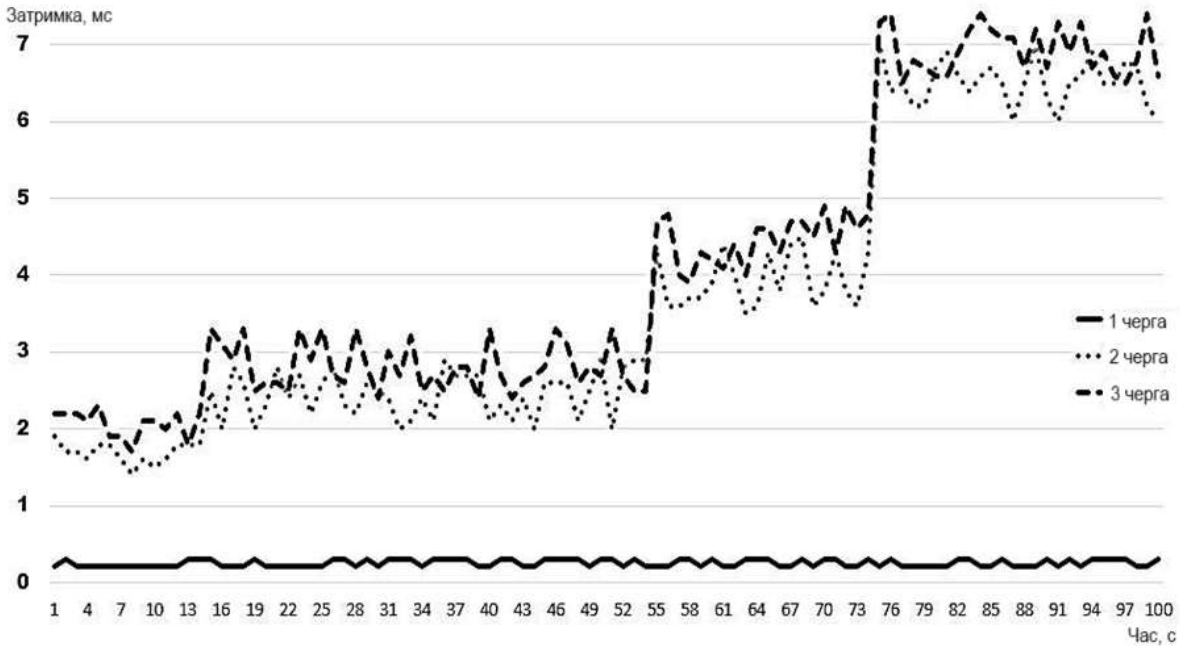


Рис. 3.22. Тестування 3 режиму при модифікованому НТВ

Четвертий режим наочно показує як розподіляється затримка в мережі при початковому значенні інтенсивності. Пріоритетні черги одержують гарантовану затримку, за рахунок найменш пріоритетної третьої черги (рис. 3.23).

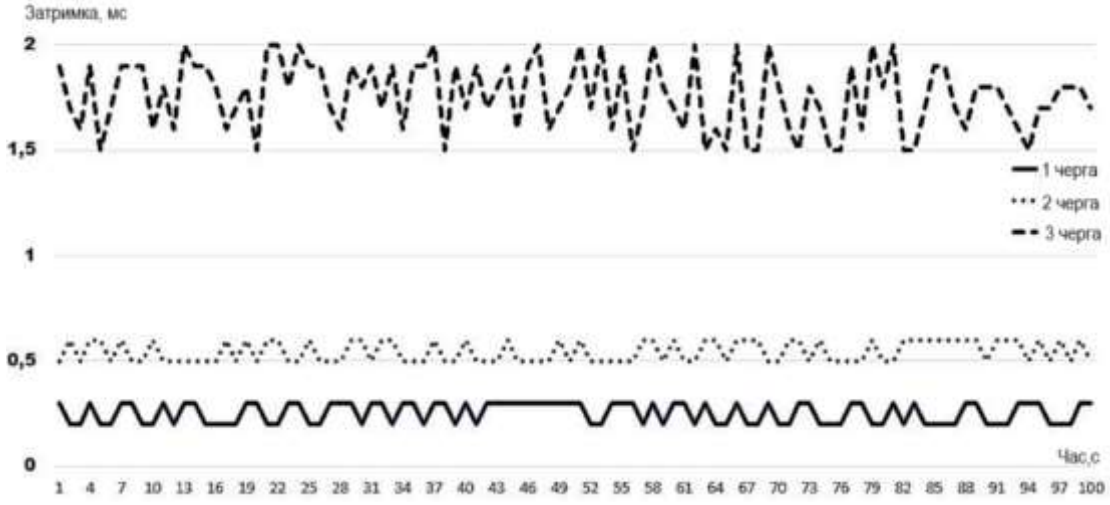


Рис. 3.23. Тестування 4 режиму при модифікованому НТВ

У п'ятому режимі тестування при стандартному НТВ значення затримки рівномірно розподіляється між трьома чергами на будь-якому кроці підвищення інтенсивності, але на останньому кроці через те, що алгоритм намагається

витримувати пріоритети, відбувається переповнення черги. Також значення затримки для будь-яких черг є високими. При модифікованому алгоритмі HTB затримка пакетів першої черги не відхиляється від заданого значення, затримки другої та третьої черги зростають внаслідок загального зростання інтенсивності та забезпечення гарантованої затримки першої черги. На наступному етапі підвищення інтенсивності спостерігається втрата пакетів, але загальна передача трафіку ведеться шляхом відмови від фіксованої затримки першої черги. Оскільки умова переповнення буфера спрацьовує, затримки всіх черг підвищуються залежно від пропускної здатності (рис. 3.24).

