

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
на Всеукраїнський конкурс студентських наукових робіт з технічної галузі
наук
“Телекомунікаційні системи та мережі”

СТУДЕНТСЬКА НАУКОВА РОБОТА
на тему:
«Створення універсальної моделі системи балансування навантаження в
телекомунікаційних мережах»

Шифр «_____»

2010

ЗМІСТ

Вступ

1 Аналіз об'єкта дослідження

1.1 Аналіз сучасних тенденцій розвитку телекомунікаційної сфери та послуг, що надаються

1.2 Аналіз системи розподілення навантаження в центрах обробки даних

1.3 Балансування навантаження в мережі LTE-SON за рахунок використання властивості самоорганізації

1.4 Висновки

2 Розробка універсальної моделі

2.1 Розробка аналітичної універсальної моделі для системи з розподільником потоків

2.2 Розробка програмної універсальної моделі для системи з розподільником потоків

2.3 Висновки

3 Адаптація моделі до конкретних систем балансування навантаження

3.1 Адаптація моделі до системи перерозподілу потоків в мережі LTE-SON

3.2 Застосування універсальної моделі для порівняння різних алгоритмів системи балансування навантаження в ЦОД

3.3 Висновки

Перелік використаної літератури

ВСТУП

Сьогодні стала загальноприйнятою думка про те, що мережа Internet трансформується в середовище з багатьма службами, в якій відбудеться злиття голосу, відео, даних. Об'єм потоків даних, що передаються у мережі Internet зростає в геометричній прогресії з одночасним прискоренням процесу об'єднання різнорідних потоків інформації. Великі провайдери служб Internet відреагували на зростання мережі шляхом реалізації трьох нових напрямів, які взаємно доповнюють один одного: розробкою структур розширюваних мереж, підвищенням пропускної спроможності та перерозподілом потоків даних (Traffic Engineering - TE).

При експлуатації інформаційних телекомунікаційних мереж однією з основних проблем є нерівномірність навантаження, що виникає, у різний час і на різних ділянках мережі. У результаті зменшується економічна ефективність використання недовантажених ділянок, а перевантажені ділянки погіршують якість послуг, що надаються. Проблема нівелюється впровадженням додаткових ресурсів, які перекривають надмірне навантаження. Ці ресурси можна знаходити двома способами: статично встановлювати додатковий запас, або динамічно перерозподіляти вільні ресурси всієї мережі. Так як ця нерівномірність навантаження є процесом стохастичним за часом, то ефективно її вирішити статичними методами не можливо.

Динамічні системи перерозподілу навантаження можна знайти у будь-якій області застосування сучасних телекомунікаційних мереж: розподільник навантаження у сучасних центрах обробки даних; протоколи керування потоками[1], які застосовуються в IP/MPLS, це складова частина концепції SON (мереж, що самоорганізуються), на базі якої краще розгортати мобільні мережі 4-го покоління[2]. Але всі ці системи використовують різні методи, алгоритми та принципи організації задля вирішення задачі балансування навантаження (перерозподілу потоків).

Такі системи є складними та дорогостоячими для реалізації. Їх ефективність дуже залежить від використаних методів аналізу одержуваного навантаження, систем моніторингу станів роботи різних ділянок мережі, методів керування, швидкості реакції на динамічні зміни характеру навантаження і т.д. Недооцінка будь-якого з факторів може призвести до погіршення ефективності самої системи, тобто до невиправданих витрат.

Тому є сенс промоделювати та дослідити їх програмними методами. Розробка програм до кожної із систем є дуже довгостроковою задачею, крім того багато методів та рішень в майже усіх системах будуть однаковими. Щоб прискорити процес вирішення цієї задачі потрібно розділити задачу на 2 структурні частини: розробку «універсальної» моделі (ядра програми, що реалізує загальні взаємодії та методи усіх систем) та адаптації ядра до різних систем.

Тема даної роботи є актуальною, оскільки все більше поширення отримують послуги ширококутного доступу до Internet[3]. Навантаження, що створює один абонент під час користування цієї послугою, змінюється у великих межах, а трафік, що генерують багато подібних абонентів, може легко призвести до перевантаження різних ділянок мережі на короткий період (у той же час недовантажувати мережу на інших ділянках). Це обумовлено особливостями стека протоколів TCP/IP[4]. Також актуальність підтверджується проведеними роботами в цьому напрямку, а в нещодавно створених стандартах (LTE, WiMAX), протоколах, технологіях (IP/MPLS) – робився акцент на вирішення поставленої проблеми.

