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 оптимізації трафіку комп'ютерних мереж

(тема)

Виконав:
студент II курсу, групи СПм-22-2
Куриленко 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
Системне програмування
(повна назва освітньої програми)

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

Допускається до захисту

Зав. кафедри ЕОМ _____ Коваленко А.А.
(підпис) (прізвище, ініціали)

2023 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 Системне програмування
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
“____” _____ 20___ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студенту Куриленку Андрію Олександровичу
(прізвище, ім’я, по батькові)

1. Тема роботи Методи оптимізації трафіку комп'ютерних мереж

затверджена наказом по університету від “ 6 ” листопада 2023 р. № 1299 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 15 січня 2024р.

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: ОС Windows 10, OpNet 14, NS-3.

4. Перелік питань, що потрібно опрацювати в роботі

1) аналіз сучасного стану проблеми

2) огляд технологій управління перевантаженням та середньою затримкою

3) моделі управління мережним трафіком

4) вибір програмних та апаратних засобів реалізації

5) проведення експериментальних досліджень

6) висновки

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

Слайдів презентації – 17 шт.

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	07.11.23 – 09.11.23	
	вирішення		
2	Огляд технологій управління перевантаженням	10.11.23 – 14.11.23	
3	Розробка моделі управління мережним	15.11.23 – 11.12.23	
	трафіком		
4	Вибір програмних та апаратних засобів	12.12.23 – 20.12.23	
	реалізації		
5	Тестування запропонованого метода	21.12.23 – 25.12.23	
6	Оформлення пояснювальної записки	26.12.23 – 12.01.24	

Дата видачі завдання 7 листопада 2023 р.

Студент _____
(підпис)

Керівник роботи _____ доц. Янковський О.А.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 66 с., 12 рис., 1 дод., 18 джерел.

АДИТИВНЕ ЗБІЛЬШЕННЯ/МУЛЬТИПЛІКАТИВНЕ ЗМЕНШЕННЯ, КАНАЛ ЗВ’ЯЗКУ, ПЕРЕВАНТАЖЕННЯ, ПРОПУСКНА ЗДАТНІСТЬ, СКИДАННЯ ПАКЕТІВ, ТАЙМ-АУТ ПЕРЕДАЧІ, ТОПОЛОГІЯ.

Оптимізація базових параметрів протоколу на прикладному рівні (тобто відкриття кількох паралельних TCP-потоків, налаштування розміру буфера TCP і розміру блоку вводу/виводу) є одним із способів покращити пропускну здатність.

З іншого боку, вузькі місця наскрізної передачі даних у високопродуктивних мережевих системах виникають здебільшого в системах зберігання даних, а не в мережі. Продуктивність системи зберігання значною мірою залежить від швидкості її дискової та центральної підсистем.

Таким чином, надзвичайно важливо оцінити пропускну здатність системи зберігання на обох кінцевих точках на додаток до пропускної здатності мережі.

ABSTRACT

Master’s thesis: 66 pages, 12 figures, 1 appendices, 18 sources.

ADDITIVE INCREASE/MULTIPLICATIVE DECREASE, COMMUNICATION CHANNEL, BANDWIDTH CAPACITY, OVERLOAD, PACKET DROPPING, TRANSMISSION TIMEOUT, TOPOLOGY.

Optimizing basic protocol parameters at the application level (ie, opening multiple parallel TCP streams, adjusting TCP buffer size and I/O block size) is one way to improve throughput.

On the other hand, end-to-end data transfer bottlenecks in high-performance network systems occur mostly in the storage systems, not in the network. The performance of a storage system largely depends on the speed of its disk and core subsystems.

Therefore, it is extremely important to evaluate the storage system bandwidth at both endpoints in addition to the network bandwidth.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ACK – підтвердження нового пакету TCP (англ., New TCP packet acknowledgment)

AIMD – адитивне збільшення/мультиплікативне зменшення (англ., Additive increase multiplicative decrease)

CWA – дія вікна перевантаження (англ., Congestion window action)

Cwnd – вікно перевантаження TCP (англ., TCP congestion window)

DACK – підтвердження повторного TCP-пакету (англ., Duplicate TCP packet acknowledgment)

IP – Інтернет-протокол (англ., Internet protocol)

OSI – взаємозв'язок відкритих систем (англ., Open System Interconnection)

RTO – очікування повторної передачі (англ., Retransmission timeout)

RTT – час подвійного оберту (англ., Round trip time)

Ssthresh – поріг повільного запуску (англ., Slow start threshold)

TCP – протокол керування передачею (англ., Transmission Control Protocol)

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

ВСТУП

Для великомасштабних розподілених додатків ефективне використання доступної пропускної здатності мережі та оптимізація швидкості передачі даних має вирішальне значення для наскрізної продуктивності додатків. Сьогодні багато регіональних і національних оптичних мереж забезпечують високошвидкісне підключення до мережі для своїх користувачів. Однак більшість користувачів не можуть отримати навіть частки теоретичної швидкості, обіцяної цими мережами, через такі проблеми, як неоптимальне налаштування протоколу, вузьке місце доступу до жорсткого диску на сторонах відправлення та/або прийому та обмежень пов’язаних з процесором. Це означає, що наявність високошвидкісних мереж є важливим, але недостатнім для покращення пропускної здатності наскрізної передачі даних. Можливості ефективного використання цих високошвидкісних мереж стає все більше і важливіше.

Оптимізація базових параметрів протоколу на прикладному рівні (тобто відкриття кількох паралельних TCP-потоків, налаштування розміру буфера TCP і розміру блоку вводу/виводу) є одним із способів покращити пропускну здатність мережі. З іншого боку, вузькі місця наскрізної передачі даних у високопродуктивних мережевих системах виникають здебільшого в системах зберігання даних, а не в мережі. Продуктивність системи зберігання значною мірою залежить від швидкості її дискової та центральної підсистем. Таким чином, надзвичайно важливо оцінити пропускну здатність системи зберігання на обох кінцевих точках на додаток до пропускної здатності мережі. Вузьке місце на дисках можна усунути за допомогою кількох дисків (черговість даних), а вузьке місце ЦП можна усунути за допомогою кількох процесорів (паралелізм).

1 МЕТА КВАЛІФІКАЦІЙНОЇ РОБОТИ

Магістерська кваліфікаційна робота передбачає розробку моделі прикладного рівня для прогнозування найкращої комбінації параметрів протоколу для оптимальної продуктивності мережі, включаючи кількість паралельних потоків даних та розмір буфера протоколу в модель продуктивності для прогнозування оптимальної кількості потоків і розміру буфера для найкращої наскрізної пропускної здатності даних.

Виконання кваліфікаційної магістерської роботи передбачає:

- проведення аналіза літературних джерел, пов’язаних з проблемою збільшення мережевої пропускної здатності;
- розробку моделі оптимізації пропускної здатності передачі даних, в якій використовується якомога менше інформації, водночас забезпечуючи точність і масштабованість незалежно від архітектури кінцевих систем;
- проведення імітаційне моделювання для підтвердження теоретичних викладок;
- проведення аналізу отриманих результатів моделювання.

2 ОГЛЯД СУЧАСНОГО СТАНУ ПРОБЛЕМИ

2.1 Вузьке місце

Розвиток науки в багатьох галузях (наприклад, моделювання узбережжя та навколишнього середовища, біоінформатика, медична візуалізація, динаміка рідини та фізика високих енергій) із зростаючими потребами в управлінні даними можливий лише за наявності наскрізних мережевих можливостей, які підтримуватимуть їхню мережу, вимоги до умов і даних. У зв'язку з недавніми розробками технології оптичних мереж, пропускна спроможність яких перевищує 100 Гбіт/с, використання мережевих ресурсів поставило перед існуючими рішеннями проміжного програмного забезпечення завдання забезпечення розподілених петамасштабних наукових досліджень.

Високошвидкісні мережі створили основу для покоління суперкомп’ютерів, і це стало обов’язковим для того, щоб хост-системи, стек протоколів і розробники мережі були узгоджені один з одним, щоб мати можливість забезпечити високошвидкісні I/O можливості. У сучасних технологіях досягнення пропускної здатності в кілька Гбіт/с традиційно через мережі на базі TCP стало тягарем.

Пропускна здатність, яку досягають сучасні IP-мережі, обмежена продуктивністю TCP, який є надійним протоколом передачі даних і основою стандартних протоколів передачі на рівні програми (наприклад, FTP, GridFTP). Протягом багатьох років було реалізовано багато варіантів протоколу TCP для покращення його продуктивності, однак він досі залишається найпоширенішим протоколом транспортування даних.

Окрім реалізованих транспортних протоколів низького рівня, існувала низка методів високого рівня для оптимізації пропускної здатності передачі даних незалежно від використовуваного базового протоколу.

Налаштування базових параметрів кінцевої системи та відкриття паралельних потоків є одним із способів зробити це та широко використовується в багатьох сферах застосування, від інтенсивних наукових обчислень до мультимедійних і однорангових парадигм.

Показано, що паралельні потоки досягають високої пропускної здатності шляхом імітації поведінки окремих потоків і отримують несправедливу частку доступної смуги пропускання [1, 3, 4, 11, 12, 16, 17]. З іншого боку, використання занадто великої кількості одночасних підключень призводить до того, що мережа досягає точки перевантаження, і після цього порогу досяжна пропускна здатність починає падати для низькошвидкісних мереж. На жаль, важко передбачити точку перевантаження, оскільки вона є змінною щодо деяких параметрів, які є унікальними як у часі, так і в домені передачі інформації. Таким чином, прогнозування оптимальної кількості потоків є дуже складним без деяких параметрів поточних умов мережі, таких як доступна пропускна здатність, RTT, швидкість втрати пакетів, пропускна здатність зв’язку вузького місця та розмір блоку даних.

У разі масової передачі даних легше оптимізувати параметри мережі. Через певний момент часу збирається достатньо статистичної інформації, а параметри оптимізуються відповідно до стану мережі. Цей тип оптимізації вже використовувався з планувальником даних Stork [2] раніше. Однак для окремих передач даних оптимізація стає складнішою. Замість того, щоб покладатися на статистичну інформацію, передачу слід оптимізувати на основі миттєвого зворотного зв’язку. Цю оптимізацію можна здійснити шляхом досягнення оптимальної кількості паралельних потоків для отримання найвищої пропускної здатності.

Однак техніка оптимізації, яка не покладається на статистичні дані, у цьому випадку не повинна спричиняти надто великі витрати через миттєвий збір даних, які будуть більшими, ніж прискорення, досягнуте кількома потоками для певного розміру даних. Миттєвий збір інформації для моделей прогнозування можна здійснювати за допомогою інструментів вимірювання

продуктивності мережі. Однак дуже важко вибрати інструмент вимірювання для ефективного та точного збору інформації. Незважаючи на складність, передбачення оптимального рівня паралелізму для передачі даних, незалежно від базових характеристик мережі, є неможливим (незалежно від того, чи це високошвидкісна мережа, або низькошвидкісна WAN або LAN).

З розвитком останніх мережевих технологій із високою пропускною здатністю, які досягають швидкості 100 Гбіт/с, вузькі місця в кінцевих системах стали основним фактором, який впливає на верхню межу швидкості наскрізної передачі даних. Кінцеві системи, які викликають і отримують передачу даних, можуть варіюватися від суперкомп'ютерів до обчислювальних кластерів або хостів із різними можливостями [3]. Вони мають потенційні вузькі місця, такі як пропускна здатність диска, швидкість процесора, MTU та розмір вікна. Однак ці вузькі місця можна подолати належним налаштуванням систем на основі властивостей і архітектури кінцевої системи.

