

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Методи буферизації пакетів в маршрутизаторах
IP-мереж

(тема)

Виконав:

студент II курсу, групи КСМм-23-1
Вірко А.О.
(прізвище, ініціали)

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

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

Освітня програма Комп'ютерні системи та мережі
(повна назва освітньої програми)

Керівник: доц. Янковський О.А.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ЕОМ

(підпис)

Коваленко А.А.

(прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерної інженерії та управління _____

Кафедра _____ електронних обчислювальних машин _____

Рівень вищої освіти _____ другий (магістерський) _____

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

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

Освітня програма _____ Комп'ютерні системи та мережі _____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

“ _____ ” _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студенту _____ Вірку Артему Олексійовичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи Методи буферизації пакетів в маршрутизаторах IP-мереж

затверджена наказом по університету від “ 22 ” листопада 2024 р. № 1237 Ст

2. Термін подання студентом роботи до екзаменаційної комісії _____ 20 січня 2025 р.

3. Вхідні дані до роботи _____

1) Моделі та методи для керування мережевими інформаційними потоками

2) Сучасні вимоги до мережних показників

3) Перелік використаних програмних та апаратних засобів: OpNet 14, NS-3

4. Перелік питань, що потрібно опрацювати у роботі _____

1) Аналіз сучасного стану проблеми

2) Огляд технологій управління перевантаженням та чергами маршрутизаторів

3) Моделі управління мережним трафіком

4) Вибір програмних та апаратних засобів моделювання

5) Проведення експериментальних досліджень

6) Висновки

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) 17 слайдів

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

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

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз стану проблеми та сучасних методів її вирішення	26.11.24–27.11.24	
2	Огляд технологій управління перевантаженням	28.11.24–02.12.24	
3	Розробка моделі управління мережним трафіком	03.12.24–11.12.24	
4	Вибір програмних та апаратних засобів моделювання	12.12.24–16.12.24	
5	Тестування запропонованого метода	17.12.24–23.12.24	
6	Оформлення пояснювальної записки	24.12.24–03.01.25	

Дата видачі завдання 25 листопада 2024 р.

Студент _____
(підпис)

Керівник роботи _____
(підпис)

доц. Янковський О.А.
(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 77 с., 20 рис., 1 табл., 1 дод., 20 джерел.

АДИТИВНЕ ЗБІЛЬШЕННЯ/МУЛЬТИПЛІКАТИВНЕ ЗМЕНШЕННЯ, АКТИВНЕ ТА ПАСИВНЕ УПРАВЛІННЯ ЧЕРГОЮ, ПЕРЕВАНТАЖЕННЯ, ПРОТОКОЛ, СКИДАННЯ ПАКЕТІВ, ТАЙМ-АУТ.

Метою кваліфікаційної роботи є поліпшення продуктивності комп'ютерних мереж у вигляді застосування розширених моделей масового обслуговування для організації черг маршрутизаторів.

У ході виконання кваліфікаційної роботи розглянуто різноманітні архітектури мережі, основні поняття теорії масового обслуговування, якості обслуговування в мережах, ролі моделей масового обслуговування для підтримки QoS та різних методів активного керування чергами.

Представлено розширена модель справедливої черга без збереження стану з нечітким виявленням перевантаження. Вдосконалена модель WFQ, що поєднує функції зваженої справедливої черги з низькою затримкою і технологію скидання пакетів.

ABSTRACT

Master's thesis: 77 pages, 20 figures, 1 tables, 1 appendices, 20 sources.

ADDITIVE INCREASE/MULTIPLICATIVE DECREASE, ACTIVE AND PASSIVE QUEUE MANAGEMENT, OVERLOAD, PROTOCOL, PACKET DROP, TIMEOUT.

The purpose of the qualification work is to improve the performance of computer networks in the form of the application of advanced mass service models for the organization of router queues.

In the course of the qualification work, various network architectures, the basic concepts of mass service theory, quality of service in networks, the role of mass service models for QoS support, and various methods of active queue management were considered.

An extended stateless fair queue model with fuzzy congestion detection is presented. An improved WFQ model combining the features of weighted fair queuing with low latency and packet drop technology.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	8
ВСТУП	9
1 МЕТА КВАЛІФІКАЦІЙНОЇ РОБОТИ	10
2 ОГЛЯД СУЧАСНОГО СТАНУ ПРОБЛЕМИ	11
2.1 Бурхливість Інтернет-трафіку	12
2.2 Уникнення бурхливості	17
2.2.1 Алгоритми Token Bucket і Leaky Bucket	18
2.2.2 Швидкісний TSP	21
3 ЧЕРГИ МАРШРУТИЗАТОРІВ	24
3.1 Поняття черги	24
3.2 Архітектура мережі	25
3.3 Теорія черги	27
3.4 Якість обслуговування	28
3.5 Моделі черги	28
4 ПОКРАЩЕНА СПРАВЕДЛИВА ЧЕРГА З ВИЯВЛЕННЯМ ПЕРЕВАНТАЖЕННЯ НА ОСНОВІ НЕЧІТКИХ ПАРАМЕТРІВ	33
4.1 Справедливі черги	33
4.2 Огляд безкласової черги	34
4.3 Недоліки перевантаження мереж	35
4.4 Система нечіткої логіки	36
4.5 Метод справедливої черги з множинної чергою	37
4.6 Критерії максимально-мінімальної справедливості	38
4.6.1 Швидкість прибуття потоку	39
4.6.2 Оцінка справедливої швидкості з'єднання	39
4.6.3 Загальна швидкість прибуття	39
4.6.4 Обчислення справедливості	40

4.7 Розробка алгоритму ECSFQ-FBCD	41
4.8 Розширена основна модель черги (ECSFQ)	42
4.8.1 Оцінка затримки та втрат	42
4.8.2 Класифікація потоку за вхідними вузлами.....	44
4.8.3 Пріоритетний планувальник	45
4.9 Виявлення перевантаження на основі нечіткої моделі (FBCD)	45
4.9.1 Швидкість надходження пакетів	46
4.9.2 Затримка в черзі	46
4.9.3 Фазифікація.....	48
4.9.4 Нечіткі набори правил	49
4.9.5 Дефазифікація.....	49
4.10 Імітаційне моделювання.....	50
4.11 Результати моделювання.....	51
4.11.1 Імітаційна модель і параметри.....	51
4.11.2 Показники продуктивності	52
4.11.3 Вплив різної пропускної здатності вузького місця	52
5 СПРАВЕДЛИВА ЧЕРГА З НИЗЬКОЮ ЗАТРИМКОЮ ТА ДИФЕРЕНЦІЙНИМ ВІДКИДАННЯМ ПАКЕТІВ	55
5.1 Огляд зваженої справедливої черги	55
5.1.1 Проблеми зваженої справедливої черги	56
5.1.2 Черга з низькою затримкою	56
5.2 Модель.....	57
5.2.1 Операції WFQ.....	57
5.2.2 Планувальник LLWFQ	58
5.3 Диференційна обробка відкидання	59
5.4 Результати моделювання.....	62
ВИСНОВКИ.....	65
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	66
ДОДАТОК А Графічний матеріал кваліфікаційної роботи.....	68

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ
І ТЕРМІНІВ

AQM – алгоритм активної черги (англ., Active Queue Mechanism)

AWFQ – адаптивна зважена справедлива черга (англ., Adaptive Weighted Fair Queuing)

BE – найкраще рішення (англ., Best Effort)

CBWFQ – зважена справедлива черга на основі класу (англ., Class Based Weighted Fair Queuing)

EFWFQ – покращена зважена справедлива черга (англ., Enhanced Weighted Fair Queuing)

FQ – справедлива черга (англ., Fair Queuing)

FR – справедлива швидкість потоку (англ., Fair Rate of Flow)

FWFQ – справедлива зважена черга (англ., Fair Weighted Fair Queuing)

OWFQ – опортуністична зважена справедлива черга (англ., Opportunistic Weighted Fair Queuing)

QD – затримка в черзі (англ., Queuing Delay)

RED – випадкове раннє виявлення (англ., Random Early Detection)

RTT – випадкове раннє виявлення (англ., Round Trip Time)

UDP – протокол дейтаграм користувача (англ., User Datagram Protocol)

WFQ – зважена справедлива черга (англ., Weighted Fair Queuing)

ВСТУП

Оскільки сучасний Інтернет розвивається, існує постійно зростаючий попит на високу пропускну мережеву здатність.

Цей попит частково зумовлено широко розповсюдженими відеододатками реального часу, такими як різноманітні онлайн-ігри, телеконференції або потокове відео високої чіткості. Сучасні оптичні комутатори та маршрутизатори вже давно розглядаються як багатообіцяюче технічне рішення для зростаючого попиту на трафік. Але, існуючі розміри буферів оптичних комутаторів та маршрутизаторів дуже обмежені через різноманітні труднощі виробництва великих буферів.

З іншого боку, сучасний Інтернет-трафік дуже стрімкий. Наявність стрибків мережевого трафіку була доказана для всіх масштабів часу, від кількох десятків мілісекунд до годин. Широко поширена проблема має дуже значний вплив на показники продуктивності мереж з малим буфером, що в кінцевому підсумку призводить до значної ймовірності скидання пакетів у чергах і низького показника використання каналу.

У літературі було запропоновано багато рішень для вирішення проблеми розриву мережевого трафіку. Методи організації інформаційних потоків, наприклад формування трафіку, були доступні в різноманітних маршрутизаторах або комутаторах вже з періоду мереж асинхронного режиму передачі. Крім того, вдосконалення протоколу TCP, як необхідне вирішення проблеми зниження швидкості TCP, давно досліджується. Також, запропоновано кілька новітніх методів формування та планування трафіку для скоординованої передачі трафіку в загальній мережі. Але, всі існуючі такі рішення недостатні для забезпечення ефективного вирішення проблеми переривання високошвидкісної передачі.

1 МЕТА КВАЛІФІКАЦІЙНОЇ РОБОТИ

Контроль перевантажень для IP-мереж був постійною проблемою протягом багатьох років. Протягом останнього десятиліття контроль перевантажень TCP (протокол керування передачею) використовувався для адаптивного контролю швидкості окремих з'єднань, які спільно використовують мережеві посилення IP.

Але контроль перевантаження TCP використовує механізм Drop-tail, у якому пакети відкидаються, коли буфер заповнюється. Основним недоліком цього методу є те, що джерело TCP знижує швидкість передачі лише після того, як пакети відкидаються. Тому для подолання цього недоліку пропонуються різні алгоритми черги, такі як Red, Green, SFB тощо.

Отже, очевидно, що впровадження нових технологій, які покращують загальну продуктивність мережі, яка складається з трафіку в реальному часі та трафіку не в реальному часі, має велике значення.

Таким чином, метою магістерської кваліфікаційної роботи є розробка нових методів організації черг маршрутизаторів для боротьби з перевантаженнями в комп'ютерних мережах із використанням набору характеристик трафіку, для чого необхідно:

- провести всебічний аналіз методів організації черг маршрутизаторів;
- провести огляд чинників, впливаючих на появу перевантаження;
- провести аналіз існуючих моделей та методів управління трафіком;
- запропонувати метод організації черг маршрутизатора;
- провести імітаційне моделювання запропонованого методу;
- провести аналіз отриманих результатів.

2 ОГЛЯД СУЧАСНОГО СТАНУ ПРОБЛЕМИ

Постійно існує великий попит на дуже значну пропускну здатність мереж завдяки швидкому зростанню обсягів на трафік, що передається в глобальному Інтернеті. Зараз більшість сучасної інфраструктури мереж складається з електронних маршрутизаторів та комутаторів, які забезпечують наскрізні з'єднання між клієнтами. Проте дослідники вказують на можливість того, що існуючі «вузькі місця» в сучасних маршрутизаторах зрештою приведуть до обмеження пропускну здатності мереж [1]. Також, проблеми з споживанням енергії та розсіюванням виділеного тепла стали головною проблемою під час розгортання великих маршрутизаторів.

Тому, оптичні маршрутизатори вже досить давно розглядаються як можливе рішення для задоволення дуже швидкого зростання попиту на збільшення пропускну здатності мереж та подолання проблеми розсіювання тепла в маршрутизаторах [2, 3].

Значною особливістю оптичного маршрутизатора є те, що пакети передаються на майже всьому шляху через маршрутизатор лише в оптичній формі. Це потребує буферизації пакетів в маршрутизаторах у вигляді світла. Найпоширенішими сучасними методами організації оптичного буфера є використання волоконно-оптичних ліній затримки [2, 4] та ліній затримки повільного світла. Тому, ці методи дуже обмежують розміри оптичних буферів і роблять їх досить малими. Використання таких малих буферів в оптичних мережах було обумовлено теоретичним аналізом та експериментальними висновками.

Результати високої продуктивності мереж, досягаються лише тоді, коли сеанси ТСП керуються певним механізмом контролю швидкості, відомим як темп ТСП, або канали доступу мають пропускну здатність, набагато меншу, ніж вузьке місце в мережі. Це вказує на те, що в мережі з малим буфером час обертання є необхідною технікою для запобігання високої ймовірності

відкидання пакетів на оптичних маршрутизаторах/комутаторах із малими буферами. Однак, як показ час, без ретельно розроблених модифікацій ці існуючі методи формування трафіку не здатні ефективно зменшити ймовірність скидання пакетів у малих буферах маршрутизаторів. Будучи природним рішенням для вирішення проблем ТСП, швидкість ТСП бере свій початок у протоколах явного керування швидкістю, відмінних від ТСП [5]. Однак існуючі результати показують, що темпові потоки ТСП мають меншу частку вузьких місць, ніж нерегульовані потоки ТСП.

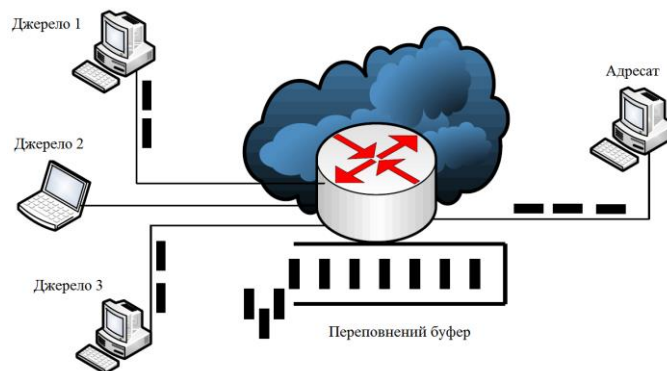


Рисунок 2.1 – Перевантаження через переповнення буфера в маршрутизаторі

Ця ослаблена конкурентна здатність ТСП з стимуляцією проти ТСП без стимуляції та перешкоджає широкому застосуванню ТСП з підтримкою стимулювання. Інші методи, запропоновані в [6], покладаються або на попередні статистичні дані трафіку, або на планування глобальної мережі.