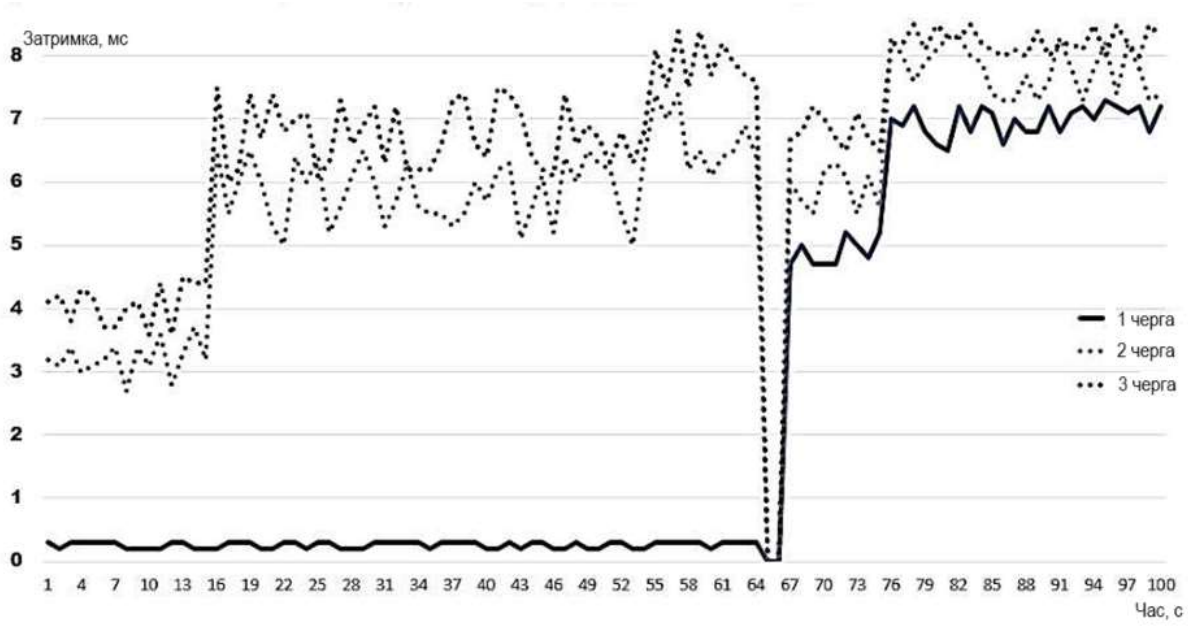


Рис. 3.24. Тестування 5 режиму при модифікованому HTB

У порівнянні з алгоритмом HTB у типовій конфігурації розроблений алгоритм показує нижчі середні значення затримки для різних класів трафіку. Забезпечується гарантована затримка для першої та другої черги. Даний алгоритм можна модифікувати залежно від топології мережі та переваг за пріоритетами трафіку.

Висновки до розділу 3

Експериментально були виявлені фактори, які впливають на затримку передачі пакетів у системі управління трафіком HTB. Основні з них включають контроль інтенсивності вхідного потоку пакетів, динамічну зміну пропускної

здатності каналу відносно вхідної інтенсивності та оптимізацію надання смуги для різних класів трафіку в залежності від вхідної інтенсивності.

Для оцінки ефективності розроблених алгоритмів було проведено імітаційне моделювання їх роботи у середовищі AnyLogic та порівняно з алгоритмом HTB у типовій конфігурації. Результати порівняльного тестування свідчать про те, що розроблені алгоритми демонструють менші сумарні значення затримки для різних класів трафіку, що гарантує високу доступність сервісів. Відповідно, запропоновані алгоритми планування черг передачі даних та підтримки низькопріоритетних сервісів у умовах домінування високопріоритетних сервісів дозволяють оптимізувати використання пропускної здатності та забезпечувати мінімально можливу затримку для пріоритетних класів трафіку.

Також було розроблено програмне забезпечення у вигляді модуля ядра операційної системи Linux, що реалізує алгоритм планування черг передачі даних у програмно-керованих мережах. Проведено дослідження розробленого модуля на експериментальному стенді, і його результати підтвердили отримані під час моделювання.

РОЗДІЛ 4
ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1. Охорона праці

Дослідження методів керування пріоритизацією трафіку в комп’ютерних мережах проводились з дотриманням правил та норм охорони праці і вимог техніки безпеки.

Охорона життя і здоров’я людини є пріоритетним напрямком соціальної політики держави. В Україні прийнято закон прямої дії «Про охорону праці», який регламентує захист конституційного права працівників на безпечні умови праці. Законодавство України про охорону праці складається із загальних законів України та спеціальних законодавчих актів. Загальними законами України, що визначають основні положення з охорони праці є Конституція України, Закон України «Про охорону праці», Кодекс законів про працю (КЗпП), Закон України «Про загальнообов'язкове державне соціальне страхування від нещасного випадку на виробництві та професійного захворювання, які спричинили втрату працездатності».

Одним із найбільш важливих нормативних документів щодо забезпечення охорони праці користувачів ПК є "Державні санітарні норми і правила роботи з візуальними дисплейними терміналами (ВДТ) електронно-обчислювальних машин" ДСанПіН 3.3.2.007-98. Дотримання даних правил значно знижує наслідки несприятливої дії на працівників шкідливих та небезпечних факторів, які супроводжують роботу з 106 відео-дисплейними матеріалами, зокрема можливість зорових, нервово-емоційних переживань, серцево-судинних захворювань. Виходячи з цього, роботодавець повинен забезпечити гігієнічні й ергономічні вимоги щодо організації робочих приміщень для експлуатації електронно-обчислювальних машин (ЕОМ) з ВДТ, робочого середовища, робочих місць з ЕОМ, режиму праці і відпочинку при роботі з ЕОМ тощо, які викладені у нормах НПАОП 0.00-7.15- 18.