Мережа є основним джерелом вузьких місць, особливо для архітектур з низькою пропускною здатністю. Системи, пов'язані з таким типом мережі, зазвичай складаються з одного процесора/вузла, доступу до одного диска та мережевих карт зі швидкістю до 1 Гбіт/с. Серед можливих причин низької продуктивності передачі даних можна назвати затримку, поточний мережевий трафік, використовувані протоколи та неналаштований розмір буфера. Хоча перші дві проблеми безпосередньо пов'язані з вузьким місцем мережі, дві останні пов'язані опосередковано, оскільки ці параметри налаштовуються на кінцевій системі.

Кінцеві системи, з'єднані з високошвидкісними мережами, зазвичай складаються з паралельних кластерів, суперкомп'ютерів або паралельних систем зберігання. Першою і головною проблемою в цих кінцевих системах знову ж таки є використовуваний протокол. Наприклад, протокол TCP не розроблений для таких типів мереж і не може використовувати більшу частину доступної пропускної здатності.

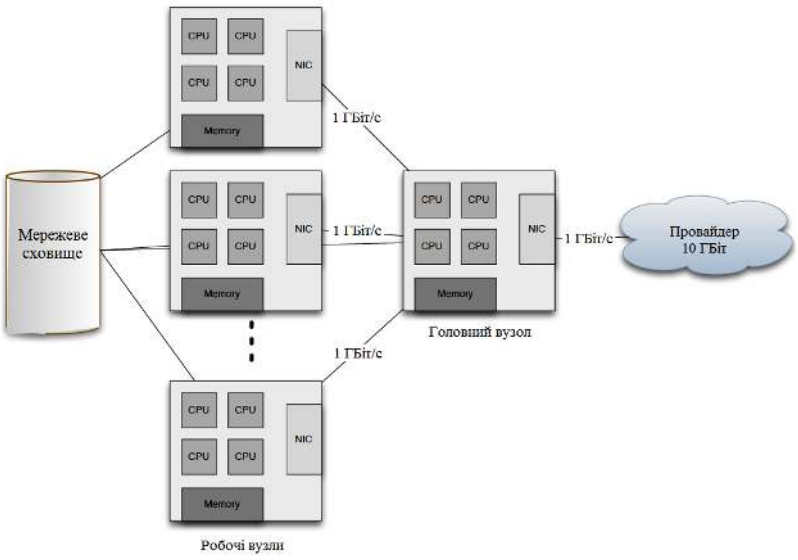

Рисунок 2.1 –Вузьке місце протоколу

Використання кількох паралельних потоків і правильний розмір буфера може вирішити проблему на прикладному рівні. На рисуннку 2.1 представлено приклад архітектури мережі.

У цій архітектурі головний вузол і робочі вузли мають NIC 1 Гбіт/с. За замовчуванням максимальний розмір буфера встановлено на 128 Кб. Без будь-якої оптимізації досягається пропускна здатність близько 100 Мбіт/с. Однак при правильному використанні паралельних потоків він збільшується приблизно до 900 Мбіт/с.

Використання паралельних потоків або великого розміру буфера створює навантаження на ЦП, а також на мережевий адаптер. Для паралельних архітектур проблему може вирішити використання кількох ЦП та мережевих карт.

У типовій архітектурі розподіленої спільної пам’яті, хоча швидкість NIC на головному вузлі сумісна зі швидкістю мережі, на робочих вузлах вона може бути іншою.

Враховуючи, що доступ до одного диска вільний, можливі причини вузьких місць, які можуть виникнути в такій системі, можна назвати швидкістю ЦП, налаштуваннями протоколу та мережевою карткою на

робочих вузлах. На рисунку 2.2 показано приклад архітектури, у якій головний вузол має мережеву плату 10 Гбіт/с, тоді як робочі вузли мають мережеві карти 1 Гбіт/с або 10 Гбіт/с.

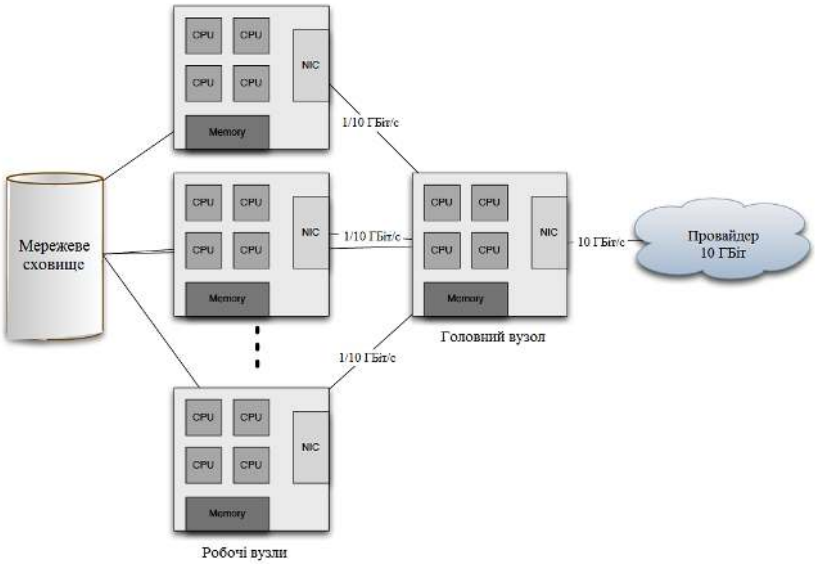

Рисунок 2.2 – Вузьке місце центрального процесора та мережевої карти

У першому випадку (1 Гбіт/с NIC на робочому вузлі) можуть виникнути дві проблеми: ЦП на головному вузлі може не конкурувати зі швидкістю NIC або, якщо це робочий вузол, сам NIC стає вузьким місцем. У другому випадку єдиною можливою причиною вузького місця є процесор, який розглядає передавання з пам’яті в пам’ять, де диск не є частиною наскрізного шляху передачі даних.

Багато паралельних систем, підключених через високошвидкісні мережі, мають паралельні системи зберігання.

Швидкість доступу до диска є основним джерелом вузьких місць у системах доступу до одного диска.

Однак у паралельних системах зберігання це вузьке місце можна подолати шляхом читання/запису даних у масивах дисків. Подолаючи це вузьке місце, можна створити нові вузькі місця, оскільки інші частини системи (наприклад, ЦП, мережева карта) можуть не впоратися з

навантаженням. У цьому випадку серед можливих причин вузького місця можна назвати швидкість диска, швидкість ЦП, параметри протоколу та мережевий адаптер. На рисунку 2.3 представлено приклад такої архітектури.

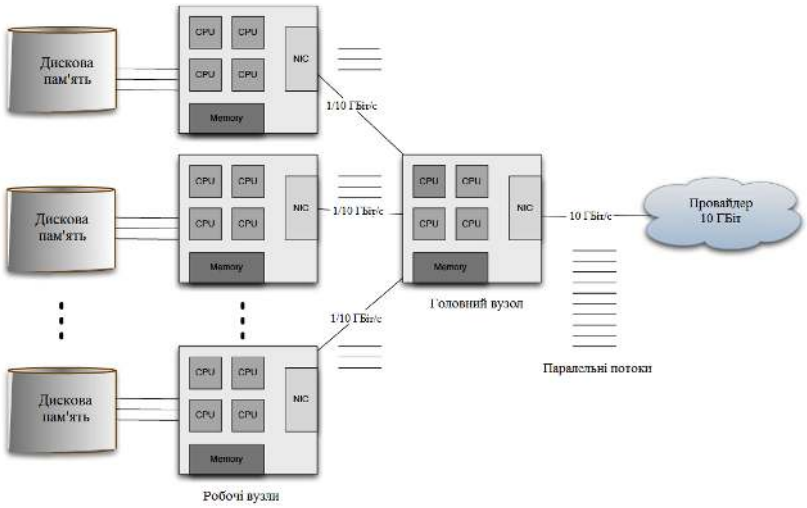

Рисунок 2.3 – Паралельний доступ до дисків

В кваліфікаційній роботі пропонується підхід на прикладному рівні для подолання вузьких місць протоколу та кінцевої системи та розробки моделей для забезпечення оптимальної пропускної здатності наскрізної передачі даних. Ці моделі визначають оптимальні параметри (наприклад, кількість паралельних потоків на масиві дисків, розмір буфера, кількість жорстких дисків на вузол, кількість вузлів) для конкретної передачі даних з урахуванням поточної архітектури систем і прогнозують доступну пропускну здатність для користувачів.

Існуючі високошвидкісні мережі страждають від неадекватності існуючих протоколів, які не були розроблені з урахуванням їх властивостей.

TCP – це протокол транспортного рівня, призначений для використання смуги пропускання мережі, що надає важливості чесності розподілу ресурсів мережі між потоками, які спільно використовують мережу. Він має дві фази, які називаються «Повільний старт» і «Уникнення заторів» (рисунок 2.4).

Розмір вікна перевантаження – це кількість пакетів, які відправник може надіслати. Чим більше вікно перевантаження, тим вище пропускна здатність. У фазі повільного запуску вікно перевантаження починається з 1 пакета та експоненціально збільшується, щоб швидко використовувати пропускну здатність, доки не станеться подія втрати пакета.

Потім вікно перевантаження ділиться навпіл і починає лінійно збільшуватися. Це відомо як властивість адитивного збільшення – мультиплікативного зменшення (AIMD) [4].

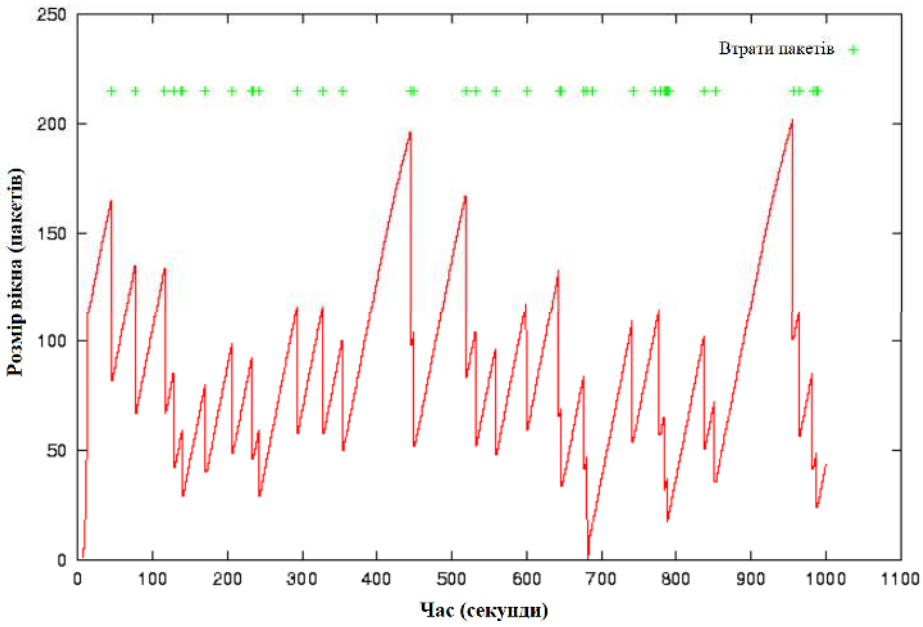

Рисунок 2.4 – Варіація вікна TCP на каналі з втратами

Коли відбувається тайм-аут, цикл повертається до початкового розміру вікна перевантаження та знову входить у фазу повільного запуску. Ця властивість TCP забезпечує справедливість, однак вона дає низьку продуктивність з точки зору пропускної здатності. Тому були розроблені інші методи компенсації його низької продуктивності.