2.1 Бурхливість Інтернет-трафіку

Самоподібність Інтернет-трафіку вказує на те, що його стрибкоподібність проявляється в широкому діапазоні часових масштабів, від десятків мілісекунд до кількох хвилин і навіть годин [7]. Сплески Інтернет-трафіку умовно поділяються на два класи: «довгострокові» і «короткострокові». Довгострокова та короткочасна нестабільність в

основному викликана атрибутами користувача/сеансу на макроскопічному рівні та механізмами контролю перевантаження TCP на мікроскопічному рівні, відповідно. Зокрема, довготривала стрибкоподібність відноситься до часових масштабів від сотень мілісекунд до десятків хвилин, а короткочасна стрибкоподібність – до сотень мілісекунд.

Часові масштаби дуже важливі в управлінні комп'ютерною мережею. У контексті IP Quality of Service (QoS) визначено три часові рамки:

- мілісекунди;
- 100 мілісекунд;
- 10 секунд і більше.

Перший часовий проміжок такий, що затори в основному викликані короткочасними спалахами окремих потоків трафіку або сукупного трафіку, де обсяг трафіку перевищує доступну смугу пропускання. Механізми QoS, що мають відношення до цього періоду часу, включають черги, планування та методи відкидання.

Другий часовий інтервал визначає час зворотного зв'язку мережі (RTT), який важливий для додатків із замкнутим циклом на основі TCP. Активне керування чергами (AQM) як технологія контролю перевантаження працює в межах цього періоду часу.

Третій часовий інтервал стосується управління довгостроковими середніми показниками та пропускну здатністю мережевого трафіку, що досягається за допомогою планування пропускну здатності та проектування трафіку. Щоб зафіксувати поведінку глобального масштабування (тобто довгострокову різкість), відображену в трафіку локальної мережі (LAN) і глобальної мережі (WAN), використовують модель генератора ієрархічного трафіку HTTP на основі TCP. Зокрема, ієрархія моделі трафіку складається з ряду сеансів TCP, кожен з яких містить певну кількість сторінок.

Атрибути користувач/сеанс у термінах міжсеансового часу, сторінок за сеанс, міжсторінкового часу, об'єктів на сторінку, міжоб'єктного часу та об'єктного часу явно вказуються як параметри в цій моделі. За допомогою

відповідних налаштувань ця модель може успішно відтворити поведінку глобального масштабування, а саме властивість самоподібності, що демонструється в отриманих даних мережевого трафіку.

Можна стверджувати, що замкнуті механізми контролю перевантажень у TCP є основною причиною короткочасного розриву IP-трафіку [8]. Протокол керування передачею (TCP) – це протокол ковзного вікна.

З'єднання TCP складається з відправника та одержувача з обох сторін з'єднання. У будь-який конкретний момент часу відправник підтримує дві змінні стану, розмір вікна перевантаження та розмір вікна отримання, позначені CWND і RWND відповідно, де RWND оголошується одержувачем. Отримавши підтвердження (ACK), відправник надсилає один або кілька сегментів, залежно від фази TCP-сеансу.

Отримавши сегмент, одержувач надсилає назад підтвердження, яке підтверджує останній отриманий пакет. Кумулятивний сегмент визначається як такий, для якого всі попередні сегменти успішно надійшли до одержувача. Протягом кожного часу зворотного зв'язку (RTT) кількість непідтверджених сегментів, що передаються, обмежена мінімумом CWND і RWND. RWND використовується для контролю потоку [9]. Якщо припустити, що RWND настільки великий, що CWND завжди менший за RWND протягом усього часу сеансу TCP, кількість непідтверджених сегментів обмежується виключно CWND.

Алгоритм керування перевантаженням TCP має три основні компоненти: повільний запуск, уникнення перевантаження та реакція на події тайм-ауту.

Повільний старт

Коли починається з'єднання TCP, значення CWND ініціалізується до одиниці. Тут припускається, що одиниця CWND дорівнює розміру пакета, а всі пакети мають однакові розміри. Після отримання підтвердження CWND збільшується на одиницю. Відправник продовжує збільшувати свій CWND таким чином, доки не буде виявлено подію втрати пакету, у цей час CWND

скорочується наполовину, а фаза повільного початку завершується. Під час фази повільного запуску CWND подвоюється кожен RTT, починаючи з одного, що називається експоненціальним зростанням вікна перевантаження. Сеанс TCP переходить у фазу уникнення перевантажень одразу після завершення фази повільного запуску.

Подія втрати пакета визначається отриманням трьох послідовних підтверджень з однаковим підтвердженням порядковим номером, відомих як дублікати ACK.

Уникнення заторів

Під час фази уникнення перевантаження, після отримання підтвердження, CWND збільшується на $1/CWND$. Після виявлення події втрати пакета CWND зменшується до $CWND/2$. Правила, що визначають зміни вікна перевантаження у відповідь на підтвердження або втрату пакета, відомі як правила адитивного збільшення та мультиплікативного зменшення (AIMD).

Термін AIMD походить від того факту, що згідно з цим механізмом CWND збільшується на одиницю кожного RTT і зменшується на фіксовану частку CWND у разі втрати пакета. Сеанс TCP продовжує працювати в межах фази уникнення перевантажень, доки не настане подія тайм-ауту або з'єднання не буде явно розірвано.

Реакція на події таймауту

Сеанс TCP встановлює таймер для кожного надісланого пакета, і таймер запускається, щойно відповідний пакет надіслано. Якщо підтвердження цього пакету повертається відправнику до закінчення таймера, таймер деактивується та скидається.

В іншому випадку, якщо таймер закінчиться, відбудеться подія тайм-ауту, яка зменшує CWND назад до одиниці, незалежно від того, наскільки великий поточний CWND.

З цього моменту сеанс TCP переходить у фазу повільного запуску, починаючи з CWND одиниці.

Звісно, що механізми контролю перевантажень TCP можуть спричинити зрив TCP-трафіку. Автори представили в [10] три аспекти механізму контролю перевантажень TCP, які призводять до розриву трафіку TCP. Це повільний старт, втрати та стиснення АСК.

У ситуації, коли вузьке місце обслуговує лише один сеанс TCP, пропускна здатність сеансу TCP обмежена швидкістю зв'язку вузького місця. Вузьке місце зв'язку може бути насичене пакетами сеансу TCP. Таким чином, підтвердження також надсилаються назад відправнику зі швидкістю вузького місця зв'язку. Поки пов'язаний буфер не заповнений, вікно перевантаження сеансу TCP може збільшуватися під час створення черги. У цьому випадку пакети в межах одного RTT рівномірно розподіляються по всьому RTT, і, як наслідок, TCP-трафік не демонструє вибухів. Це явище відоме як синхронізація АСК. Незважаючи на те, що синхронізація АСК фактично згладжує TCP-трафік, це трапляється рідко, оскільки через мультиплексування швидкість з'єднання вузьких місць значно вища за доступну пропускну здатність окремого TCP-з'єднання.

Під час фази повільного старту, після отримання підтвердження, відправник надсилає два пакети. Без втрати загальності можна припустити, що в момент часу t $CWND=W$.

Також припустимо, що в цей час пропускна здатність TCP-сеансу нижча, ніж його доступна пропускна здатність B , тобто $(W/RTT) < B$, де RTT – це час зворотного проходження TCP-сеансу в момент часу t . З припущенням, що жодні пакети не відкидаються, доки буфер у зв'язку з вузьким місцем не заповнений, усі W -пакети надходять до одержувача зі швидкістю B . У результаті всі W -підтвердження надсилаються назад відправнику також зі швидкістю B . Оскільки надходження кожного підтвердження ініціює надсилання двох пакетів, отримані пакети $2W$ надсилаються зі швидкістю надсилання $2B$. Слід зазначити, що навіть незважаючи на те, що ці пакети $2W$ надсилаються зі швидкістю $2B$, вони надходять до одержувача зі швидкістю B , оскільки B є швидкістю вузького

місця з'єднання. У межах того самого RTT пакети W буферизуються у зв'язку з вузьким місцем, тоді як інші пакети W доставляються відправнику зі швидкістю B . Оскільки вікно перевантаження постійно збільшується від RTT до RTT, кількість пакетів, що зберігаються в буфері зв'язку з вузьким місцем, також збільшується.

Оскільки на практиці розмір буфера у зв'язку з вузьким місцем безсумнівно обмежений, кількість пакетів, що зберігаються в буфері, завжди може досягти ліміту, і тоді відбудеться відкидання пакета з новим пакетом, який надходить до повного буфера. Таке скидання пакетів завершує етап повільного запуску.

Одним із способів виявлення відправником втрат є отримання дублікатів АСК. Після успішної повторної передачі втраченого пакета наступне підтвердження одержувача підтверджуватиме не лише втрачений пакет, але й інші пакети, успішно отримані одержувачем. Коли підтвердження надійде до відправника, йому буде дозволено надіслати групу пакетів.

Стиснення АСК відноситься до ситуації, коли купа підтвердження надходить до відправника бурхливим способом. Така поведінка може бути викликана підтвердженнями в черзі на деяких проміжних маршрутизаторах через перевантаження на зворотному шляху підключення ТСП. Навіть вважаючи, що підтвердження в ідеалі можуть розподілятися рівномірно на швидкості вузького місця зв'язку, такий ефект синхронізації АСК може бути послаблений або навіть усунутий перевантаженням, яке досить часто виникає на шляху проходження туди й назад ТСП.

2.2 Уникнення бурхливості

Механізми IP QoS використовуються, щоб гарантувати, що мережа може надавати певні рівні послуг, які відповідають вимогам, зазначеним в угодах про рівень обслуговування (Service level agreements – SLA).

Бурхливість, як характеристика IP-трафіку, має тісний зв'язок із показниками, визначеними в SLA, оскільки ступінь, в якому трафік є пакетним, суттєво впливає на значення показників. В IP-мережах метриками, визначеними в SLA, є затримка, тремтіння, втрата пакетів, пропускна спроможність, доступність послуг і збереження послідовності потоків.

Звісно, що за однакової середньої швидкості, чим більший трафік, тим довша черга. Як наслідок, пакети, що проходять через маршрутизатори, відчують триваліші затримки, більші тремтіння, вищу ймовірність відкидання пакетів і, як наслідок, страждають від низької пропускної здатності.

Оскільки пакетність має значний вплив на показники SLA, існує ряд механізмів і методів, розроблених для обробки пакетності IP-трафіку, таких як контроль і формування. Алгоритми маркерного відра та витоку відра є двома основними алгоритмами, які використовуються для реалізації технології контролю трафіку та його формування. Крім того, оскільки TCP є основним транспортним протоколом, по якому передається більша частина Інтернет-трафіку, швидкість TCP також привернула велику.

2.2.1 Алгоритми Token Bucket і Leaky Bucket

Алгоритми маркерного відра та витоку відра широко використовуються як методи контролю трафіку або формування для встановлення максимального обмеження швидкості основного трафіку, який може бути окремим потоком, класом трафіку, що складається з кількох потоків, що задовольняють певним критеріям класифікації, або навіть сукупний трафік за посиланням.

Однак терміни «токен-відро» та «діряве відро» плутають і часто помилково використовують. Для уточнення вони представлені разом. Алгоритм відра маркерів можна використовувати для забезпечення максимальної швидкості потоку трафіку, навіть незважаючи на те, що

блокувальник і формувальник принципово відрізняються у своїй обробці пакетів. У подальшому спочатку представимо концепцію алгоритму відра маркерів, а потім проілюструємо, як він застосований для реалізації обмежувача та формувача.

Алгоритм Token Bucket

В алгоритмі відра маркерів є відро з глибиною B (у байтах). Токени розміром 1 байт додаються до відра зі швидкістю R (у байтах за секунду). Маркери можна додавати щоразу, коли обробляється пакет, або через регулярні проміжки часу, залежно від конкретної реалізації. У будь-якому випадку швидкість, з якою токени додаються у відро, має дорівнювати R . Якщо токен потрапляє у відро, заповнене токенами, токен буде викинуто. Таким чином, гарантується, що кількість маркерів (у байтах) завжди менше або дорівнює B . На рисунку 2.2(a) показано структуру сегмента маркерів.

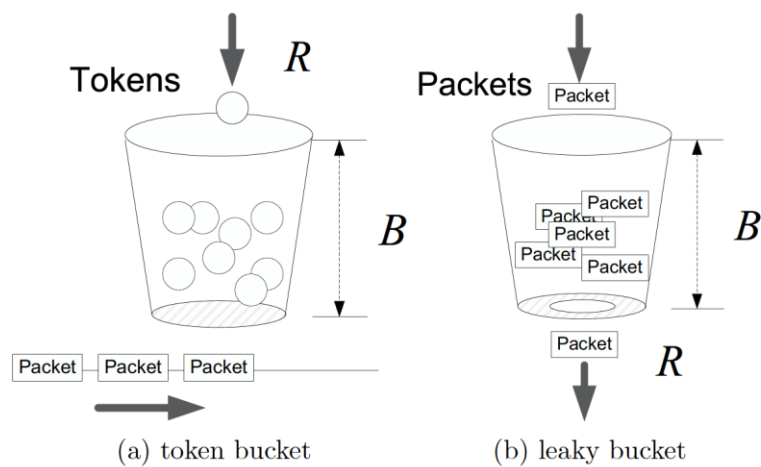


Рисунок 2.2 – Алгоритми Token Bucket і Leaky Bucket

Коли алгоритм сегмента маркерів застосовується до потоку трафіку, кожен пакет, що належить до потоку трафіку, просить сегмент маркерів надати йому кількість маркерів, що дорівнює його розміру (у байтах). Якщо у відрі залишається достатня кількість маркерів для пакета, тоді цей пакет відповідає визначенню відра маркерів.