При виконанні досліджень методів та засобів пріоритизації, які передбачали використання ПК, площа та об'єм для одного робочого місця оператора

визначається згідно вимог СанПіН 3.3.2-007-98 «Державні санітарні правила і норми. Гігієнічні вимоги до організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин», зокрема площа повинна становити не менше 6,0 м², об'єм - не менше 20,0 м³, відстань робочого місця від стіни повинна складати 1м, а відстань між робочими місцями повинна становити 1,7 м.

При виборі кімнат для розміщення робочих місць ПК враховано ступінь відбиття світла на екранах дисплеїв, яке проходить через вікна і яке може викликати значне осліплення в тих, хто сидить перед ними, особливо влітку та в сонячні дні. Тому, ПК і оргтехніка розміщені біля стін, які не знаходяться біля вікон або навпроти них.

Оскільки, при незадовільному освітленні знижується продуктивність праці користувачів ПК, і можливі негативні впливи на здоров’я такі, як короткозорість, швидка втомленість, тому всі приміщення, які облаштовані робочими місцями з ПК, мають природне і штучне освітлення. Не допускається розташування робочих місць з ПК в підвальних приміщеннях.

Штучне освітлення у приміщеннях повинно бути виконано у вигляді комбінованої системи освітлення з використанням люмінесцентних джерел світла у світильниках загального освітлення, які розташовувати над робочими поверхнями у рівномірно-прямокутному порядку. Штучне освітлення забезпечує на робочих місцях з ПК освітленість 300 – 500 Лк.

Для запобігання засвітленню екранів ПК прямими світловими потоками лінії світильників розташовані з достатнім бічним зміщенням відносно рядів робочих місць, а також паралельно до світлових отворів. При цьому кожне вікно повинно мати світлорозсіюючі штори з коефіцієнтом відбивання 0,7.

Отже, при дослідженні методів та засобів оптимізації передачі трафіку в SDN мережах, проаналізовано та враховано необхідні вимоги щодо охорони праці при використанні електронно-обчислювальної техніки і забезпечено умови для зручної та ефективної роботи працівників.

4.2. Безпека в надзвичайних ситуаціях

4.2.1. Державна система моніторингу довкілля, як складова частина національної інформаційної інфраструктури, сумісної з аналогічними системами інших країн

Державна система моніторингу довкілля - це система спостережень, збирання, оброблення, передавання, збереження та аналізу інформації про стан довкілля, прогнозування його змін і розроблення науково-обгрунтованих рекомендацій для прийняття рішень про запобігання негативним змінам стану довкілля та дотримання вимог екологічної безпеки.

Відповідно до Положення про державну систему моніторингу довкілля [24] (затверджене постановою Кабінету Міністрів України від 30.03.1998 №391) спостереження за станом об’єктів навколишнього природного середовища та рівнем його забруднення здійснюють 8 суб’єктів моніторингу довкілля - центральні органи виконавчої влади (Мінприроди, МНС, МОЗ, Мінагрополітики, Держжитлокомунгоспом, Держводгоспом, Держкомземом, Держкомлісгоспом) та їх органи на місцях, а також підприємства, установи та організації, що належать до сфери їх управління, які є суб'єктами системи моніторингу за загальнодержавною і регіональними (місцевими) програмами реалізації відповідних природоохоронних заходів.

Система моніторингу спрямована на:

– підвищення рівня вивчення і знань про екологічний стан довкілля;

– підвищення оперативності та якості інформаційного обслуговування користувачів на всіх рівнях;

– підвищення якості обґрунтування природоохоронних заходів та ефективності їх здійснення;

– сприяння розвитку міжнародного співробітництва у галузі охорони довкілля, раціонального використання природних ресурсів та екологічної безпеки.

Основними завданнями суб'єктів системи моніторингу є:

– довгострокові систематичні спостереження за станом довкілля;

– аналіз екологічного стану довкілля та прогнозування його змін;

– інформаційно-аналітична підтримка прийняття рішень у галузі охорони довкілля, раціонального використання природних ресурсів та екологічної безпеки;

– інформаційне обслуговування органів державної влади, органів місцевого самоврядування, а також забезпечення екологічною інформацією населення країни і міжнародних організацій.

Система моніторингу ґрунтується на використанні існуючих організаційних структур суб'єктів моніторингу і функціонує на основі єдиного нормативного, організаційного, методологічного і метрологічного забезпечення, об'єднання складових частин та уніфікованих компонентів цієї системи.

Організаційна інтеграція суб'єктів системи моніторингу на всіх рівнях здійснюється органами Мінприроди на основі:

– загальнодержавної і регіональних (місцевих) програм моніторингу довкілля, що складаються з програм відповідних рівнів, поданих суб'єктами системи моніторингу;

– укладених між усіма суб'єктами системи моніторингу угод про спільну діяльність під час здійснення моніторингу довкілля на відповідному рівні.

4.2.2.Оцінка стійкості роботи об’єкта до дії проникаючої радіації і радіоактивного забруднення місцевості, які виникають після ядерного вибуху

Стійкість роботи об'єкта – це здатність його в надзвичайних ситуаціях випускати продукцію у запланованому обсязі, необхідної номенклатури і відповідної якості, а у випадку впливу на об'єкт вражаючих факторів, стихійних лих та виробничих аварій – в мінімально короткі строки відновити своє виробництво.

Залежить вона від таких основних факторів як розміщення об'єкту, природно-кліматичних умов, надійності захисту працюючих, населення від впливу вражаючих факторів, стійкості управління виробництвом і ЦО, психологічної підготовленості керівного складу, спеціалістів і населення до дій в екстремальних умовах, навченості командно-керівного складу ЦО об'єкту і населення правильно виконувати комплекс заходів цивільної оборони.

Оцінка уразливості об'єкта від радіоактивного забруднення і проникаючої радіації починається з визначення максимальних очікуваних значень рівня радіації і дози проникаючої радіації.

За показник стійкості об'єкта приймається допустима доза радіації, яку можуть одержати люди за час робочої зміни.