Було проведено низку досліджень, які намагалися оптимізувати мережу та кінцеві системи для отримання максимальної наскрізної пропускної здатності з точки зору кількох параметрів, таких як кількість паралельних

потоків, розмір вікна, розмір MTU (максимальний блок передачі), розподіл навантаження за допомогою зв’язування IRQ (запит на переривання) та об’єднання переривань. Більшість оптимізацій виконується вручну та методів динамічної оптимізації мало.

2.2 Оптимізація паралельних потоків

Досліджень, які намагаються знайти оптимальну кількість потоків, дуже мало і вони здебільшого базуються на наближених теоретичних моделях [5, 8, 9, 13]. Усі вони мають певні обмеження та припущення. Крім того, правильність запропонованих моделей здебільшого підтверджується лише результатами моделювання.

Стверджується, що загальна кількість потоків поводиться як один гігантський потік, який передає загальну пропускну здатність кожного потоку [9]. У цій моделі досяжна пропускна здатність залежить від трьох параметрів: час проходження в обидві сторони, максимального розміру сегмента та швидкості втрати пакетів.

Наступне рівняння представляє верхню межу досяжної пропускної здатності n потоків:

$$\mathrm{Th} \le \mathrm{n}\frac{\mathrm{MSS}}{\mathrm{RTT}} \times \frac{\mathrm{c}}{\sqrt{\mathrm{p}}}, \tag{2.1}$$

де RTT являє собою час проходження пакету туди й назад, MSS являє собою максимальний розмір сегмента, p являє собою коефіцієнт втрати пакетів, а c є константою.

Однак ця модель працює лише для неперевантажених мереж і припускає, що рівень втрати пакетів є стабільним і однаковим для кожного з’єднання та не збільшується зі збільшенням кількості потоків. У той момент, коли мережа стає перевантаженою, швидкість втрати пакетів починає різко

зростати, а досяжна пропускна здатність починає зменшуватися. Тому важливо знайти точку перелому в коефіцієнті втрати пакетів. Загалом ця модель вимагає занадто багато інформації, яку також важко зібрати, і вона не дає відповіді на те, в який момент мережа буде перевантажена.

Деякі автори (Дінда та ін.) моделюють пропускну здатність кількох потоків як часткове рівняння другого порядку, що потребує вимірювання пропускної здатності двох різних номерів потоків, щоб передбачити інші пропускні здатності [14]. Модель намагається розв'язати рівняння 2.1 шляхом обчислення співвідношення між p, RTT і n. MSS і c вважаються константами. Відношення представлено новою змінною p'_n, яка прирівнюється до часткового полінома другого порядку, який вважається найкращим [15]:

$$p'_n = p_n \frac{RTT_n^2}{c^2 MSS^2} = a'n^2 + b'. \tag{2.2}$$

Після розміщення p'_n у рівнянні 2.1 загальна пропускна здатність n потоків обчислюється наступним чином:

$$Th_n = \frac{n}{\sqrt{p'_n}} = \frac{n}{\sqrt{a'n^2 + b'}}, \tag{2.3}$$

де a' і b' – параметри, які визначаються вимірюванням.

Щоб розв'язати це рівняння, потрібні два досяжні вимірювання пропускної здатності для двох різних рівнів паралельності. Єдиний можливий спосіб знайти ці значення пропускної здатності – використовувати інструмент, який має можливість виконувати паралельні передачі, або використовувати інформацію про минулі передачі. Експериментальні результати стверджують, що рівень паралелізму можна правильно передбачити, однак у цих результатах сукупна пропускна здатність з'єднань

збільшується, а потім стає стабільною, але ніколи не падає. Тому не враховується, що відкриття занадто великої кількості потоків може створити тягар для мережі та кінцевої системи та спричинити зниження пропускної здатності після оптимальної точки.

В іншій моделі загальна пропускна здатність завжди показує однакові характеристики [3] залежно від потужності з'єднання, оскільки кількість потоків збільшується, і 3 потоків достатньо для отримання 90% використання. Ця модель базує свою правильність на тому факті, що відкриття занадто великої кількості з'єднань створює накладні витрати на обробку та залишає меншу пропускну здатність для інших потоків. Слід зазначити, що для ситуації з одним потоком TCP-з'єднання може використовувати лише 75% пропускної здатності через властивість AIMD (адитивне збільшення та мультиплікативне зменшення). Тільки за допомогою паралельних потоків можна збільшити це співвідношення. Оскільки лише невелика підмножина з'єднань зазнає мультиплікативного зменшення під час перевантаження, це сприяє швидкому відновленню паралельних потоків.

Якщо припустити, що лише один із зв'язків зазнає мультиплікативного зменшення в будь-який час, наступну формулу можна використати для прогнозування пропускної здатності:

$$Th_n = C\left(1 - \frac{1}{1 + \frac{1+B}{1-B}n}\right). \qquad (2.4)$$

В дорівнює 1/2 для з'єднань TCP, оскільки воно зменшує своє вікно вдвічі для мультиплікативного зменшення, а C – це пропускна здатність з'єднання. Є дві проблеми, які необхідно вирішити, щоб застосувати цей підхід. Перш за все, потрібно знати пропускну здатність вузького місця зв'язку для загального з'єднання, щоб отримати правильні результати,

оскільки ця формула визначає пропускну здатність по одному каналу. По-друге, ця формула дає загальну кількість потоків, які підтримуються, щоб отримати цю сукупну пропускну здатність, однак, якщо потрібно знати додаткову кількість потоків для відкриття, необхідно мати уявлення про те, скільки інших потоків використовують посилання як перехресний трафік. Ми повинні знайти спосіб визначити існуючу кількість потоків.

Нове дослідження протоколу [17], яке регулює швидкість надсилання відповідно до обчисленого відставання, представляє модель для прогнозування поточної кількості потоків, яка може бути корисною для прогнозування майбутньої кількості потоків. Цільове відставання обчислюється на основі поточного виміряного вузького місця для досягнення заявлених цілей.

Для розрахунків потрібні чотири вхідні параметри: пропускна здатність вузького місця зв’язку, середній і мінімальний час проходження в обидві сторони та швидкість втрати пакетів. За допомогою цих вхідних даних можна отримати пропускну здатність перехресного трафіку та середню кількість потоків, які спільно використовують вузьке місце. Якщо відомо середню кількість потоків, які мають спільний доступ до вузького місця, можна поєднати це з моделлю в [3]. Скажімо, потрібно використовувати канал на 98% і відома пропускна здатність вузького місця. Можна обчислити оптимальне n і відняти середню кількість потоків, які спільно використовують канал. Отже, знаходимо число додаткових потоків для відкриття.

$$n_{add}=n_{opt}-n., \tag{2.5}$$

Однак інформацію для цих обчислень можна отримати на рівні протоколу низького рівня. Наприклад, швидкість надсилання може визначатися лише основним протоколом, який залежить від динамічного налаштування розміру вікна.

2.3 Оптимізація розміру буфера

Ще один важливий параметр, який потрібно налаштувати для високої пропускної здатності – це розмір буфера TCP. Параметр розміру буфера впливає на максимальну кількість пакетів, які будуть на льоту, перш ніж відправник чекатиме на підтвердження. Якщо мережевий буфер занижений, мережа не може бути використана повністю. Однак, якщо він занадто великий, пропускна здатність також може знизитися через втрату пакетів, що призведе до зменшення вікна передачі. Зазвичай він налаштовується вручну користувачами програми або на рівні ядра операційної системи.

Загальний спосіб налаштування розміру буфера полягає в тому, щоб встановити його на подвійне значення пропускної здатності помножене на затримку (BDP). Однак це припущення про збільшення розміру буфера, ніж подвійний BDP, справедливе лише тоді, коли на шляху немає перехресного трафіку, що абсолютно неможливо. Таким чином, виникає питання, чи використовувати пропускну здатність мережі чи доступну пропускну здатність, а також мінімальне RTT чи максимальне RTT. Отже, існує досить різноманітне розуміння концепцій пропускної здатності та затримки. Нижче наведено список різних значень BDP [15]:

BDP1: $B = C \times RTT_{max}$;
BDP2: $B = C \times RTT_{min}$;
BDP3: $B = A \times RTT_{max}$;
BDP4: $B = A \times RTT_{min}$;
BDP5: $B = BTC \times RTT_{ave}$;
BDP6: $B = B_{\infty}$.

У наведених вище рівняннях B представляє розмір буфера, C представляє ємність з’єднання, A представляє доступну пропускну здатність, а RTT – час проходження в обидва кінці. BTC у BDP5 – це середня

пропускна здатність масової передачі з обмеженим перевантаженням, розрахована на основі розміру вікна перевантаження. Нарешті, B∞ у BDP6 є великим значенням, яке завжди перевищує вікно перевантаження, тому з'єднання завжди буде обмежено перевантаженням.

Більшість методів налаштування, описаних у літературі, здійснюють налаштування на рівні ядра, а підходів, які використовують автоматичне налаштування на рівні програми, дуже мало. Методи, які використовують одне з наведених вище рівнянь, зазвичай покладаються на інструменти для вимірювання доступної смуги пропускання та RTT і не враховують ефект перехресного трафіку та заторів, створених використанням великих розмірів буфера.

Методи, які потребують модифікації ядра, як правило, базуються на динамічних змінах розміру буфера під час передачі на основі вікна перевантаження або параметрів вікна керування потоком. На основі поточного вікна перевантаження, RTT і часу читання сервера вони обчислюють наступне перевантаження та встановлюють змінні буфера на основі поточного та наступного розмірів вікна перевантаження. Ці змінні визначають верхню межу буфера сокета, перш ніж він зможе прийняти дані від програми, а також нижню межу вільного простору та поточної кількості підтверджених байтів. У цьому сенсі цей підхід схожий на BDP5.

Буфер прийому є достатньо великим, щоб не обмежувати пропускну здатність. Інші два конкурентоспроможні методи, які широко використовуються, це Dynamic Right Sizing [15] і Linux 2.4 Auto-Tuning [16].

DRS – це в основному підхід на основі одержувача, коли одержувач намагається оцінити добуток пропускної здатності на затримку, використовуючи інформацію заголовка пакета TCP і мітки часу. Замість використання статичних вікон керування потоком, оголошене вікно отримання динамічно змінюється, щоб відправник не був обмежений керуванням потоком. З іншого боку, автоматичне налаштування Linux – це техніка керування пам'яттю, за якої розмір вікна постійно збільшується або

зменшується на основі доступної пам’яті та буферного простору сокета. Технологія, яку краще використовувати, може залежати від характеристик передачі даних. Наприклад, у той час як автоматичне налаштування Linux 2.4 добре підходить для великої кількості малих з’єднань, Dynamic Right Sizing є кращим для меншої кількості великих з’єднань, таких як масова передача даних або FTP.

Крім того, затримка є ще одним важливим фактором для вибору техніки налаштування розміру вікна. Хоча розмір вікна налаштування не має великого впливу на передачу з невеликою затримкою, він має великий вплив на передачу з великою затримкою.

Існують також методики, які застосовуються на прикладному рівні без зміни ядра та протоколу. Ці методи зазвичай є статичними, і після встановлення розміру буфера він не змінюється під час передачі. Передбачувана пропускна здатність з’єднання обчислюється на основі ймовірності втрати пакетів, RTT і тайм-ауту повторної. Потім за допомогою затримки BDP визначається розмір буфера:

$$\text{Buffersize} = \text{EstimatedThroughput} \times \text{RTT}.$$

Незважаючи на те, що параметр розміру буфера налаштовано належним чином, він не показує кращої продуктивності, ніж використання паралельних потоків, оскільки паралельні потоки швидше відновлюються після втрати пакетів, ніж один потік, налаштований буфером. Використання налаштованого розміру буфера для одного потоку забезпечує гарну продуктивність у порівнянні з неналаштованим потоком. Використання паралельних потоків забезпечує навіть кращу продуктивність, хоча буфер не налаштований. Хороша комбінація налаштованого розміру буфера з паралельними потоками може ще більше підвищити продуктивність. Однак із налаштованим розміром буфера потрібна менша кількість потоків у порівнянні з неналаштованими паралельними потоками (рисунок 2.2).