Якщо у відрі залишилося менше маркерів, тоді цей пакет перевищив визначення відра маркерів. Для пакетів, які відповідають визначенню відра маркерів, і блокувальник на основі маркерів, і формувач негайно пересилають їх. Для пакетів, які перевищують визначення відра маркерів, блокувальник на основі відра маркерів просто відкидає їх, а формувальник на основі маркерів поміщає їх у виділену чергу. Щоразу, коли у відрі є більше доступних токенів, формувач витягує пакети, призначає їм токени та пересилає їх. Тут можна побачити, що різниця між блокувальником і формувальником полягає в тому, чи зберігаються «надлишкові» пакети для пізньої передачі.

Слід зазначити, що максимальна швидкість R , яку забезпечує блок-блок маркерів або блок-формувальник, є максимальною середньою швидкістю, тобто протягом тривалого періоду максимальна середня швидкість трафіку, контрольована або сформована алгоритмом сегментів маркерів, дорівнює R . Миттєва пропускна здатність контрольованого або сформованого трафіку може бути вищою, ніж R . Наприклад, якщо серія пакетів надходить у відро, повне маркерів один за одним зі швидкістю, вищою за R , тоді вони будуть перенаправлені з блокувальника або формувача знову з тією ж швидкістю, доки сума розмірів цих пакетів буде меншою за глибину відра B .

На відміну від алгоритму маркерного відра, алгоритм дірявого відра використовується лише у формувачі та суворо обмежує миттєву пропускну здатність сформованого трафіку до R , що є максимальною швидкістю, з якою пакети можуть витікати з відра. В алгоритмі дірявого відра також є відро з глибиною B (у байтах). Унизу є отвір, який дозволяє пакетам витікати зі швидкістю R (у байтах). Якщо відро повне, коли до нього надходить пакет, пакет скидається. Формувальник на основі дірявого відра працює точно так само, як черга «першим прийшов-перший вийшов» (FIFO) з вихідною швидкістю R і обмеженням черги B . На рисунку 2.2(b) показано структуру формувача на базі витоку.

Відмінності між формувачем на основі відра токенів і формувачем на основі відра з витокком підсумовуються наступним чином. По-перше, замість маркерів, що зберігаються у відрі маркерів, пакети зберігаються у дірявому відрі. Немає поняття токена у випадку алгоритму витокку відра взагалі. По-друге, максимальна пікова швидкість у формувальнику на основі відра може перевищувати R , тоді як максимальна пікова швидкість у формувачі на основі дірявого відра суворо обмежена R . Найвідомішим прикладом алгоритму витокку є алгоритм Generic Cell Rate Algorithm (GCRA), який використовується для формування трафіку в мережах АТМ.

2.2.2 Швидкісний TCP

Рівномірне надсилання TCP вивчалось деякий час як природне рішення для подолання стрімкого трафіку TCP. Рівномірне надсилання TCP спочатку було запропоновано, щоб виправити стиснення підтвердження через перехресний трафік. З того часу були проведені дослідження, які пропонували різні способи використання пейсингу в TCP для різних цілей, таких як компенсація АСК-тактування при повільному запуску і швидке відновлення після втрати пакета.

Надсилання пакетів більше не запускається надходженням підтвердження. Натомість із заданим $CWND$ і заданим RTT пакети планується рівномірно надсилати протягом усього RTT , тобто відправник надсилає один пакет за час $RTT/CWND$.

Після отримання підтвердження $CWND$ оновлюється так само, як і в звичайному TCP (Reno). Оскільки для обчислення інтервалу $RTT/CWND$ необхідна детальна оцінка RTT , для отримання точних зразків RTT увімкнено параметр позначки часу TCP.

Крім того, оцінка RTT обчислюється за допомогою алгоритму експоненційного зваженого ковзного середнього (EWMA). Щоразу, коли оновлюється $CWND$ або RTT , інтервал перераховується, а потім

застосовується до наступних передач пакетів. Таким чином, замість того, щоб надсилатися пакетно, пакети рівномірно розподіляються по всьому RTT, усуваючи короточасну пакетність, яка була притаманна звичайному нешвидкісному TCP.

Висновки щодо впливу синхронізації TCP на продуктивність мережі були суперечливими. З одного боку, емпіричні дослідження, вказують на те, що хоча швидкість TCP може підвищити пропускну здатність і справедливість у деяких ситуаціях, вона погіршує продуктивність TCP в цілому. Погана продуктивність пояснюється здебільшого «синхронізованим падінням» і затримками пакетів, які неправильно інтерпретуються як перевантаження.

Більше того, сеанси TCP із швидким темпом є менш конкурентоспроможними, коли конкурують із сеансами TCP без темпу, оскільки, рівномірно розподіляючи пакети по всьому RTT, сеанси TCP із темпом із більшою ймовірністю зіткнуться з відкиданням пакетів, ніж TCP-сеанси без темпу. З іншого боку, темп TCP необхідний для сеансів TCP для досягнення високого рівня використання каналу, коли маршрутизатор оснащений невеликими буферами [11].

Допоможе чи ні покращити темп використання зв'язку залежить від того, якою мірою на швидкість відкидання пакетів впливає короточасний сплеск трафіку. Коли розмір буфера помірний або великий, короточасні пакети поглинаються буфером, і втрати пакетів виникають рідко. Коли розмір буфера невеликий, відкидання пакетів відбувається дуже часто через переповнення буфера. Оскільки короточасна нестабільність зменшується за допомогою синхронізації TCP, ймовірність скидання пакетів зменшується, і, як наслідок, збільшується використання каналу.

Два чинники перешкоджають широкому застосуванню швидкісному TCP в поточному Інтернеті: великі розміри буферів і менші частки пропускну здатності темпованого TCP. У наш час комутатори та маршрутизатори в Інтернеті оснащені буферами емпіричного правила, тобто

продуктом пропускної здатності та затримки. У такій ситуації темп ТСР не перевершує ТСР без темпу. Крім того, враховуючи величезну кількість комп'ютерів у світі та різноманітність операційних систем, що працюють на цих комп'ютерах, навряд чи вдасться стандартизувати та активувати синхронізацію ТСР на всіх комп'ютерах.

Швидкість має вирішальне значення для ефективної роботи майбутніх базових мереж з невеликим буфером, тому що у випадку мереж з малим буфером на використання каналу значно впливає короткострокова перерва.

3 ЧЕРГИ МАРШРУТИЗАТОРІВ

Управління мережею є складною проблемою, яка потребує надійних інтелектуальних методологій для досягнення кращої продуктивності.

Потоки даних, що належать мультимедійним додаткам, набувають сьогодні все більшого значення в комунікаційних мережах. Ключовою характеристикою таких потоків даних є те, що вони вимагають якості обслуговування (QoS). Функція мережевих пристроїв у підтримці QoS полягає в тому, щоб переконатися, що пакети пересилаються ефективно.

QoS відіграє ключову роль в аналізі мережевого трафіку і може бути охарактеризована з точки зору пропускну здатності, затримки та втрат.

Математичні моделі систем масового обслуговування відіграють важливу роль в аналізі, моделюванні та проектуванні різноманітних мереж. Механізми масового обслуговування є важливими для керування перевантаженням мережевого трафіку будь-якого типу в сучасній технологічній ситуації для підтримки QoS. Теорія масового обслуговування ймовірно та математично описує взаємодію між надходженням та обслуговуванням клієнтів з метою підтримки ефективної системи.

Модель черги використовується для опису ситуації в реальному часі, а поведінку черги можна проаналізувати математично шляхом вимірювання різних показників продуктивності.

3.1 Поняття черги

Сучасні комп'ютерні системи складніші та швидше розвиваються, тому потрібні інструменти та методи, які допомагають зрозуміти поведінку цих систем. Моделювання мережі масового обслуговування – це підхід, який широко застосовується при дослідженні таких складних систем. Це особливий підхід, у якому комп'ютерні системи представлені як мережа черг,

де всі елементи мережі, такі як маршрутизатори та комп'ютери, з'єднані, а пакети є клієнтами, які обслуговуються різними елементами мережі. Моделі мереж масового обслуговування можна класифікувати на три групи [12]:

- відкриті мережі масового обслуговування;
- закриті мережі масового обслуговування;
- змішані мережі масового обслуговування.

У закритих мережах масового обслуговування одні і ті ж клієнти залишаються в мережі весь час. Жоден новий клієнт не приєднується і жоден клієнт не залишає мережу. Клієнти, які завершили своє обслуговування в одній системі масового обслуговування, переходять до іншої, потім до іншої і так далі, і ніколи не залишають мережу. У відкритих системах масового обслуговування нові клієнти з-за меж мережі можуть приєднатися до будь-якої черги, і після завершення обслуговування в мережі вони можуть покинути мережу. У змішаних мережах масового обслуговування можуть бути присутні клієнти мережі як відкритого, так і закритого типу.

3.2 Архітектура мережі

Багаторівневою моделлю, яка домінувала в літературі щодо передачі даних і мереж до 1990 року, була модель OSI (Взаємозв'язок відкритих систем). Модель OSI – це багаторівнева структура для проектування мережевих систем, яка забезпечує зв'язок між усіма типами комп'ютерних систем [13]. Модель OSI складається з семи впорядкованих рівнів:

- фізичний рівень (рівень 1);
- канальний рівень (рівень 2);
- мережевий рівень (рівень 3);
- транспортний рівень (рівень 4);
- сеансовий рівень (рівень 5);
- рівень презентації (рівень 6);
- прикладний рівень (рівень 7).

Фізичний рівень координує функції, необхідні для передачі потоку бітів через фізичне середовище. Він також визначає процедури та функції, які мають виконувати фізичні пристрої та інтерфейси для здійснення передачі, а також напрямок передачі між двома пристроями.

Канальний рівень перетворює фізичний рівень, новий засіб передачі, на надійний канал. Канальний рівень відповідає за кадрування, фізичну адресацію, контроль потоку, контроль помилок і контроль доступу.

Рівень третій, мережевий рівень відповідає за доставку пакета від джерела до пункту призначення, можливо, через кілька каналів. На відміну від канального рівня, який контролює доставку пакетів між двома системами в одній мережі, мережевий рівень гарантує, що кожен пакет дістанеться від точки походження до кінцевого пункту призначення. Цей рівень відповідає за логічну адресацію та маршрутизацію.

Транспортний рівень забезпечує доставку всього повідомлення від джерела до адресата. Канальний рівень обробляє кожен пакет незалежно, ніби кожна частина належить до окремого повідомлення, незалежно від того, чи є це насправді. Але транспортний рівень, з іншого боку, гарантує, що все повідомлення надходять в цілості та в порядку, виконує контроль помилок та контроль потоку на рівні джерела до адресата. Основними обов'язками цього рівня є адресація точки обслуговування, сегментація та повторна збірка, контроль з'єднання, контроль потоку та контроль помилок.

Рівень сеансу – це контролер мережевого діалогу. Він встановлює, підтримує та синхронізує взаємодію між комунікуючими системами.

Шостий рівень, який є рівнем представлення, стосується синтаксису та семантики інформації, якою обмінюються дві системи. Цей рівень піклується про переклад, шифрування та стиснення.

Сьомий рівень, який є прикладним рівнем, дозволяє користувачеві чи програмному забезпеченню, отримувати доступ до мережі. Він надає інтерфейс користувача та підтримку таких послуг, як електронна пошта, віддалений доступ до файлів і передача, керування спільною базою даних.

Послуги, що надаються цим рівнем, включають мережевий віртуальний термінал, передачу файлів, доступ і керування, поштові служби та служби каталогів.

3.3 Теорія черги

Піонер теорії масового обслуговування А. К. Ерланг, датський математик, опублікував «Теорію ймовірностей». Він розробив математичний спосіб визначення кількості телефонних операторів, які повинні обслуговувати певну кількість дзвінків. Щоб виміряти ефективність моделі масового обслуговування, потрібно вивчити:

- модель надходження клієнтів;
- схему обслуговування серверів;
- дисципліну черги;
- пропускну здатність системи;
- кількість каналів обслуговування.

У системах масового обслуговування процес надходження є імовірнісним. Схема прибуття визначається розподілом часу між прибуттями, тобто часом між послідовними прибуттями. Прибуття може бути одиночним або гуртовим.

Для випадкових надходжень розглядається розподіл Пуассона. Шаблон обслуговування описує розподіл часу обслуговування, доступні сервери та те, чи з'єднані сервери послідовно чи паралельно. Обслуговування може бути одноразовим або пакетним.

Дисципліна черги стосується того, як клієнти обслуговуються, коли вони чекають у черзі. Найпоширенішими дисциплінами черги є «перший прийшов, перший обслужений» (FIFO), «останній прийшов, перший обслужений» (LIFO), випадкове обслуговування (RS) і пріоритетне обслуговування. Кількість клієнтів, які очікують у черзі, і клієнтів, які обслуговуються разом, називають потужністю системи.

Кількість каналів обслуговування означає кількість паралельних станцій обслуговування, які можуть обслуговувати клієнтів одночасно. Система лише з одним сервером називається односерверною моделлю, а система з двома чи більшою кількістю паралельних серверів називається багатосерверною моделлю. У випадку багатосерверних моделей клієнти можуть сформувати одну чергу або паралельну чергу для отримання послуги.

3.4 Якість обслуговування

QoS – це механізм, який відіграє важливу роль в аналізі комунікаційних мереж. Метою QoS є надання ефективних послуг для програм, які цього потребують, забезпечуючи достатню пропускну здатність, контролюючи затримку та тремтіння, а отже зменшуючи втрату даних. QoS відноситься до набору технологій для управління мережевим трафіком економічно ефективним способом. Дві основні моделі QoS, визначені IETF (Internet Engineering Task Force) для мереж на основі IP, це інтегровані послуги (IntServ) і диференційовані послуги (DiffServ). QoS можна покращити шляхом впровадження відповідного механізму активної черги (AQM). Важливими факторами QoS є пропускну здатність, затримка та втрати.

3.5 Моделі черги

Моделі черги та механізми планування черги забезпечують QoS у комунікаційних мережах, уникаючи або контролюючи перевантаження даних.

Перевантаження виникає, коли пакети надходять на вихідний порт швидше, ніж вони передаються. Проблеми, з якими стикаються мережі, полягають у справедливому розподілі обмеженої кількості ресурсів для користувачів, програм і класів обслуговування. Дисципліна планування черги

дозволяє керувати такою ситуацією та контролювати її. Механізм планування черги може виконувати кілька завдань, таких як справедливий розподіл пропускної здатності для різних класів обслуговування, захист між різними класами обслуговування, як-от погана поведінка класу обслуговування в одній черзі, не може впливати на пропускну здатність і затримку, що надається іншим класам обслуговування. Це також дозволяє отримати доступ до невикористаної смуги пропускання. Є багато моделей масового обслуговування.