Метою роботи є підвищення ефективності процесу дослідження та розробки засобів динамічного перерозподілу потоків у телекомунікаційних мережах за рахунок розробки спеціалізованого програмного пакету.

Задля реалізації мети були поставлені наступні задачі:

- аналіз сучасних тенденцій розвитку телекомунікаційної сфери та послуг, що надаються;
- аналіз існуючих методів балансування навантаження (розподілення потоків) в різноманітних системах;

- розробка «універсальної» аналітичної та програмної моделі для системи з розподільником навантаження;
- адаптація моделі до дослідження керування розподілу потоків у центрах обробки даних (ЦОД);
- адаптація моделі до дослідження балансування навантаження в мережах, що самоорганізуються.

В роботі вперше запропонована «універсальна» модель для системи, яка реалізує функції розподілення потоків. На базі моделі були створені ряд програм, що дозволяють аналізувати та досліджувати системи балансування навантаження, що використовуються в новітніх стільникових мережах 4-го покоління та в центрах обробки даних, заснованих на кластерних мережах.

1 АНАЛІЗ ОБ'ЄКТА ДОСЛІДЖЕННЯ

1.1 Аналіз сучасних тенденцій розвитку телекомунікаційної сфери та послуг, що надаються

Проведемо аналіз статистики телекомунікаційних послуг, що надаються, та тенденції їх змінення у нашій країні[5], та у лідера впровадження сучасних технологій – Швеції[6] (додаток А).

Аналізуючи статистичні дані, доходимо висновку: ринок стаціонарного зв'язку зменшується (додаткове підтвердження[7]), а мобільного (стільникового) зростає. Так само зростають темпи надання послуг Internet і широкосмугового доступу до цієї послуги. Ця тенденція підтверджується компанією MTC[8] і аналізом, що був проведений iKS-CONSTULTING[3] – рисунок 1.1:



Рисунок 1.1 Кількість приватних абонентів ШСД в Україні

Причиною цьому є те, що голос і відео – це лише один з різновидів даних, які можна передавати у пакетних Internet мережах. Саме поява нових технологій, що дозволили забезпечити гарну якість передачі голосу та відео у пакетних мережах, дала старт переходу надання всіх інформаційних послуг в одному

середовищі – це фактично ідея маркетингового ходу Triple Play, що набирає обертів.

Підтвердження цих тенденцій можна відзначити в статистиці зниження використання стаціонарного голосового зв'язку (зазначалося вище) та ефірного телебачення[9], а також в підвищенні попиту на послуги VoIP та зростанні відео-інтернет трафіку[10]. Зокрема в 2008 році частка прибутку VoIP в загальному прибутку послуг зв'язку в Росії не перевищувала 8 %, а за 2009 рік зростає майже вдвічі - до 15 %. Очікується, що наприкінці 2012 р. частка трафіку VoIP сягатиме 40%[11].

Звіт Morgan Stanley прогнозує взагалі у світі 103%-е зростання трафіку VoIP к 2014 р. – показник, який поступається лише темпам росту відео-трафіку. Важливу роль у цьому відіграла зростаюча популярність передачі даних в мобільних мережах за допомогою смартфонів. З лютого 2009 року по лютий 2010 року цей вид трафіку зріс на 193%[12].