Якщо вдасться досягти хорошого балансу між розміром буфера та паралельними потоками, можна знайти оптимальну комбінацію, у якій можливо уникнути надмірної кількості потоків, але в той же час отримати кращу продуктивність.

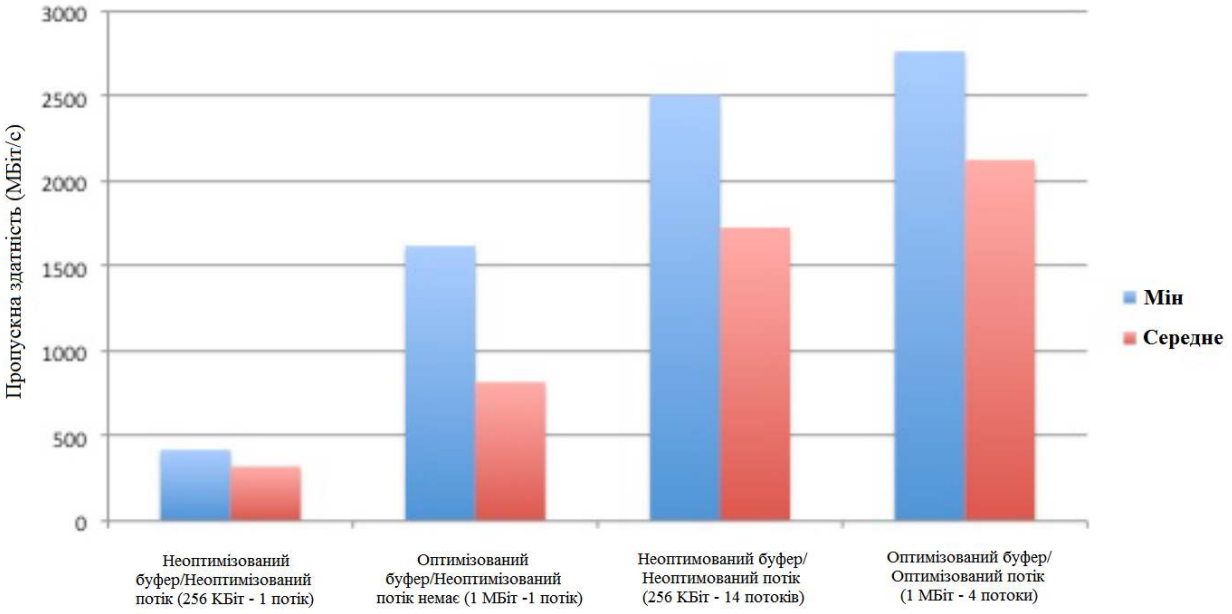

Рисунок 2.5 – Результати роботи FTP з використанням налаштованих TCP-буферів і паралельних потоків

2.3 Паралелізм ЦП і диска

Розвиток високошвидкісних мереж призвів до того, що кінцеві системи стали вузьким місцем для високої пропускної здатності даних, а не мережа. Основні вузькі місця включають жорсткий диск і обмеження ЦП. Деякі оптимізації для подолання вузького місця диска включають налаштування розміру блоку вводу/виводу, налаштування планувальника вводу/виводу та попередню вибірку [17].

Розмір блоку вводу/виводу визначає кількість байтів, які одночасно читаються/записуються з/на диск. Планувальники вводу/виводу можуть працювати по-різному в залежності від навантаження системи та

характеристик передачі. Хоча деякі планувальники працюють краще за інші на завантажених машинах, вони можуть працювати гірше на неактивних машинах.

Крім того, вони відрізняються використанням ЦП. Попередня вибірка може підвищити швидкість диска шляхом попереднього читання великих розмірів даних, однак вона може збільшити час доступу для малих розмірів даних.

Навіть якщо всі параметри налаштовані оптимально, доступна мережа з високою пропускною здатністю може використовуватися не повністю та обмежуватися продуктивністю одного диска.

У цьому випадку оптимальний рівень розмежування передачі буде способом подолання цього вузького місця. Однак вартість такого підходу має бути ретельно розрахована, а моделі мають бути розроблені для визначення найкращого рівня смуг.

Другим великим вузьким місцем у використанні високошвидкісних мереж є ЦП. Одним із методів, який використовується для подолання цього вузького місця, є об’єднання переривань.

Коли на головному вузлі суперкомп’ютера або кластерної архітектури є висока частота переривань, зазвичай генерується лише одне переривання для кількох пакетів. Інша техніка називається IRQ bonding [17], де певні переривання розподіляються між центральними процесорами. Знову ж таки, дуже важливо визначитися з оптимальним рівнем паралелізму щодо використання ЦП.

GridFTP Striped Server [2] забезпечує архітектуру, яка підтримує смугу та часткову передачу файлів. Відповідно до цієї архітектури дані можуть бути розділені або чергуватися на кількох серверах, як у паралельній файловій системі або дисковому кеші (DPSS).

Джерела та приймачі даних можуть мати різні форми, такі як кластери з локальними дисками, кластери з паралельними файловими системами, системи архівного зберігання та територіально розподілені джерела даних.

Загальною конфігурацією можуть бути вузли, з'єднані каналами 1 Гбіт/с із комутатором, який підключено до зовнішнього вузла зі швидкістю 10 Гбіт/с або швидше. Експериментальні результати [2] показали, що вони можуть досягти 27,3 Гбіт/с передачі між пам'яттю та 17 Гбіт/с передачі з диска на диск у мережі 30 Гбіт/с.

У численних експериментах збільшення потоків не означало збільшення продуктивності, тільки коли вони наближалися до швидкості вузького місця, кількість потоків починала впливати. Коли потоки починають конкурувати між собою або переповнюють буфери маршрутизатора, виникає момент, коли потоки починають конкурувати між собою. Крім того, передача даних з диска на диск значною мірою залежить від швидкості читання та запису паралельних файлових систем, які вони використовували.

2.4 Аналіз вузьких місць кінцевої системи

Використання паралельних потоків є поширеним методом усунення неадекватності протоколу TCP для мереж з великою ємністю. Вони забезпечують кілька обсягів пропускної здатності, досягнутої одним потоком, за рахунок втрати справедливості серед інших потоків, що використовують мережу.

Для неперевантажених мереж ця ситуація не є проблемою, яка часто буває для високошвидкісних мереж, оскільки вони зазвичай мають невикористану пропускну здатність через неправильне використання кінцевих систем.

Однак використання надмірної кількості паралельних потоків також не є хорошим способом використання ресурсів. Є багато причин, які забороняють використання надмірних паралельних потоків.

Однією з причин є те, що можна досягти точки перевантаження в мережі, де швидкість втрати пакетів зростає та спричиняє падіння пропускної здатності.

Інша причина полягає в тому, що можна досягти перевантаження кінцевої системи до її меж, таких як навантаження ЦП, швидкість доступу до диска та доступна ємність мережевої карти.

Інший метод, який особливо використовується для подолання вузьких місць диска, це смуга, що означає доступ до різних частин даних на кількох дисках паралельно.

Пропускна здатність, яку отримують користувачі під час передачі даних через високошвидкісні мережі без будь-якої оптимізації, зазвичай є пропускною здатністю диска.

Загалом пропускна здатність зростає зі збільшенням кількості потоків, після досягнення максимальної точки вона або залишається стабільною, або починає зменшуватися через вузькі місця в мережі чи кінцевій системі.

Щоб зрозуміти джерело вузького місця, другий набір передач виконується шляхом додавання смуг і використання кількох ЦП на додаток до паралельних потоків.

У кожному випадку використовується діапазон паралельного потоку, який дає кращі результати пропускної здатності. ЦП є джерелом вузьких місць, а також надмірне використання смуг і паралельних потоків призводить до падіння пропускної здатності. Важко сказати, що кілька смуг завжди дадуть кращі результати залежно від стану мережі. Результати смуги отримують лише за допомогою одного вузла, але цього достатньо, щоб досягти межі пропускної здатності мережі.

Диск зазвичай є найповільнішою частиною наскрізної передачі даних. Щоб забезпечити більш швидкий доступ до даних, можна використовувати декілька дисків паралельно. Використовувані методи доступу поділяються на дві категорії: однофайловий паралельний і багатофайловий паралельний. У першому методі читання та запис виконуються з різних частин одного файлу, тоді як у другому кілька файлів читаються та записуються з послідовним кодом.

2.5 Вплив протоколу TCP

Втрата пакетів і час зворотного зв’язку є двома важливими показниками продуктивності мережі, які впливають на пропускну здатність TCP-з’єднань. Моніторинг цих двох значень у режимі реального часу в глобальній мережі або Wi-Fi може допомогає швидко визначити повільність, з якою стикаються кінцеві користувачі та програми.

Втрата пакетів обчислюється як відсоток пакетів, отриманих у неправильному форматі або не отриманих взагалі. Цей ключовий показник продуктивності є хорошим показником якості мережі, оскільки він точно відображає надійність мережі. Втрата пакетів спричинена тим, що адресат отримує неправильно сформовані пакети, або, що ще гірше, пакети не надходять взагалі.

Round-Trip Time – це час, потрібний для того, щоб пакет даних перейшов туди й назад до певного пункту призначення. Значення часу зворотного зв’язку залежить від довжини всіх мережевих посилань, а також від затримки, яка виникає на кожному стрибку, включаючи хост призначення.

Трафік даних на основі протоколу керування передачею (TCP) є домінуючим у мережах IP. Детальний аналіз функцій і поведінки TCP є гарячою темою останніх дослідницьких програм. Найважливішу інформацію про TCP можна знайти в RFC 793, в якому спочатку було визначено TCP, тоді як RFC 1122 і RFC 2001 містять додаткові доповнення та описано кілька додаткових функцій.

Оскільки з’єднання TCP здатні забезпечувати та інтерпретувати зворотний зв’язок, вони адаптуються до різних умов мережі, які відповідають фактичному сценарію, з яким вони стикаються. Якщо з’єднання TCP має вузьке місце з неадаптивним фоновим потоком трафіку, TCP адаптується до нього, успадковуючи та поширюючи структуру кореляції та статистичні властивості фонового потоку трафік.

Ця часова шкала адаптації залежить від властивостей наскрізного шляху, тобто часу проходження туди й назад, розміру вікна тощо.

Адаптація TCP до фонового трафіку також впливає на пропускну здатність з’єднання. Особливо, коли фоновий трафік коливається навколо характерного часового масштабу, згаданого вище, пропускна здатність TCP-з’єднань значно зменшується. Цей ефект майже не залежить від поширених версій TCP.

Існує механізм зворотного зв'язку, реалізований у TCP, який відповідає за контроль перевантаження потоку. Він регулює видкість джерела відповідно до змінних умов мережі.

Швидкість джерела знижується у разі втрати пакета. Адаптивність можна чітко спостерігати, якщо потік TCP поділяє пропускну здатність з іншим трафіком у вузькому місці. Якщо фоновий трафік не є адаптивним, наприклад, заснований на протоколі дейтаграм користувача (UDP), TCP використовує вільну ємність на каналі відповідно до спеціальних характеристик. У більшості мереж адаптивний і неадаптивний трафік передається через одну і ту ж інфраструктуру.