Але моделі масового обслуговування з дисциплінами черги «перший прийшов, перший вийшов» (FIFO), пріоритетна черга (PQ) і справедлива черга (FQ) широко використовуються в мережах зв'язку.

Модель черги FIFO є основною моделлю черги, де пакети обслуговуються в порядку їх розміщення. Поведінка FIFO дуже передбачувана; Максимальна затримка визначається максимальною глибиною черги.

Черги FIFO обробляють усі потоки однаково, оскільки середня затримка в черзі для всіх потоків збільшується зі збільшенням перевантаження. У результаті модель FIFO призводить до втрати даних, тремтіння та затримки. У моделі FIFO бурхливий потік може використовувати пропускну здатність усіх інших потоків, і через це послуга може бути недосяжна для інших потоків.

Пріоритети є мірою терміновості або важливості. Пріоритетна черга (PQ) розроблена таким чином, що підтримує диференційовані класи обслуговування. У PQ пакети класифікуються за пріоритетом і розміщуються в чергах з різними пріоритетами. У кожній черзі дотримується порядок FIFO. У PQ можна встановити високі пріоритети для додатків у реальному часі, таких як голос, відео тощо, над додатками, які не працюють у реальному часі. Але основним обмеженням PQ є те, що якщо високопріоритетний трафік обробляється неефективно, то потоки з низьким пріоритетом можуть постраждати від скидання пакетів або великої затримки.

Якщо спробувати реалізувати PQ, щоб розмістити потоки TCP у черзі з високим пріоритетом, ніж потоки UDP, потоки TCP намагатимуться використати всю доступну пропускну здатність, таким чином потоки UDP відчуватимуть притиснення.

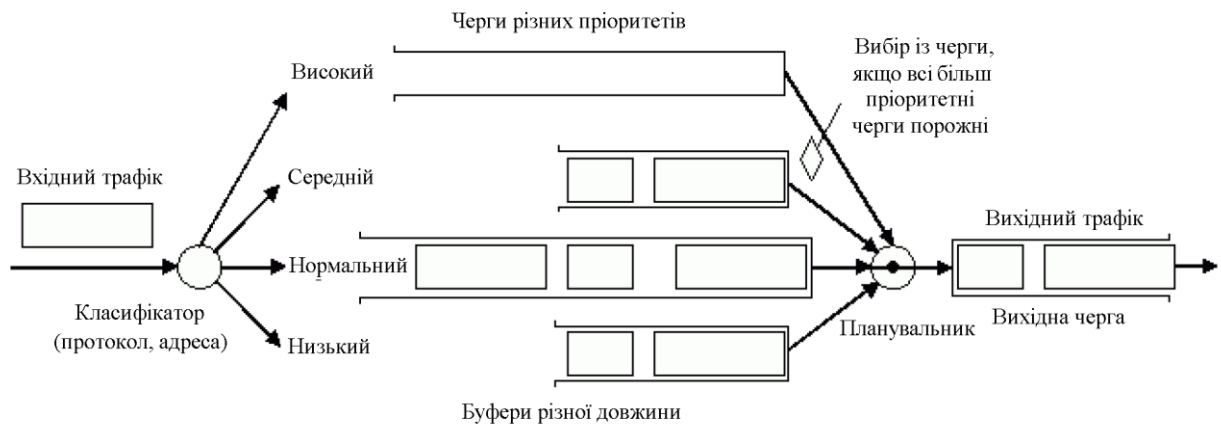


Рисунок 3.1 – Пріоритетна черга

Справедлива черга (FQ) була запропонована Джоном Нейглом у 1987 році. Це механізм черги, який розроблено для того, щоб кожен потік мав справедливий доступ до ресурсів мережі. Це запобігає використанню бурхливими потоками пропускну здатності, яка перевищує виділену пропускну здатність.

Алгоритми FQ виділяють окрему чергу для кожного потоку та планують передачу пакетів через перевантажене з'єднання, щоб кожен потік отримував середню послугу [14].

Якщо потік передає дані з невиправдано високою швидкістю, FQ штрафуює агресивний потік через накопичення та зрештою відкидає його пакети на маршрутизаторі. Оскільки механізм FQ повинен підтримувати стан, керувати буфером або виконувати планування пакетів на основі кожного потоку, його також називають чергою на основі потоку. FQ зазвичай застосовується на краю мережі, де абоненти підключаються до свого постачальника послуг.

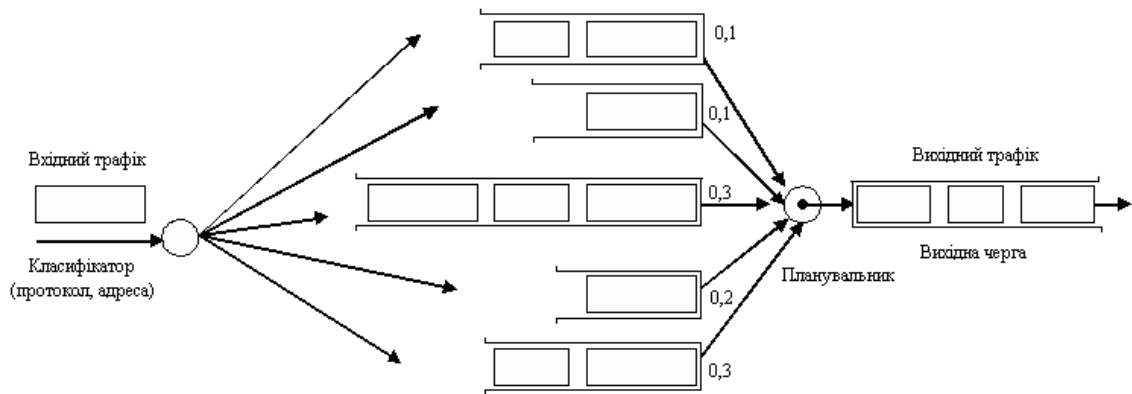


Рисунок 3.2 – Зважені налаштовуванні черги

Коли кількість черг змінюється, пропускна здатність, виділена для кожної з черг, також змінюється, тобто якщо кількість черг збільшується з n до $(n+1)$, розподіл пропускної здатності для кожної з черг зменшується з $1/n$ до $1/(n+1)$.

Справедлива черга – це технологія, яка контролює перевантаження трафіку в мережі, дозволяючи кожному потоку, що проходить через мережевий пристрій, мати справедливу частку мережевих ресурсів. Різноманітні показники, які беруть участь у вимірюванні продуктивності мережі в механізмах FQ, такі:

- затримка пакета – загальний час, потрібний мережі для доставки пакета даних від часу, коли перший біт пакета надходить у мережу, до часу, коли перший біт пакета доставляється до місця призначення;
- пропускна здатність – міра кількості даних, що передаються за одиницю часу, часто вимірюється в пакетах за секунду;
- джиттер затримки – важливий показник у деяких пакетних мережах віртуальних каналів (це міра мінливості часу між послідовними пакетами, доставленими у віртуальному колі);
- імовірність блокування – це фундаментальна метрика більшості мереж, орієнтованих на підключення, таких як комутатор каналів і мережа віртуального комутатора каналів;

- справедливість у використанні мережі – це поняття рівного ставлення до всіх сеансів у мережі;
- складність алгоритму – це міра ефективності схеми масового обслуговування на практиці.

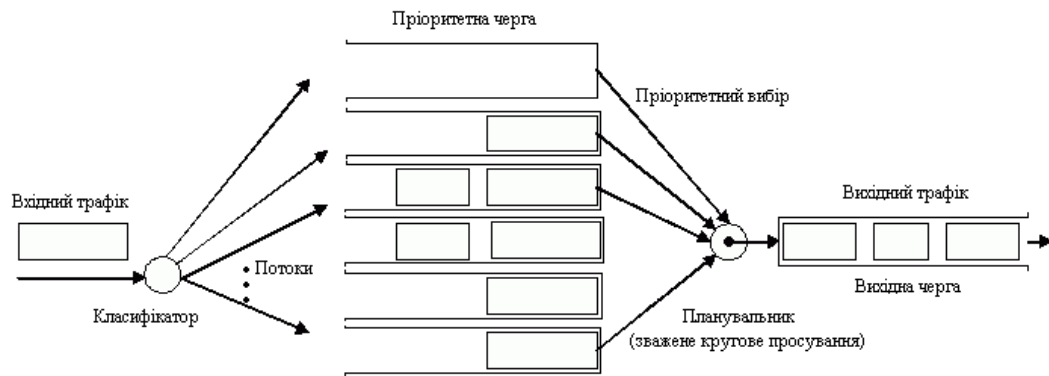


Рисунок 3.3 – Зважене справедливе обслуговування

FQ є складним, оскільки класифікація потоків може бути повільною, а головна проблема полягає в тому, що основний маршрутизатор може підтримувати 50000 – 100000 потоків за певний час. На гігабітних швидкостях може не вистачити часу або пам'яті для керування метаданими кожного потоку. Можна побудувати маршрутизатори з великою кількістю черг. Велика змінна кількість черг особливо складна і складна для реалізації у високошвидкісних маршрутизаторах.

4 ПОКРАЩЕНА СПРАВЕДЛИВА ЧЕРГА З ВИЯВЛЕННЯМ ПЕРЕВАНТАЖЕННЯ НА ОСНОВІ НЕЧІТКИХ ПАРАМЕТРІВ

4.1 Справедливі черги

За останнє десятиліття було запропоновано декілька алгоритмів справедливого обслуговування в чергах, таких як Weighted Fair Queuing (WFQ), Core Stateless Fair Queuing (CSFQ), Class Based WFQ і Multiple Queue Fair Queuing (MQFQ), які дозволяють постачальникам послуг пропонувати таке велике різноманіття послуг.

Серед алгоритмів FQ Core Stateless Fair Queuing (CSFQ) – це розподілений підхід справедливої черги без введення накладних витрат на обробку каналів основної мережі. Обмеження CSFQ включають його нездатність оцінити справедливість під час великих потоків трафіку, які є різкими (наприклад, VoIP або відео), а також він використовує єдину чергу FIFO на основному маршрутизаторі.

Для підвищення справедливості та ефективності мережі в представлено вдосконалену справедливу чергу без збереження стану з нечіткою логікою виявлення перевантажень (Enhanced Core Stateless Fair Queuing-Fuzzy Based Congestion Detection – ECSFQ-FBCD), більш ефективну ніж модель CSFQ. У цій моделі спочатку пріоритетний планувальник застосовується до потоків, що надходять на вхід маршрутизатора. Якщо це потік у реальному часі, тобто VoIP або відеопотік, то пакети класифікуються як нееластичні потоки, а для трафіку не реального часу пакети класифікуються як еластичні потоки. Для нееластичних потоків для виявлення перевантаження використовується метод нечіткого виявлення перевантаження. Якщо ступінь перевантаження є високою або середньою, тоді нееластичні потоки плануються за допомогою MQFQ, а якщо ступінь перевантаження низька, то нееластичні потоки плануються за принципом max-min CSFQ разом із еластичними потоками.

4.2 Огляд безкласової черги

Core Stateless Fair Queuing (CSFQ) – це варіант FQ для подолання обмежень справедливої черги. Основним компонентом CSFQ є те, що лише певні маршрутизатори повинні виконувати спеціальні механізми черги, тоді як більшість інших маршрутизаторів залишаються без змін. Схема справедливої черги Core-Stateless Fair розрізняє основні маршрутизатори, високошвидкісні та завантажені маршрутизатори в ядрі від периферійних маршрутизаторів. У CSFQ лише крайові вузли використовують механізми контролю перевантажень, що дозволяє основним каналам функціонувати в звичайному режимі без шкідливого впливу на існуючі швидкості маршрутизації. Цей підхід має два ключові аспекти.

Основні маршрутизатори в моделі CSFQ не підтримують стан кожного потоку, замість цього використовують інформацію кожного потоку, що передається через мітку в заголовку кожного пакету. Ця мітка містить оцінку швидкості прибуття потоку. Однак вона ініціалізується периферійним маршрутизатором на основі інформації про потік, а потім оновлюється на кожному маршрутизаторі вздовж шляху на основі сукупної інформації на цьому маршрутизаторі. CSFQ використовує чергу FIFO з імовірнісним відкиданням для подолання буферизації потоку та планування. Імовірність скидання пакета є функцією оцінки швидкості, що міститься в мітці, і справедливого розподілення на цьому маршрутизаторі, яка оцінюється на основі вимірювань сукупного трафіку.

Ця технологія дозволяє уникнути як необхідності підтримувати стан кожного потоку, так і необхідності використовувати складні алгоритми планування пакетів і буферизації на основних маршрутизаторах [15].

Процес і характеристики CSFQ включають:

- граничний маршрутизатор позначає кожен пакет оцінкою швидкості надходження для кожного потоку, тут зберігається статистика кожного потоку;

- основні маршрутизатори використовують приблизну швидкість надходження, наведену на мітках пакетів, і внутрішню міру справедливої частки для обчислення ймовірності відкидання кожного вхідного пакета (кожен прийнятий пакет обробляється та перемаркується новою інформацією про швидкість надходження);

- процедура оцінки значення «справедливої частки» швидко наближається до оптимального значення.

Основні цілі маршрутизаторів у CSFQ:

- дотримання максимальної та мінімальної справедливості для розподілу смуги пропускання;

- уникнення зберігання статистики кожного потоку у високошвидкісних основних маршрутизаторах.

Можна зазначити, що мета – запобігання для основного маршрутизатора підтримувати черги на потік. Таким чином, після того, як пакет прийнято основним маршрутизатором, він приєднується до однієї з невеликої кількості черг, доки не буде остаточно обробленим. Таким чином, єдина дія, яку основний маршрутизатор може виконати для досягнення цілі, це відкинути пакети з жадібних потоків.