Стійкість об'єкта проти радіаційного ураження можна оцінювати у такій послідовності.

Визначити: граничні рівні радіації (Р/год.) на об'єкті, за яких можлива виробнича діяльність у звичайному режимі або в режимах радіаційного захисту; ступінь захищеності працюючих; дози радіації, які може одержати виробничий персонал; втрати сільськогосподарських тварин і зниження їх продуктивності; втрати сільськогосподарських рослин та їх урожайність; втрати і ураження лісових насаджень і в результаті цього зниження господарської діяльності лісогосподарських об'єктів; стійкість роботи сільськогосподарських і лісогосподарських об'єктів.

Після аналізу зробити висновки про очікувані максимальні рівні радіоактивного забруднення території об'єкта і дози проникаючої радіації; ступінь забезпечення захисту працюючих, тварин і обладнання, техніки, урожаю, кормів, води; можливість безперервної стійкої роботи об'єкта за умови, що сумарна доза опромінення працюючих не перевищуватиме допустимої дози; можливість виробництва запланованої, доброякісної продукції тваринництва, рослинництва і лісового господарства та заходи підвищення стійкості роботи об'єкта, підвищення рівня захисту працюючих, сільськогосподарських тварин і продукції тваринництва, рослин і врожаю, води і вододжерел [27].

Отже в даному підрозділі розглянуті такі актуальні теми безпеки в надзвичайних ситуаціях, як виконання Україною міжнародних екологічних зобов’язань що вимагає від національної системи моніторингу довкілля забезпечення достовірності в оцінках показників екологічної ситуації не лише по окремих регіонах України, але й на міжнародному рівні.

Також проведена оцінка стійкості роботи об’єкта до дії проникаючої радіації і радіоактивного забруднення місцевості, які виникають після ядерного вибуху. Було розглянуто причини та види вище перелічених факторів і визначено методи запобігання, профілактики та захисту від них. Були отримані знання, які допоможуть зменшити ризик та запобігти небажаним проблемам зі здоров’ям.

ВИСНОВКИ

В результаті виконання кваліфікаційної роботи можна зробити такі висновки:

Проаналізовано існуючі рішення задачі підвищення доступності корпоративної програмно-керованих телекомунікаційних мереж та її компонентів. Що дало можливість визначитись з завданнями роботи.

Для підвищення доступності мережі запропоновано підвищити показник доступності каналів зв'язку за рахунок застосування нового алгоритму керування потоком.

Розроблено алгоритм планування черг передачі на основі модифікації підходу HTB. Експериментально виявлено фактори, що впливають на затримку передачі пакетів HTB і дозволяють виконувати оптимальне планування: контроль інтенсивності вхідного потоку пакетів, динамічна зміна пропускної здатності каналу щодо вхідної інтенсивності, оптимізація надання смуги для класів трафіку щодо вхідної інтенсивності.

Розроблено алгоритм підтримки низькопріоритетних сервісів в умовах сильного домінування високопріоритетних сервісів, що ґрунтується на перерозподілі токенів управління потоком, що дозволяє забезпечити принцип справедливості щодо всіх сервісів, що працюють у програмно-керованих мережах.

Для перевірки ефективності розроблених алгоритмів було зроблено імітаційне моделювання їх роботи у середовищі AnyLogic у порівнянні з роботою алгоритму HTB у типовій конфігурації. Результат порівняльного тестування алгоритмів дозволяє зробити висновок, що розроблені алгоритми показують нижчі сумарні значення затримки для різних класів трафіку, забезпечуючи тим самим високу доступність сервісів.

Проведено експериментальні дослідження, які підтвердили результати, отримані під час моделювання.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1.Bastian, C., Chernak, S., Herscovici, D., Witkowski, B. Prioritizing local and network traffic: пат. 8972537 США. 2015.

2.Bhandarkar, S., Behera, G., Khan KA Scalability Issues в Software Defined Network (SDN): A Survey // Advances in Computer Science and Information Technology (ACSIT). 2015. Т. 2№. 1. С. 81-85.

3.Devera, M. Hierarchical Token Bucket Theory URL: http://luxik.cdi.cz/~devik/qos/htb (дата звернення: 11.12.2023).

4.Devera, M. HTB Linux queuing discipline manual - user guide URL: http://luxik.cdi.cz/~devik/qos/htb/manual/userg.htm (дата звернення: 11.12.2023).

5.Durvy, M., Diot, C., Taft, N., Thiran, P. Network availability based service differentiation // International Workshop on Quality of Service. – Springer, Berlin, Heidelberg, 2003. – С. 305-325.

6.Guck, JW, Van Bemten, A., Reisslein, M., Kellerer, W. Unicast QoS routing algorithms for SDN: Comprehensive survey and performance evaluation //IEEE Communications Surveys & Tutorials. 2017. Т. 20. №. 1. С. 388-415.

7.Guo, L., Jayasimha, DN, Chan, J. Credit flow control scheme in router with flexible link widths utilizing minimal storage: пат. 8711867 США. 2014.

8.HTB Tools Linux – Scribd URL: https://ua.scribd.com/doc/48399733/38376664-HTB-tools-Linux (дата звернення: 11.12.2023).

9.Jerome, A., Yuksel, M., Ahmed, SH, Bassiouni, M. SDN-base load balancing for multi-path TCP //IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). IEEE, 2018. С. 859 -864.

10.Karakus, M., Durresi, A. Quality of service (QoS) в software defined networking (SDN): A survey //Journal of Network and Computer Applications. 2017. Т. 80. С. 200 - 218.

11.Keith, S. Prioritizing classes of network traffic to provide a predetermined quality of service: пат. 9130864 США. 2015.

12.Keshari, SK, Kansal, V., Kumar, S. A systematic review of quality services (QoS) в software defined networking (SDN) //Wireless Personal Communications. - 2021. - Т. 116. №. 3. С. 2593-2614.