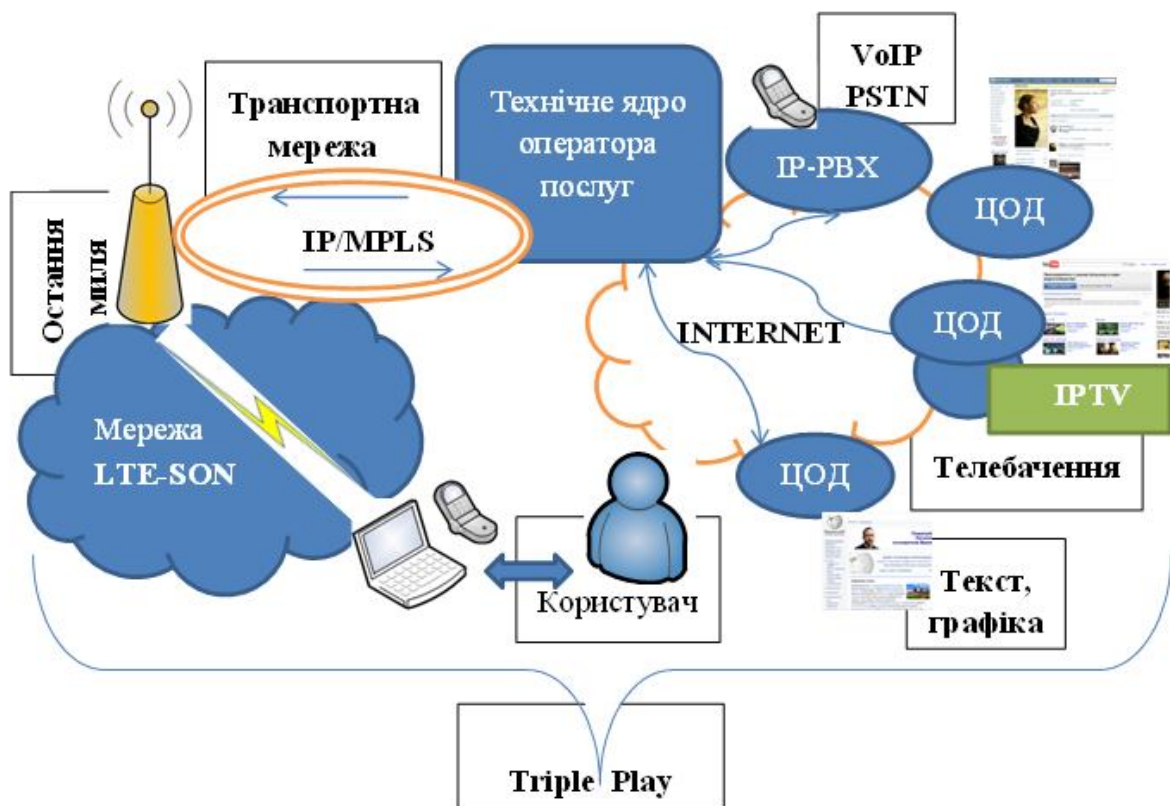


Рисунок 1.2 Мережа Triple Play, що побудована на новітніх технологіях

Можна підвести підсумок, що найбільш затребуваною послугою буде широкопasmовий доступ в Internet, та в межах цього доступу будуть реалізовані сервіси голосового зв'язку, телебачення, передачі даних. Відзначимо новітні технології, які акцентовані на надання послуг широкопasmового доступу: мобільні мережі 4-го покоління (LTE[2,13-17], WiMAX[18,19]), технологію надання мультисервісних послуг IP/MPLS[1,20,21], маркетингову концепцію Triple Play[22], сервіси розподілених обчислюваних центрів обробки даних[23-25].

Усі системи застосовуються для надання послуг ШСД на різних ділянках телекомунікаційної мережі. Їх загальну взаємодію та взаємопроникнення можна відобразити схемою – рисунок 1.2. Розглянемо деякі з цих систем більш детально, відзначивши застосування в них балансування навантаження.

1.2 Аналіз системи розподілу навантаження в центрах обробки даних

Вже зараз набули широкого поширення мультимедійні сайти, що надають можливості отримати одночасний доступ до будь-якого виду контенту: відео, текст, музика, графіка і т.д. Крім того, ці сайти є дуже великими та популярними, багато з них відвідують мільйони щодня[26]. Для реалізації такого сервісу необхідно мати величезний кластер обчислювальних систем, який зберігає інформацію та обробляє запити що надходять. Для оцінки навантаження, що надходить, наведемо дані компанії Twitter[27]:

- більш ніж 350000 користувачів (точна цифра є секретом фірми);
- близько 600 запитів в секунду;
- в середньому система підтримує 200-300 з'єднань на секунду (максимум досягається при значенні 800);
- MySQL обробляє приблизно 2400 запитів в секунду;
- в середньому відповідь на запит до бази даних складає 50-100 мілісекунд.