4.3 Недоліки перевантаження мереж

У мережах зв'язку, коли рівень вхідного трафіку перевищує пропускну здатність мережі, кажуть, що мережа перебуває в стані перевантаження. Високий коефіцієнт втрат пакетів і наскрізна повторна передача пакетів, погіршення пропускну здатності та марна витрата часу та енергії на відновлення перевантажень вважаються основними проблемами перевантаження, і ці проблеми, як наслідок, впливають на загальну продуктивність мережі. Загалом алгоритми маршрутизації використовуються для контролю рівня перевантаження. Коли ресурси спільно використовуються кількома користувачами, важливо зменшити швидкість

передачі кожного відправника, щоб зменшити перевантаження. Пакети, які надходять у перевантажену мережу, просто відкидаються, і це небажане скидання пакетів впливає на продуктивність мережі.

Ресурси мережі витрачаються як на відкинуті пакети, так і на повторно передані пакети. Таким чином, перевантаження мережі зменшує пропускну здатність мережі.

Якщо умови перевантаження не обробляються належним чином, це призводить до колапсу перевантаження, і в цьому випадку дані не можуть бути доставлені.

4.4 Система нечіткої логіки

Нечіткі системи більше підходять для невизначеного трафіку даних, особливо для таких систем, як комунікаційні мережі, де трафік даних є невизначеним [16].

У 1960-х роках було представлено теорію нечітких множин для проектування та передбачення невизначеності в природній мові. Ця теорія підходить для невизначеності та наближених міркувань у особливих випадках, таких як математичне моделювання, яке складно вивести. Це дозволяє приймати рішення з обчисленими атрибутами, використовуючи будь-яку неповну або невизначену інформацію. Тому теорія нечітких множин була застосована в системі прийняття рішень або для покращення продуктивності, або для вирішення проблеми, до якої звичайна теорія не може успішно підійти.

Типовий нечіткий механізм характеризується механізмом логічного висновку, фазифікатором і дефазифікатором, як показано на рисунку 4.1. Набір різноманітних лінгвістичних правил, що використовуються для відображення нечітких входів і виходів, відомий як база правил для нечіткої моделі. Механізм охоплює базу системних правил, процес фазифікації та дефазифікації.

Три фази нечіткої системи включають:

- фазифікацію – на цій фазі вхідні значення позначаються функціями належності нечітких множин, контролер нечіткої логіки використовує геометричні трикутники як функції приналежності визначення діапазону нечітких вхідних змінних і визначає універсум дискурсу змінної;
- механізм висновку – механізм логічного висновку оцінює вихідні нечіткі набори з вхідних наборів, використовуючи попередньо визначені нечіткі правила, що містяться в базі правил;
- дефазифікація – дефазифікація є завершальною фазою, на якій нечіткий вихід перетворюється на чітке значення.

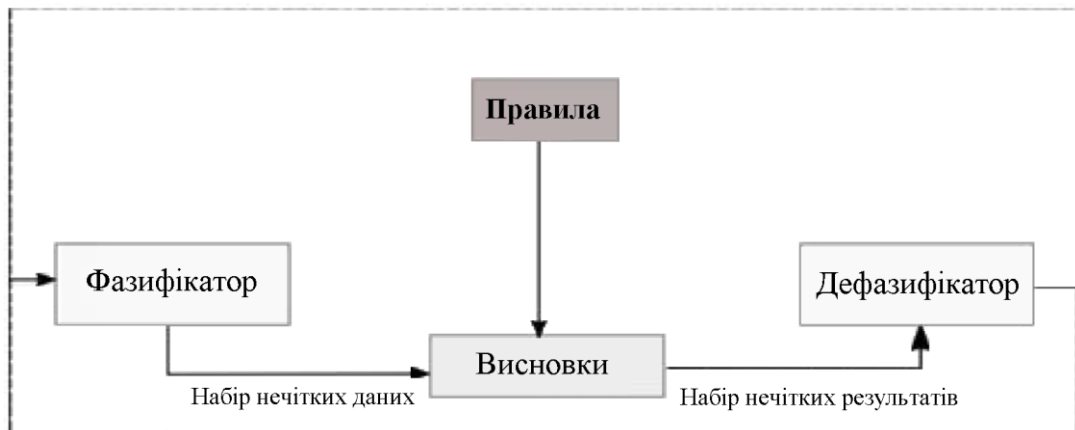


Рисунок 4.1 – Система нечіткої логіки

4.5 Метод справедливої черги з множинної чергою

Метод справедливої черги з кількома чергами (MQFQ) дозволяє потоку використовувати кілька черг [17]. Він використовує кілька хеш-функцій для визначення набору черг FIFO для потоку та обслуговує усі черги у циклічному порядку. Кроки, задіяні в методі Multiple Queue Fair Queuing:

- MQFQ визначає набір черг FIFO для потоку за допомогою кількох хеш-функцій та коли надходить пакет, усі хеш-функції застосовуються до заголовка пакета за допомогою MQFQ для обчислення потенційних черг;

- MQFQ ставить пакет у чергу з найшвидшим обслуговуванням;
- якщо черга, пов'язана з потоком зростає, то потік використовує іншу зі своїх черг і, таким чином, обходить перевантаження;
- оскільки потік може використовувати кілька черг, існує компроміс між ступенем додаткової ємності, переданої неправильно функціонуючим потоком, і кількістю потоків, які не працюють через неправильно функціонуючий потік.

Хоча MQFQ використовує різні черги для розміщення пакетів, переупорядкування пакетів не відбувається, якщо пакети мають однаковий розмір.

Якщо розміри пакетів відрізняються, зміна порядку можлива, але обмежена одним раундом обходу черги. Цього можна уникнути, буферизуючи пакети на один раунд, щоб відновити їх порядок, перед тим, як пересилати їх у канал.

З іншого боку, якщо маршрутизатор фрагментує вхідні пакети на комірочки подібного розміру, MQFQ застосовується до комірок, а потім пакети збираються заново перед пересиланням на канал, тому в цьому випадку переупорядкування не відбувається.

4.6 Критерії максимально-мінімальної справедливості

Відповідно до розмітки планувальника пріоритетів ECSFQ-FBCD, для еластичних потоків і для нееластичних потоків з низьким рівнем перевантаженості для ймовірного відкидання пакетів застосовується максимально-мінімальний критерій справедливості.

Для обчислення справедливості прийнято наступні позначення [15]:

- швидкість прибуття потоку R_f ;
- справедлива частка швидкості в момент часу t $\lambda(t)$;
- швидкість вихідного каналу маршрутизатора S_i ;
- загальна швидкість прибуття $TAR(t)$.

4.6.1 Швидкість прибуття потоку

R_{fi} оцінюється на периферійних маршрутизаторах і вставляється в мітки пакетів. На кожному граничному маршрутизаторі експоненціальне усереднення використовується для оцінки швидкості потоку. Нехай A_{fi}^n та L_i^n – час надходження та довжина n -го пакету потоку i . Розрахункова швидкість потоку i , R_{fi} оновлюється, коли отримується новий пакет, як:

$$R_{fi}^{new} = \left(1 - e^{-T_i^n/K}\right) \frac{L_i^n}{T_i^n} + e^{-T_i^n/K_{R_{fi,old}}}, \quad (4.1)$$

де $T_i^n = t_i^n - t_i^{n-1}$, це час між надходженнями між поточним і попереднім пакетом, а K – константа.

4.6.2 Оцінка справедливої швидкості з'єднання

Швидкість, з якою алгоритм приймає пакет, є функцією оцінки справедливої швидкості, яка позначається як $\lambda(t)$. Розглядаючи $F(\lambda(t))$ як швидкість прийняття:

$$F(\lambda(t)) = \sum_{i=1}^n \min(R_{fi}(t), \lambda(t)) - S_1. \quad (4.2)$$

4.6.3 Загальна швидкість прибуття

Для TAR використовується експонентне усереднення з параметром e^{-T/K_a} , де T – це час між прибуттям поточного та попереднього пакета:

$$TAR_{new} = \left(1 - e^{-T/K_a}\right) \frac{L}{T} + e^{-T/K_a} TAR_{old}, \quad (4.3)$$

де TAR_{old} – значення TAR перед оновленням.

З наведених вище обчислень виходять наступні умови:

- якщо канал перевантажений ($TAR(t) > S_1$), $\lambda(t)$ обирається як унікальне рішення для $F(x) = S_1$;

- якщо канал не перевантажений, $TAR(t) < S_1$, $\lambda(t)$ вважається найбільшою швидкістю серед потоків, які проходять через канал, тобто:

$$\lambda(t) = \max_{1 \leq i \leq n} (R_{fi}(t)). \quad (4.4)$$

З рівняння (4.4), якщо швидкість надходження $R_{fi}(t)$ відома, $\lambda(t)$ можна обчислити безпосередньо.

4.6.4 Обчислення справедливості

Справедливий розподіл пропускної спроможності по максимуму та мінімуму характеризується тим, що всі потоки, які є вузьким місцем маршрутизатора, мають однакову вихідну швидкість. Ця швидкість називається справедливою швидкістю розподілу каналу.

Якщо досягається розподіл пропускної спроможності по максимуму і мінімуму, кожен потік i отримує послугу зі швидкістю, заданою $\min(R_{fi}(t), \lambda(t))$, де $R_{fi}(t)$ та $\lambda(t)$ обчислюються з рівнянь (4.1) і (4.4) відповідно. Загальна швидкість прибуття момент часу t визначається:

$$TAR(t) = \sum_{i=1}^n R_{fi}(t). \quad (4.5)$$

Якщо $TAR(t) > S$, тоді справедлива частка $\lambda(t)$ є унікальним рішенням:

$$S_1 = \sum_{i=1}^n \min(R_{fi}(t), \lambda(t)). \quad (4.6)$$

Якщо $TAR(t) \leq S_1$, тоді жоден біт не відкидається і, таким чином, $\lambda(t) = R_{fi}(t)$.

Якщо $R_{fi}(t) \leq \lambda(t)$, тобто, потік і відправляє не більше ніж справедлива частка швидкості з'єднання, весь його трафік буде перенаправлений.

Якщо $R_{fi}(t) > \lambda(t)$, тоді дріб $\frac{R_{fi}(t) - \lambda(t)}{R_{fi}(t)}$, тоді частка його бітів буде відкинута, тому потік і матиме вихідну швидкість рівну $\lambda(t)$. Це передбачає дуже простий ймовірнісний алгоритм пересилання, при якому кожен вхідний біт потоку «і» відкидається з ймовірністю:

$$\text{Max} \left(0, 1 - \frac{\lambda(t)}{R_{fi}(t)} \right). \quad (4.7)$$

При використанні цих ймовірностей відкидання швидкість прибуття потоку і на наступний вузол визначається, як $\min[R_{fi}(t), \lambda(t)]$.

4.7 Розробка алгоритму ECSFQ-FBCD

На рисунку 4.2 показано загальний алгоритм ECSFQ-FBCD. В алгоритмі ECSFQ-FBCD, коли пакети надходять у периферійний маршрутизатор, до потоку застосовується планувальник пріоритетів. Якщо потік є VoIP або відео, він позначається як нееластичний або інакше, як еластичний потік.

У базовому маршрутизаторі еластичні потоки обробляються за принципом max-min CSFQ. У разі нееластичних потоків ступінь перевантаження визначається за допомогою системи нечіткої логіки. Якщо ступінь перевантаженості середня або висока, потоки обробляються за допомогою справедливої черги з кількома чергами, а якщо ступінь перевантаженості низька, то разом із еластичними потоками застосовується принцип максимального та мінімального рівня CSFQ.

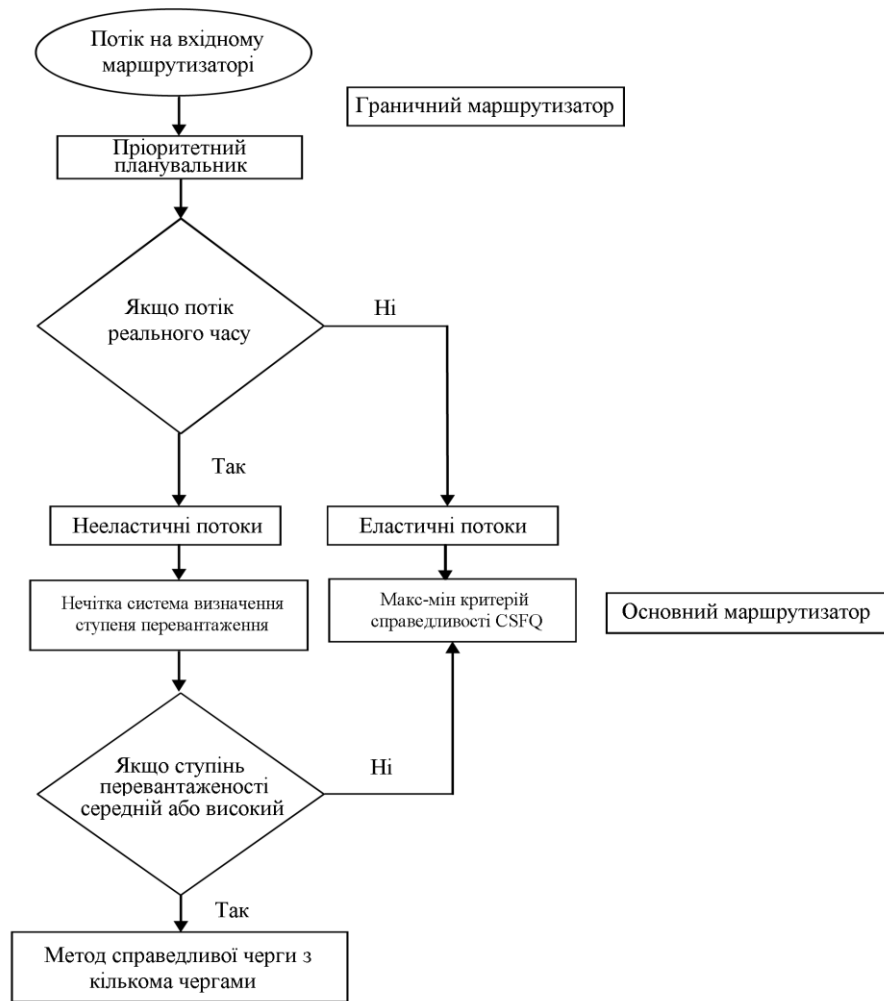


Рисунок 4.2 – Алгоритм застосування справедливої черги з нечіткою логікою виявлення перевантажень (ECSFQ-FBCD)

4.8 Розширена основна модель черги (ECSFQ)

4.8.1 Оцінка затримки та втрат

У цьому процесі оцінки шлях потоку в реальному часі був перевірений заголовком вибраного пакета. Оскільки користувач не часто змінюється всередині мережного домену, зондовий і користувальницький трафік з високою ймовірністю проходять по тому самому шляху. Таким чином, виконується груба оцінка значення затримки, яку випробовують вибрані потоки в мережному домені.