3 МЕРЕЖЕВА МОДЕЛЬ

3.1 Модель оптимізації паралельного потоку

Розроблена модель передбачає поведінку паралельних потоків, і протестувана у різних мережевих сценаріях, щоб довести їх правильність. Модель є адаптивною до середовища мережі та кінцевої системи та робить прогнози з невеликою кількістю статистичної інформації або безпосередньою інформацією про прогнозування, отриманою з інструментів прогнозування продуктивності мережі.

Навіть найбільш прості у застосуванні моделі мають низьку точність прогнозування, а інші вимагають багато інформації. Розроблено кілька моделей, які базуються на моделях Дінди [17] і Хакера [10], які можуть прогнозувати поведінку паралельних потоків у більш точному наближенні.

3.1.1 Моделювання швидкості втрат пакетів

Недолік існуючих моделей полягає в тому, що немає інформації про те, коли відбудеться точка перевантаження в результаті відкриття кількох потоків. Причина такого висновку полягає в тому, що поведінка рівня втрат пакетів із збільшенням кількості потоків є непередбачуваною.

Однак, якщо знайти модель для характеристики коефіцієнта втрати пакетів, тоді можна використати рівняння 2.1 для розрахунку пропускної здатності, отриманої n потоками. Щоб досягти цього, можна використовувати методологію, подібну до тієї, що описана в моделі Дінди [17]. Однак цього разу не можна використовувати частковий поліном другого порядку для моделювання швидкості втрат пакетів, оскільки вона зростає експоненціально, коли пропускна здатність зростає логарифмічно. Поміняви місцями Th_n і p_n у рівнянні 2.1, ми отримаємо рівняння 3.1.

$$p_n = \frac{MSS^2 c^2 n^2}{RTT^2 Th_n^2}. \quad (3.1)$$

У цьому випадку визначається нова змінна Th'_n і співвідноситься з RTT, MSS, n і Th_n. Хакер та ін. припускає, що пропускна здатність зростає лінійно в неперевантажених мережах зі збільшенням кількості потоків, однак це не вірно для перевантажених мереж. Пропускна здатність, досягнута n потоками, зростає логарифмічно, тому ми визначаємо Th'_n у такому рівнянні:

$$Th'_n = \frac{RTT^2 Th_n^2}{MSS^2 c^2} = a' n^{\frac{1}{x}} + b'. \quad (3.2)$$

Встановлюючи x від 2 і більше, можна дослідити, наскільки різким буде збільшення втрати пакетів після точки перевантаження. Розмістивши Th'_n у рівнянні 3.1, отримаємо таке рівняння для втрати пакетів n потоків:

$$p_n = \frac{n^2}{Th'_n}. \quad (3.3)$$

Враховуючи, що ми можемо зібрати показники втрат пакетів для передачі з двома різними номерами потоків p_{n_1} і p_{n_2}, можна знайти значення a' і b'.

$$a' = \frac{\frac{n_2^2}{p_{n_2}} - \frac{n_1^2}{p_{n_1}}}{n_2^{\frac{1}{x}} - n_1^{\frac{1}{x}}}. \quad (3.4)$$

$$b' = \frac{n_1^2}{p_{n_1}} - a' n_1^{\frac{1}{x}}. \quad (3.5)$$

Після визначення значення p_n ми можемо легко обчислити значення пропускної здатності n потоків за допомогою рівняння 2.1. Однак це значення являє собою верхню межу досягнутої пропускної здатності.

3.1.2 Збільшення підгонки кривої з додатковою інформацією

Дослідження в [17] показує, що характеристика кривої пропускної спроможності різко зростає, а потім стає стабільною, доки не досягне пропускної здатності зв'язку, тому представлена модель відповідає представленим даним. Однак у реальному експериментальному середовищі з використанням реальних протоколів передачі файлів (наприклад, GridFTP) результати можуть відрізнятися від запропонованої ситуації.

На малюнку 3.1 показано передачу файлу розміром 512 МБ за допомогою GridFTP через глобальну мережу з використанням до 40 паралельних потоків. Зі збільшенням кількості потоків досягнута пропускна здатність також зростає.

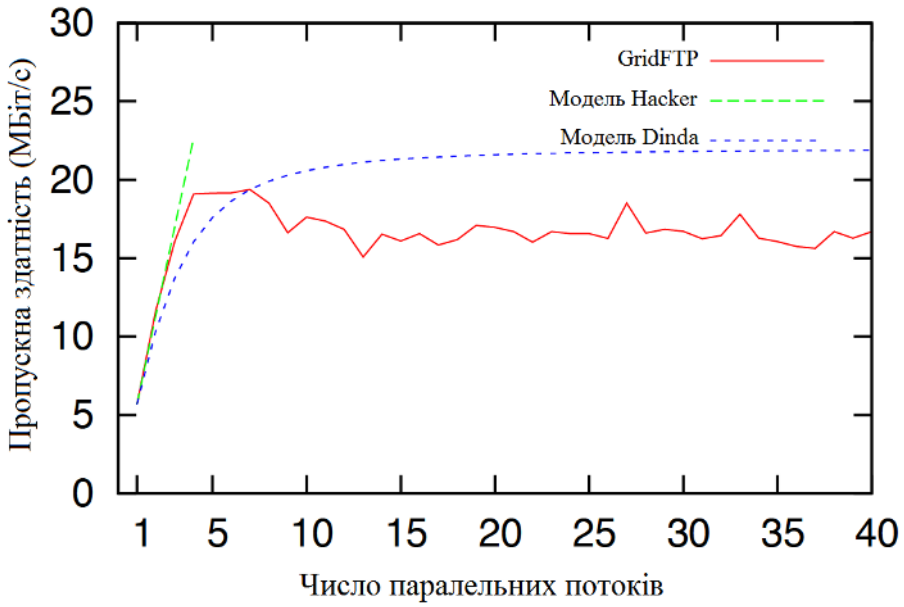

Рисунок 3.1 – Результати сукупної пропускної здатності GridFTP для передачі файлу розміром 512 МБ через мережу із затримкою 155 мс

Однак через деякий момент затори, створені через відкриття занадто великої кількості потоків, спричиняють зменшення пропускної здатності.

Існуючі моделі не можуть передбачити таку поведінку. Модель Hacker правильно прогнозує пропускну спроможність до моменту, коли починається перевантаження та швидкість втрати пакетів починає збільшуватися. Модель Dinda може правильно передбачити поведінку пропускної здатності до моменту, коли пропускна здатність почне зменшуватися, однак вона не може передбачити частину кривої пропускної здатності, яка зменшується.

Замість використання двох вимірювань пропускної здатності для двох різних рівнів паралельності планується збільшити цей рівень інформації до трьох значень вимірювання. Однак накладні витрати на збір додаткової інформації не повинні перевищувати фактичну швидкість, досягнуту відкриттям паралельних потоків.

У цьому випадку можна застосувати дві різні методики, щоб використати додаткову інформацію. Спочатку розраховуються a' і b' для використання рівнів паралельності n_{12} і n_{13} усереднюються за допомогою методів арифметичного, геометричного або квадратичного усереднення. Тоді пропускна здатність може бути розрахована з усередненими значеннями a' і b'.

По-друге, як можна побачити на рисунку 3.1, крива продуктивності діє як дві різні функції. До досягнення точки огляду (спаду) вона діє як певна функція.

Однак при падінні вона демонструє інші характеристики. Отже, замість використання однієї функції можна розбити функцію на дві. Використовуючи рівні паралелізму n_1 і n_2, можна змоделювати певну функцію, а використовуючи n_2 і n_3, можна змоделювати другу частину. Перехід між двома функціями може бути трохи різким, однак це вказує на те, що оптимальна кількість потоків, безперечно, знаходиться між n_2 і n_3. У наступному розділі представленомо модель, яка згладжує різкий перехід між двома функціями.

3.1.3 Логарифмічне моделювання кривої пропускної здатності

Зв’язок між p, RTT і n моделюється частковим поліноміальним рівнянням другого порядку в [16].

Однак у деяких випадках лінійне або повне поліноміальне рівняння другого порядку може дати кращі результати.

Звісно, що швидкість втрати пакетів зростає в геометричній прогресії. Однак можна не знати порядку використання рівняння. У цьому випадку замість часткового полінома другого порядку використовується експоненціальне рівняння, порядок якого може змінюватися залежно від параметрів a' і b'.

Наступне рівняння використовується для визначення змінної p'_n, яку ми згадували раніше:

$$p'_n = a' e^{b'n}, \tag{3.6}$$

$$Th_n = \frac{n}{\sqrt{a' e^{b'n}}}. \tag{3.7}$$

За допомогою двох вимірювань пропускної здатності Th_1 і Th_2 з різними рівнями паралельності n_1 і n_2 обчислюються наступні значення для a' і b':

$$a' = \frac{n_1^2}{Th_{n_1}^2 \cdot e^{b'n_1}}, \tag{3.8}$$

$$b' = \log_{e^{n_1 - n_2}} \frac{Th_{n_2}^2}{Th_{n_1}^2} \cdot \frac{n_1^2}{n_2^2}. \tag{3.9}$$

3.1.4 Динамічне виділення порядку рівняння моделі за допомогою ітерації Ньютона

Порядок рівняння слід отримувати динамічно. Логарифмічне моделювання пропускної здатності, яке пояснюється в попередньому пункті, здатне здійснювати плавний перехід від зростаючої до спадної частини кривої передбачення пропускної здатності.

Однак із збільшенням номера потоку крива прогнозування наближається до 0 і може не давати приблизний результат прогнозування щодо фактичної пропускної здатності в частині кривої, що зменшується. Щоб мати можливість зробити хороший прогноз, сформулюємо p'_n шляхом додавання нової змінної для прогнозування порядку таким чином:

$$p'_n = a' n^{c'} + b' . \tag{3.10}$$

У цьому випадку c' є невідомим порядком рівняння, додаткового до a' і b'. Таким чином, наше формулювання пропускної здатності виглядає так:

$$Th_n = \frac{n}{\sqrt{a' n^{c'} + b'}} . \tag{3.11}$$

Щоб вирішити це рівняння, потрібні три вимірювання Th_{n_1}, Th_{n_2} і Th_{n_3} на кривій пропускної здатності для значень потоку n_1, n_2 і n_3. Крім того, c' в степені n значно ускладнює розв'язання рівняння. Після кількох підстановок отримуємо такі рівняння для a', b' і c':

$$\frac{n_3^{c'} - n_1^{c'}}{n_2^{c'} - n_1^{c'}} = \frac{\frac{n_3^2}{Th_{n_3}^2} - \frac{n_1^2}{Th_{n_1}^2}}{\frac{n_2^2}{Th_{n_2}^2} - \frac{n_1^2}{Th_{n_1}^2}} . \tag{3.12}$$

$$a' = \frac{\frac{n_2^2}{Th_{n_2}^2} - \frac{n_1^2}{Th_{n_1}^2}}{n_2^{c'} - n_1^{c'}}. \tag{3.13}$$

$$b' = \frac{n_1^2 - \frac{n_1^2}{Th_{n_1}^2}}{Th_{n_1}^2} - a' n_1^{c'}. \tag{3.14}$$

Виведення a' і b' залежить від c'. Щоб розв'язати перше рівняння, застосуємо математичний метод знаходження кореня, який називається методом ітерації Ньютона:

$$c'_{x+1} = c'_x - \frac{f(c'_x)}{f'(c'_x)}. \tag{3.15}$$

Згідно з цим методом, після x + 1 ітерацій можна знайти дуже близьке наближення до c'. Починаючи з невеликого числа для c'_0, продовжимо обчислення до c'_{x+1}. Значення найбільш наближеного c' залежить лише від $f(c')$, у цьому випадку від першого рівняння вище, та його похідної.

Після обчислення найбільш приблизного c', яке можливо лише за кілька ітерацій, можна легко обчислити значення a' і b'.