13.Kumar, R. Hasan, M., Padhy, S., Evchenko, K. End-to-end network delay guarantees for real-time systems using SDN // 2017 IEEE Real-Time Systems Symposium (RTSS). IEEE, 2017. С. 231-242.

14.Lin, C., Wang, K., Deng, G. A QoS-aware routing в SDN hybrid networks //Procedia Computer Science. 2017. Т. 110. С. 242 -249.

15.Mondal, A., Misra, S., Maity, I. Buffer size evaluation of openflow systems in software-defined networks // IEEE Systems Journal. 2018. Т. 13. №. 2. С. 1359-1366.

16.Nencioni, G., Helvik, BE, Gonzalez, AJ, Heegaard, PE, Kamisinski, A. Availability modelling of software-defined backbone networks // 2016 46 IEEE, 2016. - С. 105 -112.

17.Stanwood, K.L., Gell, D., Bao Y. Systems and methods for prioritizing and scheduling packets in a communication network: пат. 8665724 США. 2014.

18.Y.1540. Служба передачі даних по міжмережевому протоколу (IP) – Параметри робочих характеристик перенесення та доступності IP-пакетів. 2016.

19.Y.1541. Вимоги до мережевих показників якості для служб, які базуються на протоколі IP. 2011.

20.Zhou, Y., Wang, Y., Yu, J., Ba, J., Zhang, S. Load balancing for multiple controllers in SDN based on switches group //2017 19th Asia-Pacific Network Operations and Management Symposium (APNOMS) . IEEE, 2017. С. 227 -230.

21.Глоба, Анна Андріївна. Аналіз методів забезпечення QoS в мультисервісних мережах доступу. BS thesis. Київ, 2023.

22.Дячук О.А., Жаровський Р.О Управління потоком за критеріями доступності. Матеріали XI науково-технічної конференції Тернопільського національного технічного університету імені Івана Пулюя «Інформаційні моделі системи та технології» (13-14 грудня 2023 року). Тернопіль: ТНТУ. 2023. С. 151.

23.Дячук О.А., Жаровський Р.О. Використання SDN для оптимізації передачі даних в комп’ютерних мережах. Матеріали XI науково-технічної конференції

Тернопільського національного технічного університету імені Івана Пулюя «Інформаційні моделі системи та технології» (13-14 грудня 2023 року). Тернопіль: ТНТУ. 2023. С. 149-150.

24. Про затвердження Положення про державну систему моніторингу довкілля. URL: https://zakon.rada.gov.ua/laws/show/391-98-%D0%BF#Text (дата звернення: 11.12.2023).

25.Тиш Є. В., Жаровський Р. О. Методичні вказівки до виконання лабораторних робіт з курсу «Надійність, контроль, діагностика та експлуатація ЕОМ» для студентів денної форми навчання за спеціальністю 123 «Комп'ютерна інженерія». 2019.

26.Фризюк, Микола Олександрович. Спосіб балансування навантаження в програмно-конфігурованій мережі за допомогою генетичного алгоритму. MS thesis. КПІ ім. Ігоря Сікорського, 2023.

27. Цивільна оборона та цивільний захист. URL: https://pidru4niki.com/14210923/bzhd/otsinka_urazlivosti_obyekta_vid_radioaktivnogo_zabrudnennya_pronikayuchoyi_radiatsiyi (дата звернення: 11.12.2023).

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

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

Додаток А.

Тези конференцій

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ПУЛЮЯ

М А Т Е Р І А Л И

XI НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ

«ІНФОРМАЦІЙНІ МОДЕЛІ, СИСТЕМИ ТА ТЕХНОЛОГІЇ»

13-14 грудня 2023 року

ТЕРНОПІЛЬ
2023

УДК 004.45
О.А. Дячук; Р.О. Жаровський , к.т.н.
(Тернопільський національний технічний університет імені Івана Пулюя, Україна)

ВИКОРИСТАННЯ SDN ДЛЯ ОТИМІЗАЦІЇ ПЕРЕДАЧІ ДАНИХ В КОМП'ЮТЕРНИХ МЕРЕЖАХ

O.A. Diachuk; R.O. Zharovskyi, Ph.D.

USING SDN TO OPTIMIZE DATA TRANSMISSION IN COMPUTER NETWORKS

На сучасному етапі розвитку мережевих технологій існує необхідність у розробці заходів контролю передачі трафіку. Така можливість реалізована у мережевій технології SDN. Актуальність дослідження даної теми обумовлена тим, що дана технологія є інноваційною, але не є широко розповсюдженою.

Software-defined network (SDN), або програмно-конфігурована мережа – це підхід до управління мережею, що передбачає розділення рівня контролю мережі та рівня передачі даних [1].

Даний тип мережі є новою технологією, призначеною замінити фізичний дизайн мережі мережевою інфраструктурою, керованою програмним забезпеченням. Зазвичай це виявляється практичним, порівняно економічно ефективним та динамічним рішенням.

SDN-мережа застосовується у різноманітних галузях: програми для спільної роботи, конвергентне сховище, мережевий обмін, організація послуг мобільної мережі, масштабовані мережі центрів обробки даних [2].