У разі перевірки затримки вхідні маршрутизатори кодують поточну тимчасову мітку T_c у корисне навантаження, а заголовок позначається новим значенням протоколу.

Ці пакети розпізнаються вихідним маршрутизатором та видаляються з мережі. Перед цим вихідний маршрутизатор обчислює затримку з'єднання від краю до краю пакета. Затримка з'єднання – це час, необхідний пакету для прибуття від одного вузла до наступного вузла, інакше кажучи, результуюча різниця між власним часом пакетів та T_c . Вихід класифікує пакети зонда як такі, що належать потоку i та оновлює середню затримку пакета, PD_{avi} для вибірки затримки $D_i(t)$ в момент часу t , використовуючи експоненційно зважене ковзне середнє (EWMA).

$$PD_{avi}(t) = \mu \cdot PD_{avi}(t-1) + (1-\mu) \cdot D_{i(t)}, \quad (4.8)$$

де μ це невелика частка $0 \leq \mu \leq 1$.

Перевірка від краю до краю досліджує втрату пакетів та причини, які спричинили втрати в мережевому домені. де p_j - Це невелика частка $0 < p_j < 1$.

Пакети перевірки «спина до спини» для невеликого інтервалу вибірки T секунд використовуються для визначення втрати з'єднання. Це робиться шляхом обчислення коефіцієнта втрати пакетів з набору пакетів перевірки в пункті призначення.

У цьому методі джерело пересилає серію пакетів перевірки шляхом P_1 до пункту призначення без затримки під час передачі послідовних пакетів. Коефіцієнт втрат (L_i) у вузлі N_i шляхом P_1 в інтервалі T можна розрахувати як:

$$L_i^T = P_{Lo} D_{avi} / R_a, \quad (4.9)$$

де P_{Lo} – це кількість втрачених пакетів, а R_a – розрахункова швидкість надходження пакету.

Тоді загальний коефіцієнт втрат у пункті призначення можна розрахувати як:

$$L^T = \sum L_i^T. \quad (4.10)$$

Тепер фактичні потоки трафіку передаються протягом того самого інтервалу вибірки T секунд через вхідний маршрутизатор, який позначає швидкість надходження потоку, як мітку відповідно до CSFQ. Фактичний коефіцієнт втрат (L_{act}^T) у кожному вузлі вздовж P_1 на інтервалі T можна оцінити подібно до рівняння (4.9). Тоді загальний фактичний коефіцієнт втрат L_{act} у пункті призначення можна розрахувати як:

$$L_{act}^T = \sum L_{act}^T. \quad (4.11)$$

На вихідному маршрутизаторі різницю в коефіцієнтах втрат можна оцінити як:

$$D = L_{act}^T - L^T. \quad (4.12)$$

4.8.2 Класифікація потоку за вхідними вузлами

Якщо значення D (згідно з рівнянням (4.12)) перевищує порогове значення T_1 і якщо затримка (згідно з рівнянням (4.8)) перевищує порогове значення T_2 , то потоки позначаються, як нееластичі потоки (потоки в реальному часі) за допомогою вузла входу, інакше вони розглядаються як еластичні потоки (потоки не реального часу).

Знаходяться канали з високими втратами та вихідний маршрутизатор, через який виходять потоки. Потоки, що потребують пропускну здатність, ізолюються, і швидкість цих потоків перенаправляються на вхідні

маршрутизатори, через які потоки входять в домен. Швидкість, з якою потік входить і виходить з домену мережі, порівнюється вхідним маршрутизатором. З міток CSFQ визначається ідентифікатор вхідного маршрутизатора. Цей ідентифікатор використовується для зв'язку між потоком і його точкою входу, інакше вихідний маршрутизатор не знає, через який вхідний маршрутизатор потік надходить у домен.

Швидкість надходження потоку та відповідні ідентифікатори джерела збираються з міток пакетів, які позначені вхідним вузлом.

4.8.3 Пріоритетний планувальник

Планувальник пріоритетів застосовується до визначених вище категорій потоків згідно з такою умовою: якщо потоком є VoIP або відео, тоді потоки позначаються як вищий пріоритет у низці потоків і розглядаються як нееластичні потоки. Інакше, якщо потоком є звичайний трафік, тоді потік позначається, як нижчий пріоритет у низці потоків та розглядається як еластичний потік.

Ці значення пріоритету позначаються вздовж кожного потоку та передаються на основний маршрутизатор. Основний маршрутизатор перевіряє позначені значення, і відповідно до позначеного значення еластичні потоки обробляються з максимально-мінімальним критерієм справедливості CSFQ. Оскільки нееластичні потоки мають вибухову природу, щоб перевірити рівень перевантаження нееластичні потоків, застосовано метод виявлення заторів на основі нечіткості.

4.9 Виявлення перевантаження на основі нечіткої моделі (FBFD)

Необхідно виявити перевантаження потоків із вищим пріоритетом, щоб запобігти небажаним колізіям пакетів. Нееластичні потоки проходять через систему нечіткої логіки виявлення рівня перевантаження. Швидкість

надходження пакетів та затримка в черзі беруться як вхідні змінні, а ступінь перевантаження як вихідна змінна. Вхідні та вихідні змінні піддаються нечіткої деталізації шляхом подання їх у вигляді функції приналежності, а діапазон перевантаження обчислюється за допомогою процесу дефазифікації. Коли діапазон перевантаження низький, потоки плануються за допомогою принципу максимуму-мінімуму схеми CSFQ. Навпаки, потоки із середнім та високим діапазоном перевантаження плануються з використанням методу справедливої черги з кількома чергами.

4.9.1 Швидкість надходження пакетів

У період вибірки I_s нехай p буде кількістю пакетів, наявних у буфері. Тоді середньозважена швидкість надходження пакетів у момент часу t може бути розрахована як [17]:

$$\text{avgPA}_r(t) = W \cdot \text{avgPA}_r(t - I_s) + (1 - W) \cdot p, \quad (4.13)$$

де W – ваговий коефіцієнт, $0 \leq W \leq 1$. У зваженій швидкості надходження пакетів нехай PA_u буде верхньою межею значення, тоді швидкість надходження пакетів PA_r задається як:

$$PA_r = \begin{cases} \text{avgPA}_r(t) / PA_u & \text{avgPA}_r(t) < PA_u \\ 1.0 & \text{avgPA}_r(t) \geq PA_u \end{cases}. \quad (4.14)$$

4.9.2 Затримка в черзі

Одностороння затримка пакета з індексом «n» оцінюється як різниця між переданим і отриманим часом (Jammeh et al (2007)). Це дається як:

$$D_{\text{own}} = P_r - P_t, \quad (4.15)$$

де P_r – час отримання пакету, а P_t – час передавання пакету.

Затримку на шляху можна обчислити з вимірної затримки та мінімальної затримки, яка задана як:

$$DQ_n = D_{OWn} - D_{OWmin} \cdot \quad (4.16)$$

Затримка в черзі вимірюється як відношення середньої затримки в черзі до максимальної затримки в черзі:

$$DQ = \frac{\text{avg}DQ_n}{\text{max}DQ} \cdot \quad (4.17)$$

Середня затримка в черзі ($\text{avg}DQ_n$) n -го пакету вимірюється за формулою:

$$\text{avg}DQ_n = (1 - \mu) \cdot \text{avg}DQ_{n-1} + \mu \cdot DQ_n, \quad (4.18)$$

де, μ є константою ($\mu \leq 1$).

Враховуючи ці дві змінні швидкість надходження пакетів (PA_r) і затримку в черзі (QD), результат нечіткої системи, буде розглядатися як ступінь перевантаження. Загальна схема вхідних і вихідних змінних нечіткої системи наведена на рисунку 4.3.

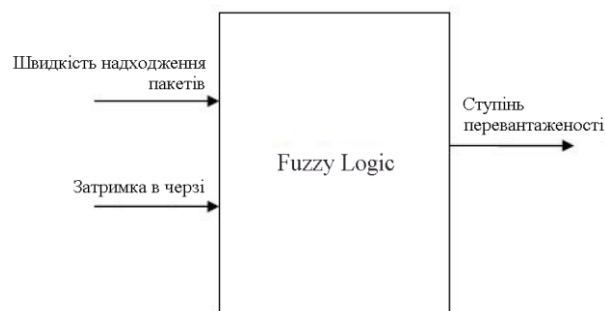


Рисунок 4.3 – Нечітка система виявлення перевантажень

4.9.3 Фазифікація

На цьому етапі вхідні змінні Queuing Delay і Packet Arrival Rate фазифікуються, а потім символізуються в термінах функцій належності.

Для представлення вхідних і вихідних змінних використовується геометричний рисунок трикутників. Функція належності вхідних даних, а саме швидкості надходження пакетів і затримки в черзі, наведена на рисунках 4,4 і 4,5 відповідно. Функція приналежності для виведення наведена на рисунку 4.6.

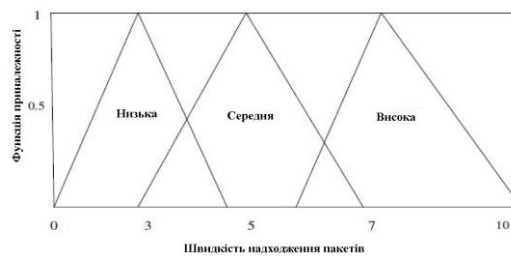


Рисунок 4.4 – Функція приналежності для швидкості надходження пакетів



Рисунок 4.5 – Функція приналежності для затримки в черзі



Рисунок 4.6 – Функція приналежності для ступеня перевантаженості

4.9.4 Нечіткі набори правил

Вхідні та вихідні змінні розподіляються на три діапазони: низький, середній і високий. На основі цих вхідних діапазонів будуються нечіткі правила (таблиця 4.1).

Таблиця 4.1 – Набір нечітких правил

Швидкість надходження пакетів (PA_r)	Затримка в черзі (QD)	Ступінь перевантаження (D)
Низька	Низька	Низька
Низька	Середня	Середня
Низька	Висока	Середня
Середня	Низька	Середня
Середня	Середня	Середня
Середня	Висока	Висока
Висока	Низька	Середня
Висока	Середня	Висока
Висока	Висока	Висока

4.9.5 Дефазифікація

Нарешті вихідне значення генерується з використанням процесу дефазифікації.

Метод стратегії центроїду області використовується для дефазифікації вхідних змінних і визначається як:

$$F_c = \frac{\sum_{R_1}^{R_n} N_i \cdot M(N_i)}{\sum_{R_1}^{R_n} M(N_i)}. \quad (4.19)$$

У наведеному вище рівнянні F_c представляє нечітку вартість, N_i є вузловим елементом, граничне значення від R_1 до R_n позначає набір нечітких правил, а $M(N_i)$ означає функції приналежності.

Як результат нечіткої системи визначається область перевантаження нееластичних потоків. На основі оціненого значення діапазону перевантаження потоки плануються за двома методами черги.

4.10 Імітаційне моделювання

Побудова мереж зв'язку в реальному часі для цілей тестування з прийнятною кількістю мережевих компонентів може вимагати великих витрат і тривалого часу. Емулятори та симулятори мереж неминучі в галузі телекомунікацій, оскільки вони дають можливість створювати, модифікувати та тестувати змодельовані мережі, виправдовувати вартість і доводити їх здатність виживати в реальному середовищі. Для моделювання комп'ютерних мереж, зокрема протоколу керування передачею (TCP), протоколів маршрутизації та багатоадресної передачі через провідні та безпроводні мережі розроблено мережевий симулятор Network Simulator 2, який є вільно розповсюджуваним відкритим вихідним кодом.

NS-2 – це симулятор мережі з дискретними подіями, який підтримує різні типи протоколів. NS-2 складається з імітаційної основи та набору компонентів моделювання, які можна гнучко налаштувати, створюючи виконувану систему моделювання. Структура моделювання структурована в планувальник NS, набір настроюваних вузлів NS і бібліотеку попередньо визначених компонентів. Для цілей аналізу NS-2 створює файл трасування, що містить дані про окремі пакети, такі як вузол джерела, час передачі, розмір пакета, вузол призначення та час отримання.

Для гнучкості складні об'єкти було розкладено на простіші. Інтерфейс конфігурації є об'єктно-орієнтованою версією Tcl (OTcl). Код інтерфейсу для інтерпретатора OTcl відокремлений від основного симулятора.

4.11 Результати моделювання

4.11.1 Імітаційна модель і параметри

У цьому розділі розглядається ефективність розширеного ядра без збереження стану чесної черги з нечітким виявленням перевантаження (ECSFQ-FBCD) із широким дослідженням моделювання на основі симулятора мережі NS-2. Результати порівнюються з традиційною моделлю CSFQ.

Топологія, використана в симуляції, показана на рисунку 4.7. Використовується суміш потоків VoIP і TCP. Розмір пакета становить 512 байт, а пропускна здатність і затримка з'єднання встановлені як 10 Мб і 10 мс відповідно.

Пропускна здатність вузького місця для каналів (0,1), (0,2) і (0,3) спочатку встановлена як 5 Мб. Для цілей моделювання враховуються значення $\mu=0,7$ (відповідно до рівняння 4.8), $W=0,9$ (відповідно до рівняння 4.13) і $\mu=0,1$ (відповідно до рівняння 4.18).