3.1.5 Повна модель другого порядку

Повна модель другого порядку може передбачити збільшення, а потім зменшення характеристик пропускної здатності паралельного потоку зі збільшенням кількості потоків.

Що стосується повної моделі другого порядку, можна зробити припущення, що p'_n пов'язане з повним поліномом другого порядку, відмінним від часткового полінома другого порядку, який був представлений

раніше. Додавання лінійного члена до моделі призведе до низки змін у відповідних рівняннях. Наступні рівняння отримані для використання в цій моделі.

$$p'_n = p_n \frac{RTT_n^2 Th_n^2}{MSS^2 c^2} = a'n^2 + b'n + c'. \tag{3.16}$$

Відповідно до рівняння 3.16 ми отримуємо:

$$Th_n = \frac{n}{\sqrt{p'_n}} = \frac{n}{\sqrt{a'n^2 + b'n + c'}}. \tag{3.17}$$

Щоб отримати значення a', b' і c', представлені в рівнянні 3.17, потрібні значення пропускної здатності трьох різних рівнів паралелізму (Th_{n_1}, Th_{n_2} і Th_{n_3}), які можна отримати з передбачень інструментів вимірювання мережі. або попередніх передач даних.

$$Th_1 = \frac{n_1}{\sqrt{a'n_1^2 + b'n + c'}}. \tag{3.18}$$

$$Th_2 = \frac{n_2}{\sqrt{a'n_2^2 + b'n + c'}}. \tag{3.19}$$

$$Th_3 = \frac{n_3}{\sqrt{a'n_3^2 + b'n + c'}}. \tag{3.20}$$

Розв’язуючи наступні три рівняння, ми можемо помістити змінні a', b' і c' у рівняння 3.17, щоб обчислити пропускну здатність будь-якого рівня паралелізму. На основі рівнянь 3.21, 3.22 і 3.23 можна легко обчислити значення a', b' і c'.

$$a' = \frac{\frac{n_3^2}{Th_{n_3}^2} - \frac{n_1^2}{Th_{n_1}^2}}{n_3 - n_1} - \frac{\frac{n_2^2}{Th_{n_2}^2} - \frac{n_1^2}{Th_{n_1}^2}}{n_2 - n_1}}{n_3 - n_2}. \quad (3.21)$$

$$b' = \frac{\frac{n_2^2}{Th_{n_2}^2} - \frac{n_1^2}{Th_{n_1}^2}}{n_2 - n_1} - (n_1 + n_2)a'. \quad (3.22)$$

$$c' = \frac{n_1^2}{Th_{n_1}^2} - n_1'a' - n_1b'. \quad (3.23)$$

Кожна модель, представлена в попередніх розділах, може показати свою найкращу продуктивність, якщо вибіркові дані про пропускну здатність для розрахунку прогнозованої пропускної здатності можна вибрати з відповідних рівнів паралелізму. Усі моделі потребують 2 або 3 даних пропускної спроможності з різними рівнями паралелізму. Отже, існує багато видів комбінацій, якщо є більше трьох пар (n, Th_n) даних, і важливо знайти найкращу комбінацію, щоб мінімізувати відстань між статистичними або прогнозними даними та обчисленою пропускною здатністю n потоків на основі представлених моделей.

Кількість комбінацій занадто велика, тому пропонується інтелектуальна стратегія вибору, яка визначає меншу кількість даних. Попередній досвід показав, що краще, якщо вибераються рівні паралельності не близько один до одного. Застосувується стратегія експоненціального збільшення, вибраються числа потоків, які є степенем 2: 1, 2, 2^2, 2^3, ..., 2^k. Кожного разу подвоюється кількість потоків, поки пропускна здатність не почне падати або зростати дуже повільно порівняно з попереднім рівнем. Після k+1 кроків збираються k+1 дані про пропускну здатність.

Схема стратегії вибірки представлена в лістингу 3.1.

Лістинг 3.1 – Псевдокод алгоритму стратегії вибірки

```
Output: ThN: набір значень пропускної здатності для різних рівнів
паралелізму від алгоритму вибірки
i←1, p←1
ThNi←getThroughput(p)
i←i+1
p←p×2
ThNi←getThroughput(p)
while ThNi>THNi-1 and ThNi−THNi-1>Precision do
  i←i+1
  p←p×2
  ThNi←getThroughput(p)
end
return i, ThN
```

Інший алгоритм необхідний для вибору цих точок даних серед доступних значень (Th_N), і він представлений в лістингу 3.2.

Лістинг 3.2 – Псевдокод алгоритму вибору точок вимірювання

```
Input: ThN: набір значень пропускної здатності для різних рівнів
паралелізму з алгоритму вибірки n
i←1
j←i+1
k←j+1
for i<=n−2 do
  for j<=n−1 do
    for k<=n do
     //Обчислити a′, b′i c′
     //Обчислити Predictedijk для рівнів паралельності від 1 до N
      m=1
      for m<=N do
        err+=abs(Predictedm-Thm)
        m=m+1
      end
      if err<minerr then
        mini←i
        minj←j
        mink←k
        minerr=err
      end
    end
  end
end
return mini, minj, mink
```

Алгоритм вибірки надає набір значень пропускної здатності для експоненціально зростаючих паралельних потоків. Однак потрібні лише три точки даних, щоб застосувати модель прогнозування. Для всіх можливих комбінацій рівнів паралельності в порядку зростання застосовується модель і обчислюються параметри a′, b′ і c′ для передбачення. Розраховується відстань між прогнозованою та фактичною пропускною здатністю, яка дає значення помилки. Повертається комбінація значень паралельності з мінімальним значенням помилки.

4 АЛГОРИТМИ ОПТИМІЗАЦІЇ

Обговорення в розділі 3 показує, що використання лише паралельних потоків недостатньо для використання великої пропускної здатності мережі. У певних ситуаціях центральний процесор, диск або мережевий інтерфейс можуть бути джерелом вузького місця.

4.1 Алгоритм оптимізації для невідомої дискової та мережевої пропускної здатності

Алгоритм приймає як вхідні дані набір значень:

- пропускну здатність для експоненціально зростаючих рівнів паралелізму, отриманих за допомогою алгоритму вибірки, представленого в лістингу 3.1;
- пропускну здатність мережевого інтерфейсу;
- потужність центрального процесора вузла призначення;
- доступну кількість ресурсів.

Робиться ряд припущень щодо застосування алгоритму. Висновок із попередньої глави вказує на те, що використання процесора м завжди більше, ніж використання джерела.

Враховуючи, що процесори неактивні на початку передачі, а однорідні системи використовуються як джерело та призначення, прийнятно застосовувати використання ЦП вузла призначення для пропускної здатності потоку. Результатом алгоритму є набір значень пропускної здатності вибірки, отриманих із використанням запропонованої кількості вузлів, кількості смуг на вузол і кількості паралельних потоків на смугу.

Пропускна здатність потоку та кількість потоків, що використовуються для цієї пропускної здатності, визначаються оптимальною пропускною здатністю, розрахованою моделлю Ньютона на основі рівнів паралельності,

вибраних алгоритмом вибору.Перш ніж збільшити номер смуги вибірки, встановлюється кількість паралельних потоків вибірки на смугу на число вибірки перед тим, як оптимальний номер потоку та пропускну здатність для цього номера потоку буде встановлено для значення однієї смуги (рядки 4-5). Це значення відповідає змінній X_{fk} у моделі потоку. Значення X_{ij} залежить від дуги.

Якщо це дуга мережі або дискової системи, це відповідає пропускній здатності загальної кількості смуг, інакше це пропускна спроможність смуг на вузол.

Якщо ємність цього потоку досягає ємності NIC, тоді значення смуги на вузол встановлюється як 1 для кожного вузла, оскільки подальше збільшення смуги в тому самому вузлі може не покращити пропускну здатність.

У цьому випадку номер вузла експоненціально збільшується разом із поточним потоком і номером смуги, доки пропускна здатність мережі не буде досягнута та пропускна здатність не почне падати, або всі доступні вузли будуть використані (рядки 6-16). Алгоритм повертає значення пропускної здатності вибірки для смуг і кількості вузлів, кількості смуг на вузол і кількості потоків на смугу.

Якщо ліміт мережевої картки не досягнуто, це означає, що ще є місце для збільшення номера смуги в тому самому вузлі. Однак вузьким місцем може бути пропускна здатність процесору ЦП і дискова або мережева пропускна здатність.

По-перше, залишкова пропускна здатність ЦП і мережевої карти обчислюється шляхом віднімання пропускної здатності поточного рівня смуги (рядок 18).

Використання ЦП перетворюється на одиницю пропускної здатності за допомогою регресійної моделі.

Крива пропускної здатності зазвичай показує характеристики логарифмічного збільшення в експериментах.

Лістинг 4.1 – Алгоритм оптимізації

```
Input: ThN: набір значень пропускної здатності для рівнів паралелізму та відповідних значень використання ЦП UNIC, Navail, UCPUdest
Output: ThS: набір значень пропускної здатності для різних значень вузла, смуги, потоків, кількості паралельних потоків на смугу (Nk), кількості смуг на вузол (Sx), кількості вузлів (Nn)
1 j←1, Sx←1, Nn←, Limit←0
2 Uf← Оптимальна пропускна здатність на основі моделі ThN
3 Nopt← Оптимальна кіл-ть потоків визначається моделлю на основі ThN
4 NSi← Номер вибірки паралельного потоку перед Nopt
5 ThSj← Пропускна здатність смуги для потоку Sx←ThNSi
6 if Uf≈UNIC then
7   while Limit=0 && 2*Nn<=Navail do
8     Nn←Nn×2
9     ThSj+1←getThroughput(Nn,Sx,NSi)
10    if ThSj+1<ThSj then
11      Limit←1
12    end
13    j←j+1
14  end
15  return ThS, Nn, Sx,Nsi
16 end
17 else if Uf<UNIC then
18  UCPUleft←UCPU−ThNsi, UNICleft←UNIC−ThNsi, Uleft←min(UCPUleft, UNICleft)
19  PI← Покращення пропускної здатності←ThNSi−ThNSi−1
20  while 2×PI<Uleft do
21    Sx←Sx×2
22    ThSj+1←getThroughput(Nn, Sx, NSi)
23    PI←ThSj+1−ThSj
24    if ThSj+1<ThSj then
25      Limit←1 Break the loop
26    end
27    UCPUleft←UCPU−ThSj+1, UNICleft←UNIC−ThSj+1, Uleft ← min(UCPUleft,UNICleft)
28    j←j+1
29  end
30  if Limit=0 then
31    while 2*Nn<=Navail do
32      Nn←Nn×2
33      ThSj←getThroughput(Nn,Sx,NSi)
34      if ThSj<ThSj-1 then
35        Limit←1 Break the loop
36      end
37      j←j+1
38    end
39  end
40  return TS, Nn, Sx, NSi
41 end
```

Таким чином, різниця пропускної здатності наступного інтервалу вибірки не може перевищувати удвічі порівняно з попереднім інтервалом, оскільки збільшення кількості вибірки експоненціально за ступенями двох. Обчислюємо цю різницю, щоб вирішити, чи є місце для додаткової смуги в тому самому вузлі (рядок 19).

4.2 Алгоритм оптимізації для невідомого диска та відомої ємності мережі

Алгоритм припускає, що і вузли джерела, і вузли призначення однорідні. Однак у випадках, коли вихідний і цільовий кластери мають різну архітектуру, важливо враховувати як джерело, так і призначення. Наприклад, модель і номер процесора, а також архітектура чи з'єднання можуть відрізнятися на обох сторонах.

У цьому алгоритмі змінюєтьсяпопередній алгоритм таким чином, щоб ці проблеми були враховані.