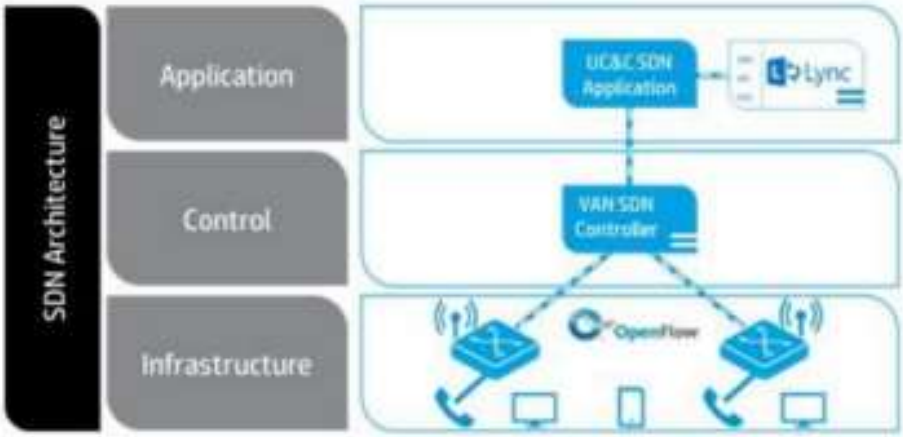


Рисунок 1. Архітектура SDN мережі

Логічна модель SDN-мережі складається з наступних компонентів [3]:

- додатки SDN: додатки, які предоставляють кінцевим користувачам бажані сервіси. Додатки SDN містять ряд вимог до стану і поведінки мережевої інфраструктури;
- контролер SDN виступає єдиною централізованою точкою управління, яка взаємодіє з рівнем додатків за допомогою відкритого інтерфейсу API, а також виконує моніторинг і управління фізичними приладами мережі за допомогою відкритого інтерфейсу – протоколу OpenFlow. Контролер складається з декількох модулів або рівнів, кожен модуль відповідає за ряд необхідних функціональних можливостей;
- OpenFlow комутатори забезпечують безпосередню взаємодію продуктів всієї мережевої інфраструктури з рівнем управління. Комутатор містить одну або кілька таблиць переадресації (flowtable), які містять всі дані про потоках інформації, що

передасться. Записи у таблицях переадресації містять набір полів з інформацією щодо кожного пакету (номер вхідного і вихідного порту, пріоритет переданих даних, лічильник і види дій, які необхідно виповнити після обробки пакета (перенаправлення, модифікація або скидання);

- FlowVisor є відповідальним за розподіл керуючої інформації між потоками даних. За своєю природою FlowVisor – це прозорий проксі-сервер між коммутаторами і контролером. При цьому FlowVisor визначає, які потоки відносяться до тієї чи іншої мережі (комутатора) і, отже, передають керуючу інформацію певного конролера. FlowVisor забезпечує віртуалізацію потоків керуючих пакетів в окремі мережі (slices), кожен з таких потоків має свою логіку управління і передачі;

- компоненти управління і адміністрування – це набір статичних даних, які включають обробку зовнішніх даних: координацію політик і правил, встановлених при проектуванні бізнес-моделі архітектури SDN, початкова конфігурація обладнання і правила розподіл мережевих ресурсів.

SDN принципово відрізняється від звичайної мережі шляхом отримання представлення про мережу: у звичайних мережах сама мережа отримує представлення про додатки, в той час як у програмно-конфігурованій мережі додатки визначають представлення мережі [4].

SDN-мережа прямо описує вимоги до мережі, на відміну від звичайних додатків, що можуть описати вимоги до мережі поступово, в декілька етапів, та потребують обробки спеціалістом. В якості прикладу можна навести перевірку наявності ресурсів та керування політикою конфіденційності для підтримки додатків.

Також звичайні мережі не можуть надати інформацію та стан мережі додаткам, що їх використовують. SDN-підхід дозволяє додаткам відстежувати стан мережі, та адекватно реагувати на зміни.

Виходячи з наведеного вище опису, SDN є комплексною системою взаємодії елементів як логічної, так і фізичної природи, що має складну архітектуру. Даний тип мережі залишається новою технологією, призначеною замінити фізичний дизайн мережі мережевою інфраструктурою, керованою програмним забезпеченням. Зазвичай це виявляється практичним, порівняно економічно ефективним та динамічним рішенням.

Література

1. Shukla, Prashant Kumar, et al. Traffic flow monitoring in software-defined network using modified recursive learning. Physical Communication, 2023, 57: 101997.

2. Yaroshevych, R.; Kovalenko, A. Аналіз технологій підвищення ефективності Тактильного Інтернету у комп'ютерних мережах. Системи управління, навігації та зв'язку. Збірник наукових праць, 2022, 1.67: 106-110.

3. Коробейнікова, Т.; Калько, Т.; Лужецька, Н. Розгляд архітектури програмно-керованих мереж. Grail of Science, 2023, 28: 228-237.

4. Abdou, AbdelRahman; Van Oorschot, Paul C.; Wan, Tao. A framework and comparative analysis of control plane security of SDN and conventional networks. arXiv preprint arXiv:1703.06992, 2017.

УДК 004.45

О.А. Дячук; Р.О. Жаровський, к.т.н.
Тернопільський національний технічний університет імені Івана Пулюя, Україна

УПРАВЛІННЯ ПОТОКОМ ЗА КРИТЕРІЯМИ ДОСТУПНОСТІ

O.A. Diachuk; R.O. Zharovskyi, Ph.D.

FLOW CONTROL BY ACCESSIBILITY CRITERIA

Одним із ефективних способів підвищення доступності в КМ є впровадження технології QoS. Використання алгоритмів пріоритизації та контролю трафіку (ПКТ) дозволяє відокремлювати трафік функціонуючих сервісів із загального потоку та забезпечувати гарантовану смугу пропускання для них. Конфігурування даних алгоритмів здійснюється на підставі виділення частини КС для різних типів трафіку. Однак, такий розподіл не враховує особливості трафіку, що проходить, і не гарантує забезпечення доступності трафіку, чутливого до затримок каналу. Тим самим, доступність сервісу знижується, що веде до потенційної загрози безпеці системи. Цю проблему можна вирішити шляхом розробки алгоритмів ПКТ, що дозволяють оптимально розподіляти трафік, що проходить через КС, з метою підвищення доступності певних типів трафіку.