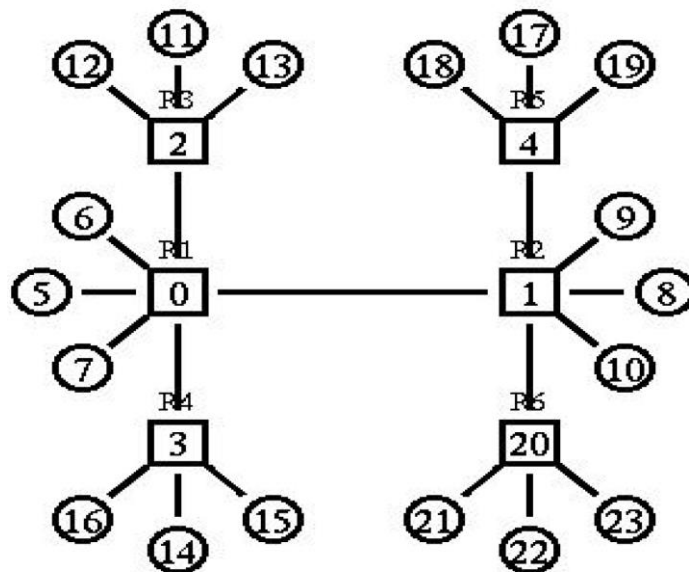


Рисунок 4.7 – Топологія мережі

4.11.2 Показники продуктивності

В експериментах з моделюванням пропускна здатність вузького місця та швидкість трафіку змінюються для вимірювання продуктивності мережі. Під час експерименту потоки TCP використовуються для трафіку, що відповідає еластичності, а потоки VoIP або Video – для трафіку, що не відповідає.

За допомогою моделювання аналізуються такі показники продуктивності:

- пропускна здатність;
- затримка;
- втрата пакетів.

4.11.3 Вплив різної пропускної здатності вузького місця

У першому експерименті пропускна здатність вузького місця змінюється для каналів (0,1), (0,2) і (0,3) як 2 МБ, 4 МБ, 6 МБ і 8 МБ, щоб обчислити втрату пакетів, затримку та пропускна здатність.

У першому випадку використовується набір потоків VoIP і TCP. Оскільки модель масового обслуговування з вдосконаленою справедливою чергою з нечіткою логікою виявлення перевантажень (ECSFQ-FBCD) розглядає еластичні та нееластичні потоки окремо, характеристики як еластичних, так і нееластичних потоків були значно покращені.

На рисунку 4.8 показано, що при збільшенні швидкості вузького місця пропускна здатність TCP була краща на 68 відсотків в ECSFQ-FBCD порівняно з CSFQ.

На рисунку 4.9 показано, що ECSFQ-FBCD має меншу затримку, ніж CSFQ, якщо збільшити пропускну здатність вузького місця. Можна відзначити, що відсоток покращення метрики затримки коливається від 83 до 98 відсотків у ECSFQ-FBCD порівняно з CSFQ.

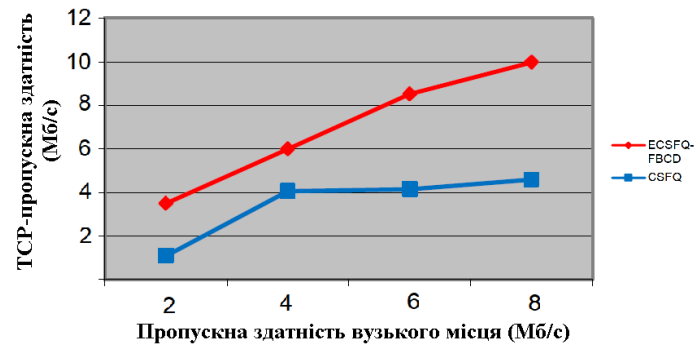


Рисунок 4.8 – Залежність ТСП-пропускної здатності від пропускної здатності вузького місця

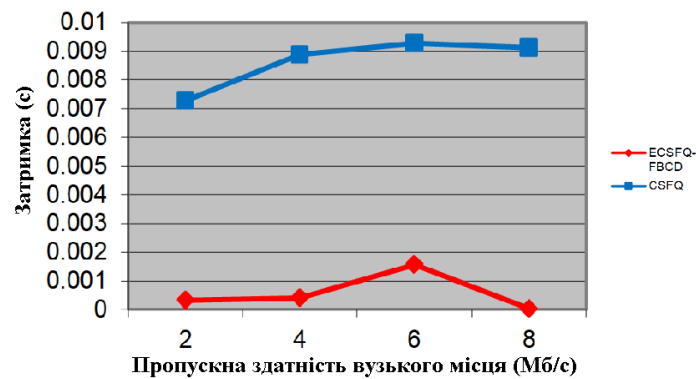


Рисунок 4.9 – Залежність затримки від пропускної здатності вузького місця

Рисунок 4.10 показує, що ECSFQ-FBCD покращується на 20–80 відсотків порівняно з CSFQ щодо показника втрати пакетів.

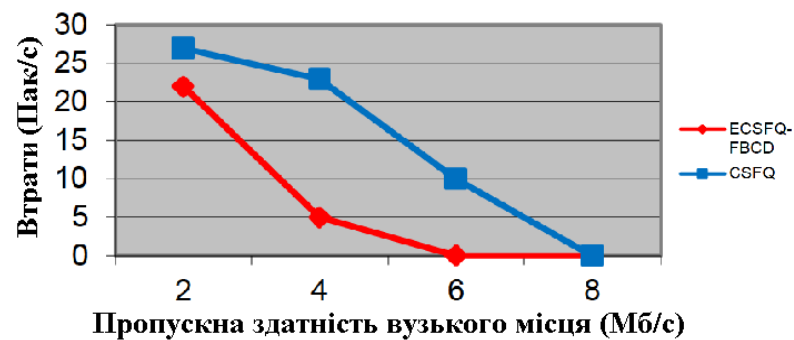


Рисунок 4.10 – Залежність кількості втрачених пакетів від пропускної здатності вузького місця

Таким чином, показано покращену модель базової чесної черги без збереження стану із нечіткою технікою виявлення перевантажень (ECSFQ FBCD). Замість того, щоб застосовувати однакову обробку для всіх потоків, у цій моделі потоки розрізняються на еластичні і нееластичні потоки. Рівень перевантаження нееластичних потоків визначається за допомогою нечіткої моделі, а якщо рівень перевантаження є середнім і високим, потоки обробляються за допомогою MQFQ. Якщо рівень перевантаження низький, тоді вони обробляються за принципом max-min CSFQ разом із еластичними потоками. Дослідження моделювання показує, що модель черги ECSFQ-FBCD підвищує пропускну здатність, зменшує затримку та втрату пакетів для різних швидкостей і пропускну здатності вузького місця, і, у свою чергу, покращується загальна продуктивність мережі.

5 СПРАВЕДЛИВА ЧЕРГА З НИЗЬКОЮ ЗАТРИМКОЮ ТА ДИФЕРЕНЦІЙНИМ ВІДКИДАННЯМ ПАКЕТІВ

Зважена справедлива черга (WFQ) – найпримітивніший алгоритм справедливого планування. WFQ пропонує справедливу чергу, яка ділить доступну смугу пропускання між чергами трафіку на основі їх ваг. За останнє десятиліття було реалізовано кілька підходів, що ґрунтуються на WFQ, але ці підходи не гарантують кращої продуктивності, коли трафік містить суміш як еластичного, так і нееластичного трафіку.

Щоб подолати цю складність, представимо справедливу чергу з низькою затримкою та диференціальним відкиданням (Low Latency Weighted Fair Queuing for inelastic flows with Differential Dropping – LLWFQD) для еластичних потоків. Планувальник WFQ призначає початковий і кінцевий теги кожному з пакетів, що надходять, і обслуговує пакети в порядку зростання їх кінцевих тегів. У моделі черг LLWFQD фінішний тег оновлюється параметром затримки і виходить новий вираз для фінішного тега. У разі еластичних потоків реалізується диференціальний підхід до відкидання пакетів, у якому пакети з різних потоків відкидаються для досягнення справедливості.

5.1 Огляд зваженої справедливої черги

Зважена справедлива черга (WFQ) пропонує справедливу чергу, яка ділить доступну смугу пропускання між чергами трафіку на основі ваги. Кожен потік або сукупність потоків пов'язані з незалежною чергою, якою призначена вага, щоб гарантувати, що важливий трафік отримує більш високий пріоритет у порівнянні з менш важливим трафіком. Під час перевантаження трафік у кожній черзі (окремий потік або сукупність потоків) захищається та обробляється справедливо відповідно до його ваги.

WFQ забезпечує управління пріоритетами трафіку, що автоматично сортує окремі потоки трафіку без визначення списків доступу. Існує дві категорії сеансів WFQ: з високою пропускнуою здатністю та з низькою пропускнуою здатністю.

Трафік з низькою пропускнуою здатністю має ефективний пріоритет над трафіком з високою пропускнуою здатністю, а трафік з високою пропускнуою здатністю ділить службу передачі пропорційно відповідно до призначених ваг. Планувальник WFQ призначає початковий і кінцевий теги кожному пакету, що прибуває, і обслуговує пакети в порядку зростання їх кінцевих тегів. Цей планувальник завершує обслуговування всього пакету з сеансу, перш ніж він перейде до наступного сеансу.

Пакетний GPS (Generalized Processor Sharing) зазвичай відомий як зважена справедлива організація черг (WFQ). WFQ і GPS пов'язані у тому сенсі, що з погляду затримки пакет завершить обслуговування у системі WFQ пізніше, ніж у відповідній системі GPS, лише на час передачі одного пакета максимального розміру.

5.1.1 Проблеми зваженої справедливої черги

WFQ в основному не доступні у всіх типах комутаторів та маршрутизаторів, оскільки WFQ має певну складність та потребує значний обсяг ресурсів при масштабуванні. Пакети різних потоків повинні відкидатися по-різному, щоб досягти мети справедливості, що не досягається за допомогою WFQ.

5.1.2 Черга з низькою затримкою

LLQ є комбінацією пріоритетної черги (PQ) і WFQ. Як правило, у цій моделі є одна пріоритетна черга та кілька зважених справедливих черг. LLQ є спробою обслуговування трафіку в реальному часі ефективним за часом

способом. LLQ дозволяє обирати з черги та відправляти насамперед чутливі до затримки дані (нееластичні потоки). Така практика забезпечує пріоритетну обробку чутливих до затримки даних, порівняно з іншим трафіком.

5.2 Модель

Вибір відповідного алгоритму планування пакетів може суттєво вплинути на якість обслуговування, що спостерігається при проходженні пакетів по мережі, та загальну продуктивність мережі, оскільки різні алгоритми націлені на оптимізацію різних показників пакетів QoS, таких як затримка, джиттер затримки, пропускна здатність і справедливість. Підтримка QoS дуже важлива у мережному середовищі.

Підхід LLWFQD забезпечує підвищення продуктивності мережі з погляду високої пропускної спроможності, меншої затримки та втрати пакетів.

У моделі LLWFQD кінцевий тег оновлюється новим виразом з погляду затримки, а пакети, що надходять, обслуговуються в порядку зростання кінцевого тега. Потім реалізується швидкість відкидання, щоб відкидати відповідні пакети у разі еластичних потоків. Оскільки пакети з меншою затримкою плануються першими, цю модель можна назвати WFQ з низькою затримкою.

5.2.1 Операції WFQ

Планувальник WFQ призначає початковий та кінцевий теги кожному пакету в міру його надходження до черги. Пакет із найменшим кінцевим тегом вибирається першим для виведення. Початковий час розраховується за рівнянням:

$$S(P_i^k) = \max(V(A(P_i^k)), F(P_i^{k-1})). \quad (5.1)$$

Кінцевий тег визначається рівнянням:

$$F(P_i^k) = S(P_i^k) + \frac{L_i^k}{r_i}, \quad (5.2)$$

де P_i^k позначає k-й пакет потоку i , L_i^k є розміром пакету P_i^k , $F(P_i^k)$ позначає кінцевий тег, $S(P_i^k)$ позначає початковий тег, r_i є вагою та $V(t)$ є віртуальним часом, визначеним як:

$$\frac{dV(t)}{dt} = \frac{C}{\sum_{i \in B(t)} r_i}, \quad (5.3)$$

де $B(t)$ – множина активних потоків у момент часу t .

5.2.2 Планувальник LLWFQ

У вдосконаленому планувальнику WFQ кінцевий тег обчислюється за рівнянням:

$$F(P_i^k) = S(P_i^k) + \frac{L_i^k \cdot D_i^T}{r_i}, \quad (5.4)$$

де D_i^T розраховується за допомогою наступного рівняння:

$$D_i^T = \frac{\sum_{j=p}^q d_i^j}{q-p}, \quad (5.5)$$

де d_i^j – час затримки пакета p у класі i , $(q-p)$ – кількість вхідних пакетів

за вказаний інтервал часу. З рівняння 5.4 можна помітити, що чим менша затримка для потоку i , тим меншим буде значення кінцевого тега $F(P_i^k)$. Таким чином, потоки з найменшою затримкою будуть заплановані першими. Тому цю покращену модель WFQ можна назвати WFQ із низькою затримкою (LLWFQ).

На наступному рисунку 5.1 наведено графічне представлення методу LLWFQ.

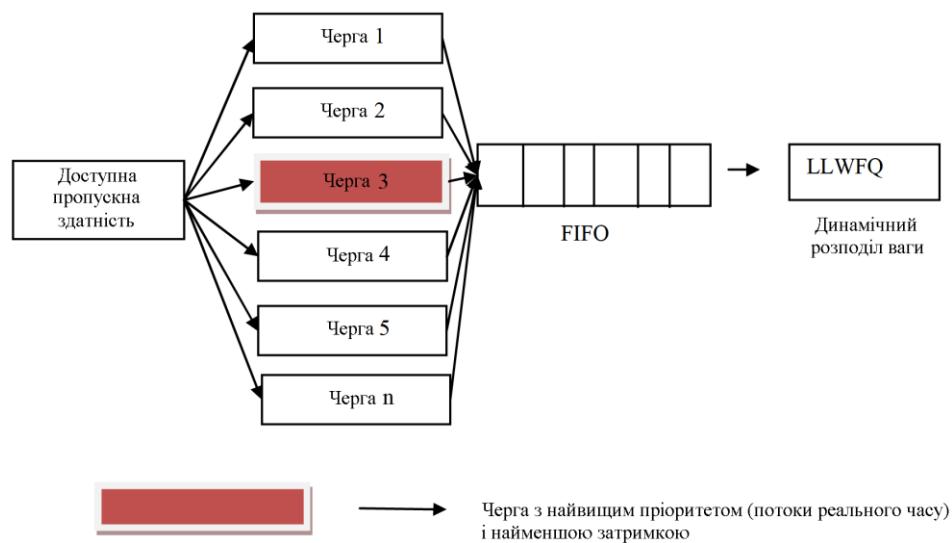


Рисунок 5.1 – Графічне представлення моделі LLWFQ

5.3 Диференційна обробка відкидання

Під час перевантаження відкидання пакетів відіграє важливу роль у підвищенні продуктивності мережі. Крім того, схеми відкидання пакетів набагато дешевші у реалізації, і для цього достатньо однієї черги FIFO. У моделі LLWFQD пакети еластичних потоків відкидаються задля досягнення справедливості. Процес LLWFQD показаний рисунку 5.2.