Вхідними параметрами, необхідними для алгоритму, є набір значень пропускної здатності з алгоритму вибірки (Th_N) разом із значеннями використання ЦП, потужністю мережевого адаптера джерела та призначення (U_{NICsrc}, $U_{NICdest}$), доступними номерами вузлів ($N_{availsrc}$, $N_{availdest}$) і ЦП ємності (U_{CPUsrc}, $U_{CPUdest}$).

Хоча це значення менше, ніж ліва ємність ЦП і мережевої карти, значення смуги експоненціально збільшується на 2 (рядки 20-29). Однак цей цикл може припинитися, якщо рівень пропускної здатності почне знижуватися, що вказує на те, що ємність мережі або диска досягнуто (рядки 24-26).

Якщо ліміт мережі та диска не досягнуто, але ємність вузла перевищено (рядок 30), тоді встановлюється поточне значення такої смуги та потоку для вузла, кількість вузлів в такому разі збільшується переважно експоненціально.

Лістинг 4.2 – Алгоритм оптимізації для відомої пропускої здатності мережі

```
Input: ThN, UNICsrc, UNICdest, Navailsrc, Navaildest, UCPUsrc, UCPUdest
Output: ThS, Nk, Sxsrc,Sxdest, Nnsrc,Nndest
42  j←j+1, Sxsrc←1, Sxdest←1, Nnsrc←1, Nndest←1, Limit←0
43  Uf← Оптимальна пропускна здатність на основі моделі ThN
44  Nopt← Оптимальна кількість потоків на основі ThN
45  NSi← Номер вибірки паралельного потоку перед Nopt
46  ThSj←ThNSi
47  Uminsrc←min(UCPUsrc, UNICsrc)
48  Umindest←min(UCPUdest, UNICdest)
49  if Uf≈Uminsrc then
50    UCPUleftdest←UCPUdest−ThSni
51    UNICleftdest←UNICdest−ThSNi
52    Uminleftdest←min(UCPUleftdest, UNICleftdest)
53    PI←ThNSi−ThNSi-1
54    while 2×PI<Uminleftdest && Nnsrc×2<=Navailsrc do
55      Sxdest←Sxdest×2
56      Nnsrc←Nnsrc×2
57      ThSj+1←getThroughput(Nnsrc,Sxsrc,Nndest,Sxdest,NSi)
58      if ThSj+1<ThSj then Limit←1 Break the loop
59      end
60      UCPUleftdest←UCPUdest−ThSj+1
61      UNICleftdest←UNICdest−ThSj+1
62      Uminleftdest←min(UCPUleftdest,UNICleftdest)
63      j←j+1
64    end
65  else if Uf≈Umindest then
66    Повторити 50-63 з обміном мітками src і dest
67  else if Uf< min(Uminsrc, Umindest) then
68    Повторити 50-63 для обох значень src і dest
69  PI←ThNSi−ThNSi-1
70  while 2×PI<Uminleftdest||2×PI<Uminleftsrc do
71    if 2×PI <Uminleftdest && 2×PI<Uminleftsrc then
72      Sxsrc←Sxsrc×2
73      Sxdest←Sxdest×2
74      Повторити строки 57-59
75      Повторити строки 60-62 для обох значень src і dest
76      j←j+1
77    else if 2×PI<Uminleftdest && Nnsrc×2<=Navailsrc then
78      Повторити строки 55-63
79    else if 2×PI<Uminleftsrc && Nndest×2<=Navaildest then
80      Повторити строку 78 шляхом обміну мітками src і dest
81    else
82      Break the loop
83    end
84  end
85 end
```

Використовується значення цієї смуги та потоку до досягнення ємності або доступний номер ресурсу досягнуто (рядки 31-39). Нарешті номер вузла, номер смуги та номер потоку повертаються разом із значеннями пропускної здатності для подальшої оцінки.

Результатом роботи алгоритму є набір значень пропускної здатності разом із його потоком на смугу, смугою джерела та призначення на номери вузлів і кількість вузлів джерела та призначення.

Рядки 42-46 показують схожість із початком попереднього алгоритму. Мінімальні значення потужностей NIC і CPU розраховуються як для джерела, так і для вузла призначення. Якщо оптимальна пропускна здатність уже досягла максимальної потужності вихідного вузла (рядок 49), обчислюється ліва пропускна здатність вузла призначення (рядки 50-51), а також значення PI. Якщо все ще є місце для подальшого збільшення значення смуги у вузлі призначення, а також доступний номер вузла джерела не досягнуто, тоді номер смуги для призначення збільшується разом із номером вузла для джерела експоненціально (рядки 53-54). Значення пропускної здатності для поточного потоку, смуг і значень вузла зчитується (або шляхом вибірки, або з історії).

Якщо поточне значення пропускної здатності менше попереднього значення пропускної здатності, це означає, що диск або мережа досягли своєї межі (рядки 58-59). У цьому випадку цикл зупиняється, інакше обчислюється нова ліва (меньша) ємність вузла призначення (рядки 60-62).

Якщо вузол призначення досяг своєї потужності, повторюються ті самі кроки з рядків 49-63, але цього разу шляхом заміни джерела та вузла призначення на змінні. Якщо жоден із вузлів не досяг своєї ємності, це означає, що можна збільшувати кількість смуг на тому ж вузлі джерела та вузлі призначення, доки будь-який із вузлів не досягне своєї ємності (рядки 70-83). У продовженні алгоритму як вузол джерела, так і вузол призначення досягають своєї потужності, і збільшення кількості вузлів неминуче, доки не буде перевищено доступну кількість вузлів вузлів (лістинг 4.3).

Лістинг 4.3 – Продовження алгоритму оптимізації

```
if Limit=0 then
  while Nnsrc×2<=Navailsrc && Nndest×2<=Navaildest do
  end
  Nnsrc←Nnsrc×2
  Nndest←Nndest×2
  Thsj+1←getThroughput(Nnsrc,Sxsrc,Nndest,Sxdest,NSi)
  if Thsj+1<Thsj then
    Break the loop
  End
end
return TS, Nnsrc, Nndest, Sxsrc, Sxdest, NSi
```

5 БАЛАНСУВАННЯ РОЗМІРУ БУФЕРА ТА ПАРАЛЕЛЬНИХ ПОТОКІВ

5.1 Експериментальна модель

Була проведена велика кількість досліджень у галузі оптимізації буфера, і результати щодо оптимізації паралельного потоку є дуже багатообіцяючими. Проте вдале поєднання налаштованого розміру буфера та паралельних потоків може навіть дати більш ефективні результати, ніж поодиноке застосування цих двох методів. На жаль, немає практичної роботи щодо балансування розміру буфера та кількості паралельних потоків для досягнення оптимальної пропускної здатності.

У цій главі представляються результати та обговорення моделювання, виконаного на NS-3, щоб описати спосіб встановлення балансу.

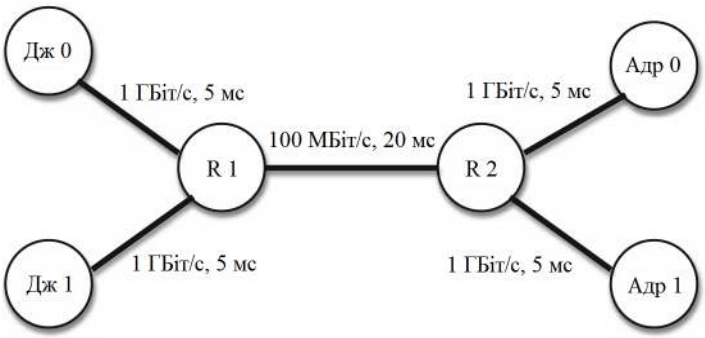

Рисунок 5.1 – Експериментальна модель

5.2 Результати моделювання та їх обговорення

В проведених експериментах виконувалися різні сценарії, змінюючи розмір буфера та кількість паралельних потоків. Використана топологія мережі представлена на рисунку 5.1. Пропускна здатність вузького місця становить 100 Мбіт/с із затримкою 20 мс. Джерела (Дж 0, Дж 1) для передач і

перехресного трафіку підключені до маршрутизатора з вузьким мсцем R0 із пропускною здатністю 1 Гбіт/с і затримкою 5 мс, а вузли призначення (Адр 0, Адр 1) підключені до R1 із пропускною здатністю 1 Гбіт/с і затримкою 4 мс. Перехресний трафік протікає від Дж 0 до Адр 0, тоді як фактична передача відбувається між Дж 1 і Адр 1.

У першій серії експериментів не використовувався перехресний трафік і змінювався розмір буфера за допомогою паралельних потоків. З дуже малим розміром буфера, таким як 16 Кбайт, повного використання мережі можна досягти лише за допомогою дуже великої кількості потоків (рисунок 5.2). Після досягнення максимальної точки пропускна здатність починає падати при близько 45 потоках. При збільшенні розміру буфера до 32 КБайт пікову пропускна здатність починає падати при 22 потоках.

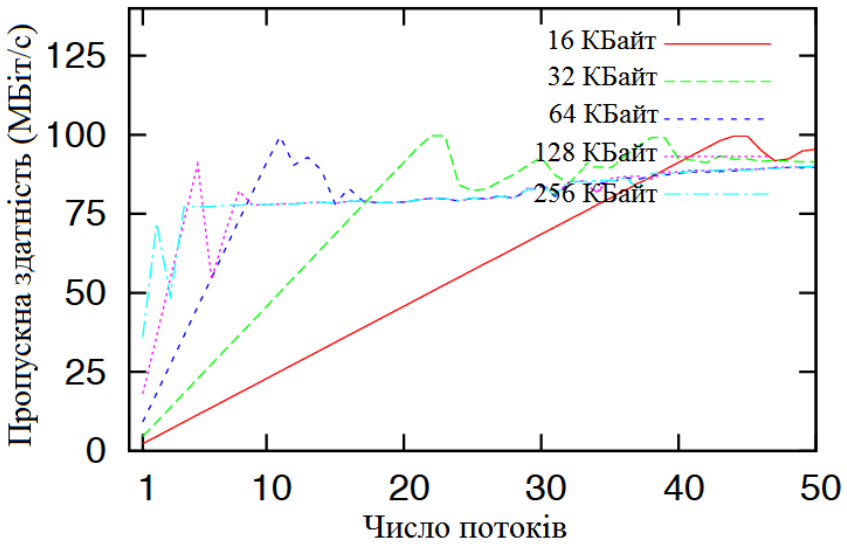

Рисунок 5.2 – Вплив паралельних потоків без перехресного трафіку

Подальше збільшення розміру буфера до 64 КБ кількість потоків до 10. Однак для більшого розміру буфера, ніж 64 КБ, пропускна здатність ніколи не може досягти максимальної точки. Розумним вибором у цьому випадку є використання розміру буфера приблизно 16-64 КБ і паралельних потоків приблизно 10-45 відповідно, щоб отримати найвищу пропускну здатність. У

цьому випадку подальше збільшення розміру буфера не допомагає, а призводить до зменшення досягнутої пропускної здатності. Максимальні значення пропускної здатності можна отримати за допомогою меншого розміру буфера, ніж корисна пропускна здатність (BDP), і використання паралельних потоків.

На рисунку показана хвилеподібна поведінка пропускної здатності: коли збільшується розмір буфера та зменшується кількість паралельних потоків, вона врешті-решт зменшується у своїй піковій точці.

На рисунку 5.3 порівнюються паралельні потоки з різним розміром буфера. Можна побачити, що більша кількість потоків може отримати більшу пропускну здатність за меншого розміру буфера.