Під підвищенням доступності каналів зв'язку ми розумітимемо гарантоване попадання в задані часові інтервали (директивний час), що забезпечується за допомогою зменшення часу відгуку сервісу шляхом мінімізації часу обробки пакетів. Часові інтервали, що задаються, необхідні для кожного типу трафіку (сервісу), вказуються в SLA, далі називатимемо ці інтервали максимальним директивним часом

Дослідження показують пряму залежність відгуку сервісу від використовуваного алгоритму планування черг. Найчастіше для класифікації трафіку використовується алгоритм планування Hierarchical Token Bucket. HTB призначений для поділу смуги пропускання між різними типами трафіку, кожен з яких може отримати частку гарантованої смуги пропускання. Алгоритм передбачає класифікацію трафіку за певними ознаками, такими як: IP-адреса призначення чи джерела, порт призначення чи джерела, протокол передачі і т.д. Кожен клас відповідає певному типу трафіку та має свій пріоритет відповідно до SLA. Кожен клас має своє чергу накопичення пакетів, у своїй алгоритм HTB вибудовує класи як дерева.

Клас служби визначає параметри контролю, такі як максимальна пропускна здатність максимальний розмір пакета та використовує дисципліну черги для забезпечення дотримання цих правил. Планувальник та клас пов'язані один з одним, а правила, визначені класом, мають бути пов'язані з визначеною чергою.

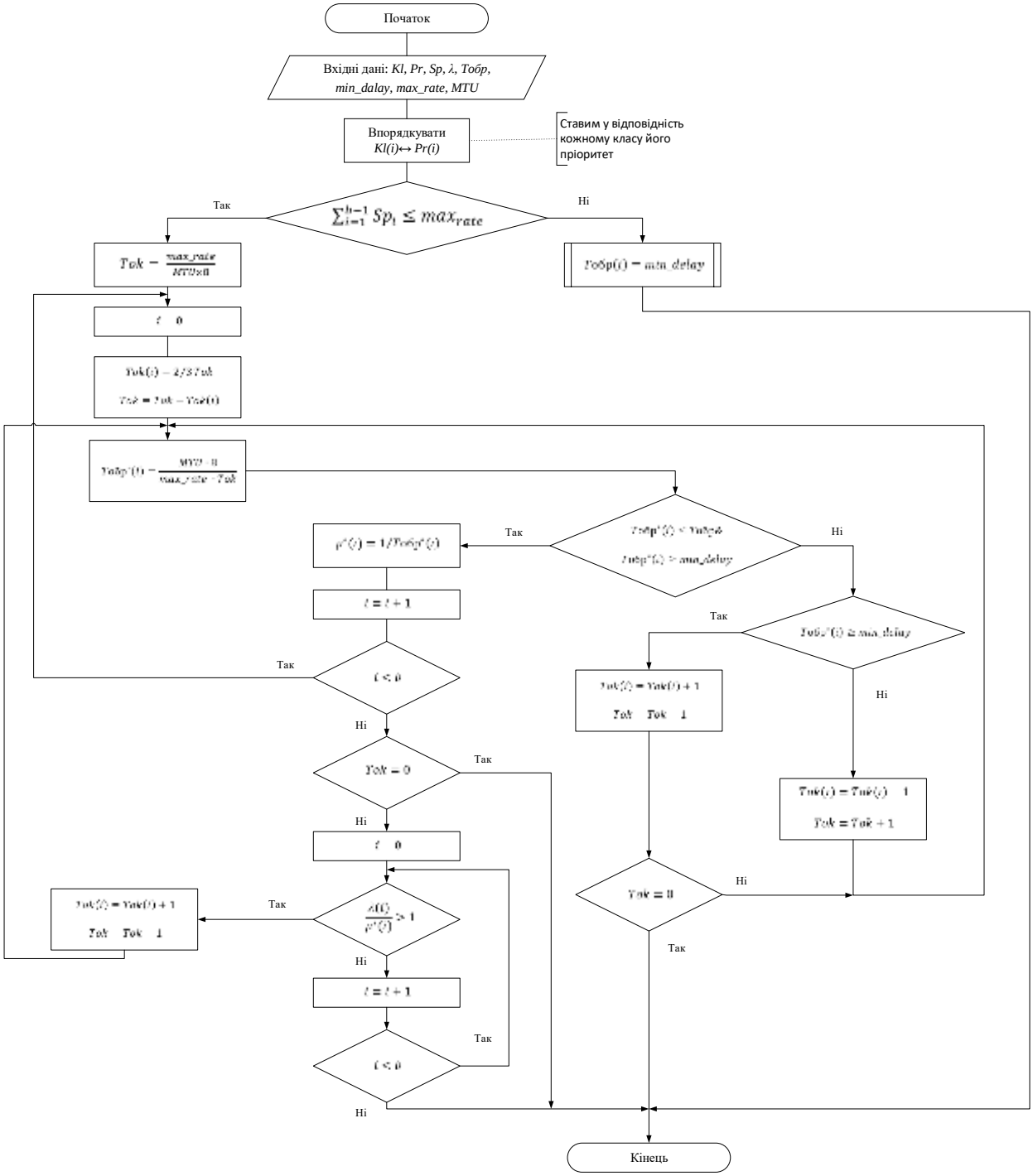
Дисципліна HTB управляє потоком мережевих пакетів шляхом виділення токенів на їх передачу відповідно до пріоритетів. Будь-який дочірній клас, який хоче запозичувати токен, буде вимагати його у свого батьківського класу, який, у свою чергу, може запозичувати у свого батьківського класу, поки токен не буде знайдений або кореневий клас не буде досягнутий. Шейпінг відбувається лише у листових класах.

HTB гарантує, що пропускна здатність, яка надається кожному класу, становить принаймні мінімум призначеного йому значення (limit). Коли клас вимагає меншу смугу пропускання, то надлишок розподіляється серед інших класів, які вимагають обслуговування.

Таким чином для підвищення доступності мережі можна скористатись підвищенням показників доступності каналів зв'язку за рахунок застосування нового алгоритму керування потоком передачі даних в комп'ютерних мережах.