На рисунку 5.2:

- черга (а) містить усі типи пакетів, які мають різне значення справедливої частки потоку;

- після застосування методу диференціальної відкидання пакетів пакети, які перевищують значення справедливої частки потоку, тобто пакети червоного кольору, відкидаються;

- черга (б) містить лише пакети з меншим значенням справедливої частки потоку.

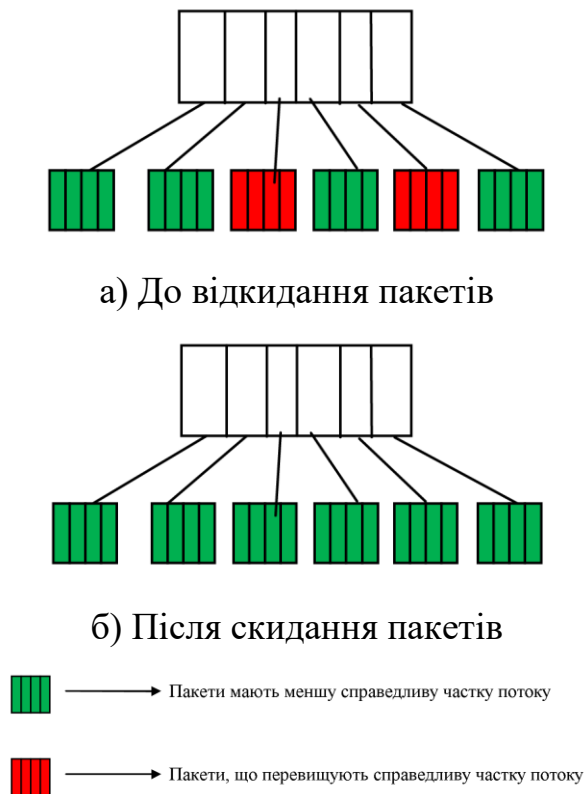


Рисунок 5.2 – Схема диференціального відкидання пакетів

Після того, як усі потоки охарактеризовано, пакети, які перевищують справедливу вартість потоку, повинні бути відкинуті. Тут розглядається вузьке місце вхідного маршрутизатора з одним довготривалим потоком TCP (еластичний потік).

Вважаємо розмір вхідної черги малим, тому середня швидкість відкидання пакетів у цьому TCP-потокі приблизно визначається як $1/(0,375 \cdot p^2)$, де p – добуток пропускної здатності (R) на затримку (RTT) у пакетах. Коли швидкість відкидання пакетів зменшується до $1/(0,66 \cdot p^2)$, тоді

еластичний потік досягає повної смуги пропускання R . Цей результат збігається з добре відомим співвідношенням між розміром вікна (w) і швидкістю відкидання (q), яке визначається як:

$$W = \sqrt{\frac{2}{3q}}.$$

Якщо вказано конкретне значення RTT , то швидкість скидання пакетів для справедливого потоку пропускну здатності визначається як:

$$\frac{1}{\left(\frac{2}{3} \left(\frac{m \cdot RTT}{l_p}\right)^2\right)}, \quad (5.6)$$

де m – це розподіл потоків у WFQ.

Отже, швидкість відкидання пакетів задана як:

$$d_i = \frac{1}{0,66 \cdot \left(\frac{m \cdot RTT}{l_p}\right)^2}, \text{ if } m < m_i, \quad (5.7)$$

де l_p – середній розмір пакету TSP-поток, m_i – спостережувана фактична кількість байтів за часовий інтервал вимірювання t_i .

Для потоку ваги w_i пов'язані з ним байти, надіслані протягом інтервалу t_i часу вимірювання, будуть $m \cdot w_i$.

Припустимо, якщо спостережуваний надісланий байт замінено, фактична швидкість випадання буде задана як[^]

$$d'_i = \frac{d_i}{(w_i)^2}, \quad (5.8)$$

5.4 Результати моделювання

У цьому розділі вимірювання продуктивності техніки LLWFQD розглядаються за допомогою моделювання на основі симулятора мережі NS-2. Результати порівнюються з методикою WFQ. Для експериментальних цілей використовується суміш потоків VoIP та TCP. Розмір пакета встановлено на 512 байт, і всього є 10 потоків. Топологія, використана в моделюванні, показана на рисунку 5.3.

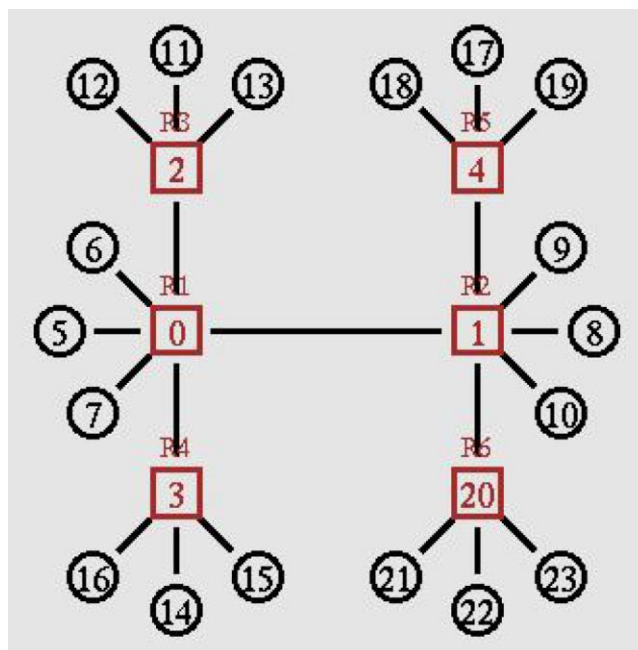


Рисунок 5.3 – Топологія мережі

Продуктивність мережі визначається за допомогою моделювання шляхом зміни пропускної здатності вузького місця.

Удосконалення WFQ із скиданням пакетів покращило продуктивність мережі з точки зору пропускної здатності, втрати пакетів і затримки для різної пропускної здатності вузького місця.

З рисунка 5.4 видно, що при порівнянні LLWFQD і WFQ спостерігається постійне збільшення пропускної здатності TCP на 35 відсотків для різних значеннях пропускання спроможності.

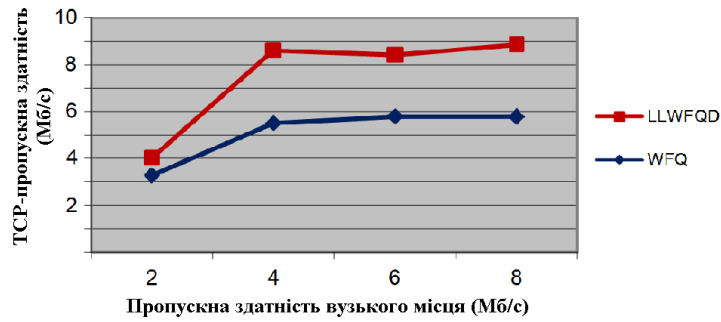


Рисунок 5.4 – Залежність ТСП-пропускної здатності від пропускної здатності вузького місця



Рисунок 5.5 – Залежність затримки передачі від пропускної здатності вузького місця

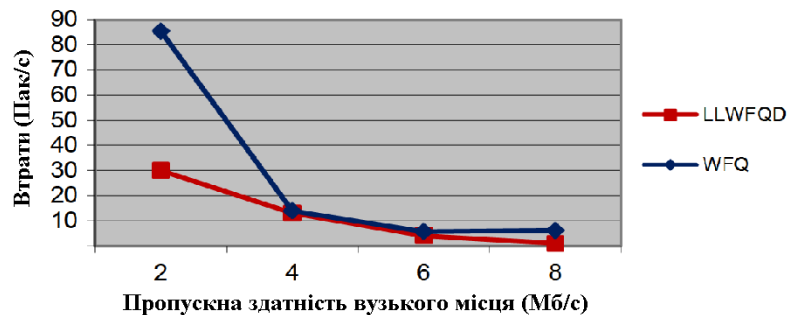


Рисунок 5.5 – Залежність кількості втрачених пакетів від пропускної здатності вузького місця

Таким чином, можна побачити, що зважена справедлива черга з низькою затримкою з відкиданням пакетів (LLWFQD) може бути ефективною моделлю черги для мережевого середовища з трафіком

реального часу. У цій моделі нееластичні потоки з меншою затримкою обробляються за допомогою вдосконаленого планувальника WFQ, а еластичні потоки обробляються відкиданням пакетів, що насправді є дуже ефективним і простим рішенням. З використанням симулятора NS-2 показано, що LLWFQD показує кращу продуктивність щодо пропускної здатності, затримки та втрат порівняно з WFQ.

ВИСНОВКИ

Методи керування чергами та алгоритми контролю перевантаження вкрай необхідні для підтримки стабільності комп'ютерних мереж. Еластичні та нееластичні потоки вимагають різних типів обробки для запобігання пригнічення останніх в чергах маршрутизаторів.

Кваліфікаційна магістерська робота в основному зосереджена на покращенні продуктивності мереж шляхом впровадження розширених моделей масового обслуговування.

Для підвищення справедливості та ефективності мережі представлено модель покращеної справедливої черги з нечіткою логікою виявлення перевантажень, більш ефективну ніж модель справедливої черги CSFQ.

Еластичні потоки розглядаються за принципом max-min CSFQ. Крім того, для нееластичних потоків застосовується нечітке виявлення перевантаження, і відповідно до рівня перевантаження потоки плануються за допомогою справедливої черги з кількома чергами MQFQ або за max-min принципом CSFQ.

Представлено модель справедливої черги з низькою затримкою та диференціальним відкиданням пакетів. У випадку еластичних потоків реалізується диференціальний підхід до відкидання пакетів, коли пакети з різних потоків відкидаються для досягнення справедливості розподілу пропускної здатності.

Проведене імітаційне моделювання підтвердило вірність теоретичних викладок, представлених в кваліфікаційній роботі.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Tucker, Rodney S. The role of optics and electronics in high-capacity routers. *Journal of Lightwave Technology* 24, 12 (2006), 4655–4673.
2. Blumenthal, D.J., Prucnal, P.R., and Sauer, J.R. Photonic packet switches: architectures and experimental implementations. *Proceedings of the IEEE* 82, 11 (nov 1994), 1650–1667.
3. Tucker, Rodney S., and Zhong, Wen De. Photonic packet switching: An overview. *IEICE Trans. Communi.* E82-B, 2 (1999), 254–264.
4. Mughal, S.S., Tucker, R.S., and Hinton, K. A taxonomy of optical buffer architectures. In *Optical Internet and Next Generation Network, 2006. COIN-NGNCON 2006. The Joint International Conference on* (july 2006), pp. 232–234.
5. Zhang, L, Shenker, S, and Clark, David D. Observations on the dynamics of a congestion control algorithm: the effects of two way traffic. In *Proc. of ACM SIGCOMM 91* (Zurich, Switzerland, Sept. 1991), pp. 133–147.
6. Sivaraman, Vijay, Elgindy, Hossam, Moreland, David, and Ostry, Diethelm. Packet pacing in short buffer optical packet switched networks. In *Proc. of IEEE INFOCOM 06* (Spain, Apr. 2006).
7. Willinger, W., Taqqu, M., Sherman, R., and Wilson, D. Self-similarity through highvariability: statistical analysis of ethernet lan traffic at the source level. 100–113.
8. Feldmann, A., Gilbert, A., Huang, P., and Willinger, W. Dynamics of IP traffic: A study of the role of variability and the impact of control. In *Proc. of ACM SIGCOMM '99* (Aug. 1999), pp. 301–313.
9. Allman, M., Paxson, V., and Stevens, W. RFC 3439: TCP Congestion Control.
10. Aggarwal, Amit, Savage, Steve, and Anderson, Thomas. Understanding the performance of TCP pacing. In *Proc. of IEEE INFOCOM 2000* (Tel Aviv, Israel, Mar. 2000), pp. 1157–1165.

11. Enachescu, Mihaela, Ganjali, Yashar, Goel, Ashish, Mckeown, Nick, and Rough- garden, Tim. Routers with very small buffers. In Proc. of INFOCOM 2006 (Barcelona, Spain, Apr. 2006).

12. Baldwin, RO, Nathaniel, J, Midiff, SF & Kobza, JE 2003, 'Queueing network Analysis: Concepts, terminology, and methods', The Journal of Systems and Software, vol. 66, pp. 99-117.

13. Forouzan, BA 2003, TCP/IP Protocol Suite, TATA McGraw-Hill, New York.

14. Georg, M, Jechlitschek, C & Gorinsky, S 2007, 'Improving individual flow performance with Multiple Queue Fair Queuing', In: Proceedings of the Fifteenth IEEE International Workshop on Quality of Service (IWQoS), Illionis, pp. 141-144.

15. Stoica, I, Shenker, S & Zhang, H 2003, 'Core-Stateless Fair Queuing: a scalable architecture to approximate fair bandwidth allocations in high speed networks', IEEE/ACM Transactions on Networking, vol. 11, no. 1, pp. 33-46.

16. Nyirenda, CN & Dawoud, DS 2006, 'Multi-objective particle swarm optimization for Fuzzy Logic based Active Queue Management', In: Proceedings of IEEE International Conference in Fuzzy Systems, Canada, pp. 2231-2238.

17. Nyirenda, CN & Dawoud, DS 2008, 'Self-Organization in a Particle Swarm optimized fuzzy logic congestion detection mechanism for IP networks', Scientia Iranica, vol. 15, no. 6, pp.589-604.

18. Попов В.Д., Вірко А.О., Янковський О.А. Управління трафіком в IP-мережах//Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління. Чотирнадцята міжнародна науково-технічна конференція. Баку – Харків – Жиліна. 2024 р. Том 1. С. 115.

19. Вірко А.О., Пилипенко А.О., Янковський О.А. Методи буферизації пакетів в маршрутизаторах IP-мереж//Збірник тез доповідей дванадцятої міжнародної науково-технічної конференції «Проблеми інформатизації». Баку, Харків, Бельсько-Бяла. 2024. С. 76.