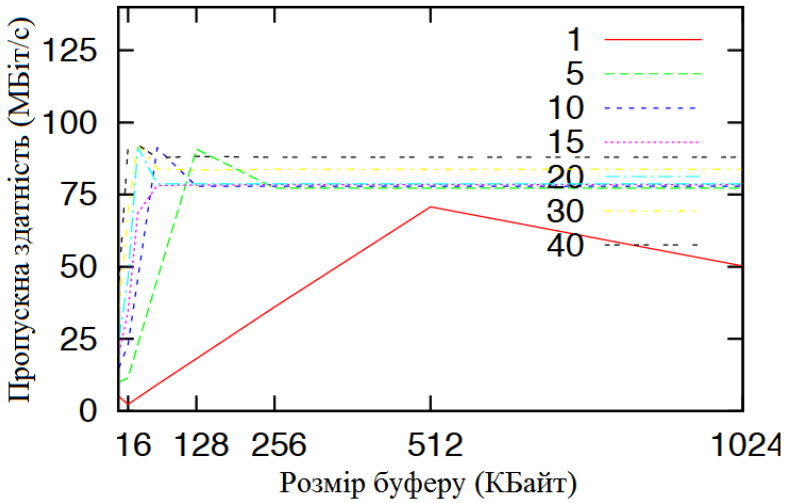

Рисунок 5.3 – Вплив розміру буфера без перехресного трафіку

У другій серії експериментів моделювалося існування не перевантаженого перехресного трафіку із 5 потоків з розміром буфера 64 КБ. Результати виявилися дуже цікавими.

При дуже маленькому розмірі буфера 16 КБ пропускна здатність збільшується лінійно до точки перевантаження без перехресного трафіку у міру збільшення кількості паралельних потоків (рисунок 5.4). Пропускна спроможність розподіляється між двома трафіками.

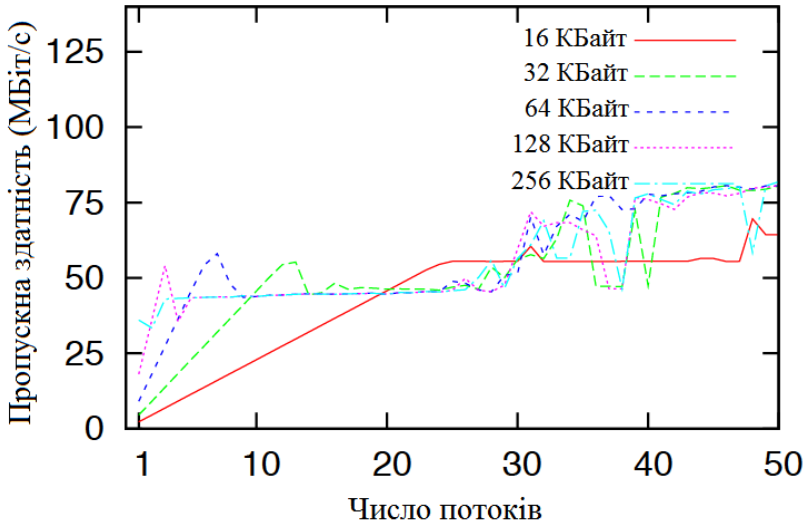

Рисунок 5.4 – Сумісний вплив паралельних потоків і розміру буфера без перехресного трафіку

Подальше збільшення розміру буфера збільшує точку максимальної пропускної здатності для меншої кількості потоків. Така поведінка схожа на попередній випадок, де немає перехресного трафіку, за винятком того, що пропускна здатність є спільною.

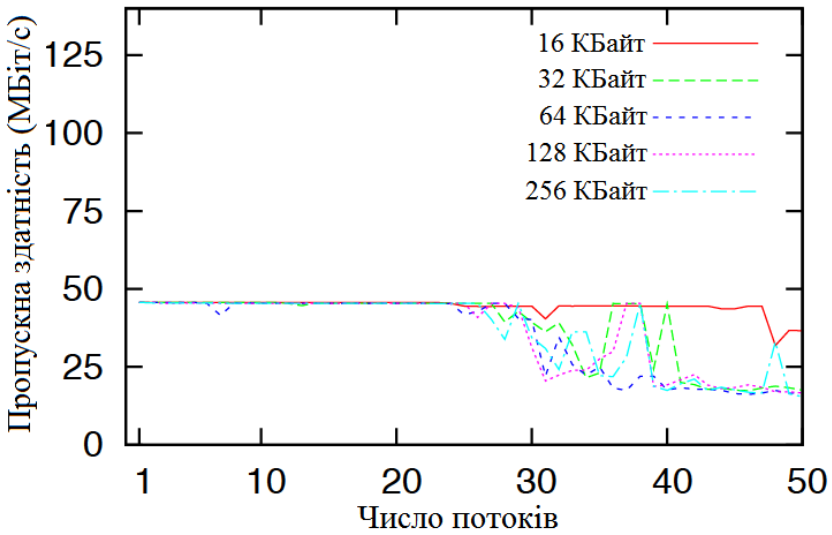

Рисунок 5.5 – Сумісний вплив паралельних потоків і розміру буфера з перехресним трафіком

У той же час немає ніякого впливу на перехресний трафік, оскільки кількість паралельних потоків збільшується (рисунок 5.5). Однак подальше збільшення кількості паралельних потоків трафіку призводить до того, що крос-трафік програє боротьбу, оскільки загальна пропускна спроможність нашого трафіку починає збільшуватися, а пропускна здатність крос-трафіку починає зменшуватися. Найкращі результати пропускної здатності без впливу на перехресний трафік у цьому випадку – це розмір буфера 32-64 КБ із діапазоном паралельних потоків 6-13.

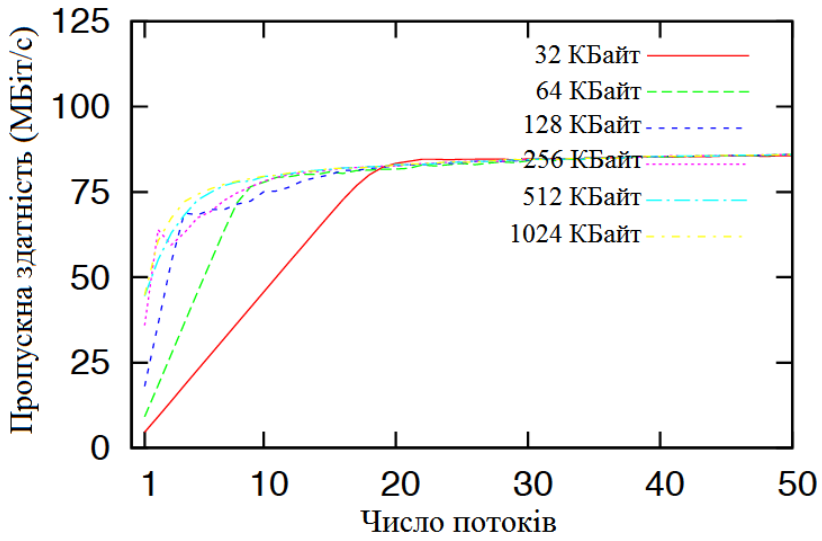

Рисунок 5.6 – Вплив паралельних потоків і розміру буфера з випадково згенерованим перехресним трафіком

У третьому випадку використано бібліотеку NS2 під назвою PackMimeHTTP для випадкового генерування Інтернет-трафіку на вузькому каналі. Бібліотека генерувала HTTP-запити випадкових розмірів і з кількістю з’єднань 200 за мілісекунди. Результати представлені на рисунку 5.6. Відповідно до цих результатів із більшим розміром буфера максимальна пропускна здатність знову досягається швидше з меншою кількістю потоків.

ВИСНОВКИ

Розвиток високошвидкісних мереж, які з’єднують мережі, суперкомп’ютери та інші паралельні системи, а також відсутність протоколу транспортного рівня, який би використовував раціонально мережеві ресурси, привели до необхідності динамічної оптимізації цих ресурсів на прикладному рівні.

Під час виконання кваліфікаційної роботи було виконано наступні завдання:

- забезпечено оптимізацію пропускної здатності передачі даних на прикладному рівні без необхідності змінювати транспортні протоколи;

- розроблено модель, в якій використовується якомога менше інформації, водночас забезпечуючи точність і масштабованість незалежно від архітектури кінцевих систем;

- проведено імітаційне моделювання, яке підтвердило теоретичні викладки, представлені в кваліфікаційній роботі.

Представлені в кваліфікаційній роботі моделі надають можливість дінамічно обирати оптимальні параметри для отримання найвищої наскрізної пропускної здатності за допомогою існуючих протоколів і інструментів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Navraj Chohan, “An Analysis of TCP through Simulation”, Technicalre port CS 276, 2006.

2. R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. Network Flows. Prentice Hall, 1993.

3. W. Allcock, J. Bresnahan, R. Kettimuthu, and M. Link. The globus striped gridftp server. In Proc. IEEE Super Computing Conference, page 54, 2005.

4. E. Altman, D. Barman, B. Tuffin, and M. Vojnovic. Parallel tcp sockets: Simple model, throughput and validation. In Proc. IEEE Conference on Computer Communications (INFOCOM’06), pages 1-12, April 2006.

5. H. Balakrishman, V. N. Padmanabhan, S. Seshan, and R. H. Katz M. Stemm. Tcp behavior of a busy internet server: Analysis and improvements. In Proc. IEEE Conference on Computer Communications (INFOCOM’98), pages 252-262, California, USA, March 1998.

6. K. M. Choi, E. Huh, and H. Choo. Efficient resource management scheme of tcp buffer tuned parallel stream to optimize system performance. In Proc. Embedded and ubiquitous computing, Nagasaki, Japan, December 2005.

7. A. Cohen and R. Cohen. A dynamic approach for efficient tcp buffer allocation. IEEE Transactions on Computers, 51(3):303-312, March 2002.

8. T. Dunigan, M. Mathis, and B. Tierney. A tcp tuning daemon. In Proc. IEEE Super Computing Conference (SC’02), Baltimore, Maryland, USA, November 2002.

9. L. Eggert, J. Heideman, and J. Touch. Effects of ensemble tcp. ACM Computer Communication Review, 30(1):15-29, January 2000.

10. W. Gropp, E. Lusk, and R. Thakur. Using MPI-2:Advanced Features of the Message-Passing Interface. The MIT Press, 1999.

11. T. J. Hacker, B. D. Noble, and B. D. Atley. The end-to-end performance effects of parallel tcp sockets on a lossy wide area network. In Proc. IEEE

International Symposium on Parallel and Distributed Processing (IPDPS’02), pages 434-443, 2002.

12. T. J. Hacker, B. D. Noble, and B. D. Atley. Adaptive data block scheduling for parallel streams. In Proc. IEEE International Symposium on High Performance Distributed Computing (HPDC’05), pages 265-275, July 2005.

13. G. Hasegawa, T. Terai, T. Okamoto, and Murata M. Scalable socket buffer tuning for highperformance web servers. In International Conference on Network Protocols(ICNP’01), page 281, 2001.

14. The lustre file system. http://wiki.lustre.org.

15. T. Ito, H. Ohsaki, and M. Imase. On parameter tuning of data transfer protocol gridftp for wide-area networks. International Journal of Computer Science and Engineering, 2(4):177-183, September 2008.

16. M. Jain, R. S. Prasad, and C. Davrolis. The tcp bandwidth-delay product revisited: network buffering, cross traffic, and socket buffer auto-sizing. Technical report, Georgia Institute of Technology, 2003.

17. Chadi Barakat, “TCP/IP Modeling and Validation”, IEEE Network, vol.15Issue 3, pp: 38-47, May 2001.

18. Лушпа Б.Є., Куриленко А.О., Янковський О.А. «Управління трафіком мереж», Тринадцята міжнародна науково-технічна конференця «Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління». –Баку-Харків-Жиліна-2023. – С. 103.