Додаток Б.

Блок-схема алгоритму планування черг



Додаток В.

Модифікований модуль HTB ядра ОС Linux

```
     static struct htb_class *htb_classify(struct sk_buff *skb, struct
Qdisc *sch, int *qerr)
     {
      struct htb_sched *q = qdisc_priv(sch); struct htb_class * cl;
struct tcf_result res; struct tcf_proto *tcf;
      int result;
     if (skb->priority == sch->handle)
       return HTB_DIRECT; /* X:0 ( direct flow) selected */ cl =
htb_find(skb->priority, sch);
      if (cl && cl->level == 0)
              return cl;
      *qerr=NET_XMIT_SUCCESS | __NET_XMIT_BYPASS; tcf = q-
>filter_list;
      while (tcf && (result = tc_classify(skb, tcf, &res)) >= 0) {
     #ifdef CONFIG_NET_CLS_ACT switch (result) { case TC_ACT_QUEUED:
case TC_ACT_STOLEN:
      *qerr=NET_XMIT_SUCCESS | __NET_XMIT_STOLEN; case TC_ACT_SHOT:
return NULL;
          }
     #endif

     static struct htb_class *htb_classify(struct sk_buff *skb, struct
Qdisc *sch,
                int *qerr)
     {
      struct htb_sched *q = qdisc_priv(sch); struct htb_class * cl;
struct tcf_result res; struct tcf_proto *tcf; int result;
      if (skb->priority == sch->handle)
       return HTB_DIRECT; /* X:0 ( direct flow) selected */ cl =
htb_find(skb->priority, sch);
      if (cl && cl->level == 0)
              return cl;

      *qerr=NET_XMIT_SUCCESS | __NET_XMIT_BYPASS; tcf = q-
>filter_list;
      while (tcf && (result = tc_classify(skb, tcf, &res)) >= 0) {
     #ifdef CONFIG_NET_CLS_ACT switch (result) { case TC_ACT_QUEUED:
case TC_ACT_STOLEN:
      *qerr=NET_XMIT_SUCCESS | __NET_XMIT_STOLEN; case TC_ACT_SHOT:
return NULL;
          }
     #endif

     static void htb_add_to_wait_tree(struct htb_sched *q,
          struct htb_class *cl, long delay)
     {
      struct rb_node **p = &q->wait_pq[cl->level].rb_node, *parent =
NULL;
```

```
      cl -> pq_key = q-> now + delay; if (cl->pq_key == q->now) cl-
>pq_key++;

      /* update the nearest event cache */ if (q->near_ev_ cache[ cl-
>level] > cl->pq_key)
      q-> near_ev_cache[ cl->level] = cl->pq_key;

      while (*p) {
                  struct htb_class *c;
               parent = * p;
      c = rb_ entry ( parent, struct htb_class, pq_node);
       if (cl->pq_key >= c->pq_key) p = &parent->rb_right; else
            p = &parent->rb_left;
     }
     rb_link_ node( &cl->pq_node, parent, p);
     rb_insert_ color( &cl->pq_node, &q->wait_pq[cl->level]);
     }
     static void htb_activate_prios(struct htb_sched *q, struct
htb_class *cl)
     {
      struct htb_class *p = cl->parent;
      long m, mask = cl-> prio_activity;

      while (cl->cmode == HTB_MAY_BORROW && p&& mask) {
      m = mask; while(m) { int prio = ffz( ~m);
             m &= ~(1 << prio);
       if (p->un.inner.feed[prio].rb_node) /* parent already has its
feed in use so that
      * reset bit in mask as parent is already ok
         */
             mask &= ~(1 << prio);

      htb_add_to_id_ tree( p->un.inner.feed + prio, cl, prio);
          }
                    p -> prio_activity | = mask;
             cl = p;
       p = cl->parent;
     }
      if (cl->cmode == HTB_CAN_SEND && mask)
      htb_add_class_to_ row( q, cl, mask);
     }
     #if OFBUF
         if( cl != HTB_DIRECT && cl) skb_get(skb);
     #endif

      if (cl == HTB_DIRECT) { /* enqueue to helper queue */
                          if (q->direct_queue.qlen <q->direct_qlen) {
           __skb_queue_ tail( &q->direct_queue, skb);
                     q ->direct_pkts++;
              } else {
            kfree_ skb ( skb);
             sch -> qstats.drops++;
```

```
            return NET_XMIT_DROP;
          }
     #ifdef CONFIG_NET_CLS_ACT
      } else if ( !cl ) {
       if (ret & __NET_XMIT_BYPASS) sch->qstats.drops++;
kfree_skb(skb);
            return ret;
     #endif
      } else if ((ret = qdisc_ enqueue ( skb, cl->un.leaf.q))! =
NET_XMIT_SUCCESS) {
     #if OFBUF
     __skb_queue_ tail( &q->ofbuf, skb); q->ofbuf_queued++;
     #else
       if (net_xmit_drop_count(ret)) { sch->qstats.drops++;
            cl ->qstats.drops++;
          }
            return ret;
     #endif
      } else {
      bstats_ update( &cl->bstats, skb); htb_activate(q, cl);
     #if OFBUF
     kfree_ skb ( skb);
     #endif
      }
     sch ->q.qlen++;
      return NET_XMIT_SUCCESS;
     }
     static inline void htb_accnt_tokens(struct htb_class *cl, int
bytes, long diff)

     static inline void htb_accnt_ctokens(struct htb_class *cl, int
bytes, long diff)
     {
      long toks = diff + cl-> ctokens;
     if (toks > cl-> cbuffer)
                toks = cl-> cbuffer;
     toks -= (long) qdisc_l2t(cl->ceil, bytes); if (toks <= -cl-
>mbuffer)
                  toks = 1 - cl->mbuffer;
      cl -> ctokens = toks;
     }
```