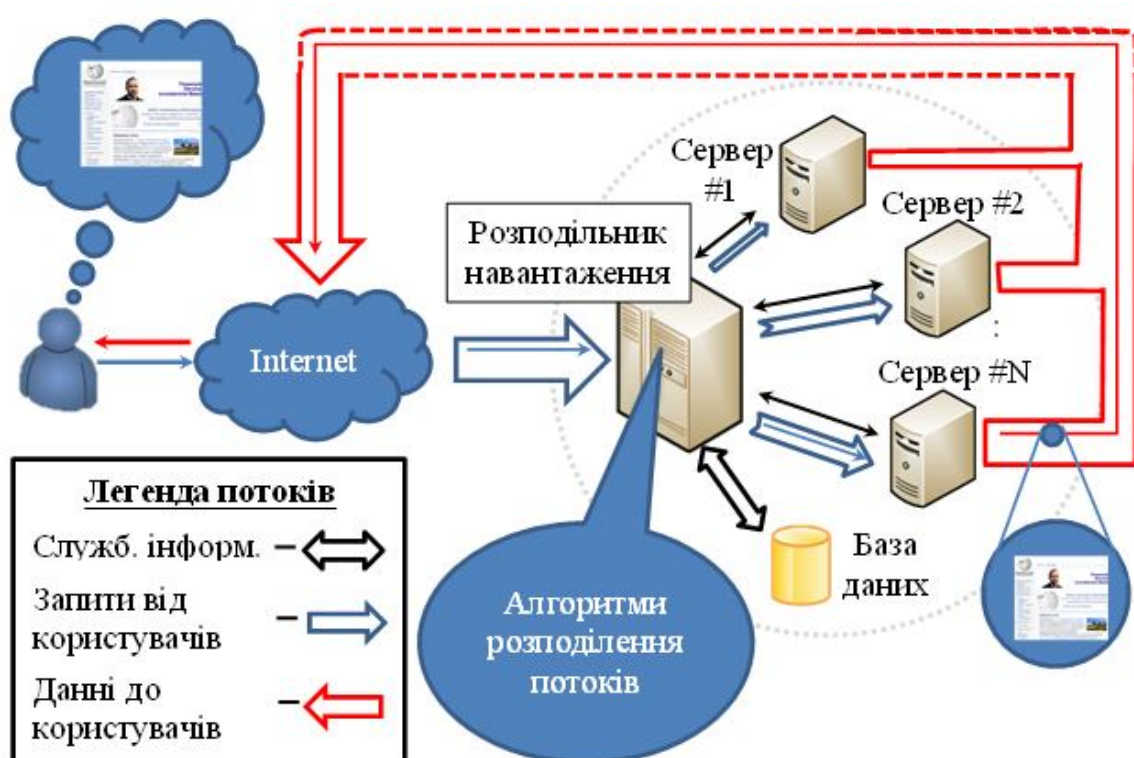


Рисунок 1.3 Схема роботи ЦОД з розподільником навантаження

Для організації й узгодження роботи таких кластерних обчислювальних центрів застосовується - розподільник навантаження, який приймає рішення про подальший рух кожного запиту (рисунок 1.3).

Пристрої, що випускаються серійно, використовують прості дисципліни обслуговування (алгоритми розподілу потоків), а в експериментальних пристроях застосовуються методи моделювання та прогностичного керування.

Алгоритми, що застосовуються, можна розділити на статичні та динамічні. Підвищена інтелектуальність динамічних алгоритмів тягне накладні витрати, пов'язані зі збором інформації з хостів, але краще адаптується до змін трафіку.

До статичних дисциплін відносять Random і Round Robbin. У Random сервер обирається випадковим чином за допомогою рівномірного розподілу. Дисципліна Round Robbin реалізує принцип циклічного перебору.

Для будь-якої динамічної дисципліни виникає питання вибору керуючого параметру. Як критерій ефективності зазвичай приймають два показники[23]:

- час перебування заявки на сервері;
- сповільнення (slowdown), у скільки раз уповільнюється обслуговування запиту через наявність черги.

Найбільш розповсюдженими динамічними дисциплінами є:

- Least Loaded (LL) – вибір сервера за критерієм найменшого завантаження його ресурсів (процесора, пам'яті, диска, кількості черг повідомлень);
- Least Connected (LC) – вибір сервера за критерієм найменшого числа поточних відкритих TCP/IP-з'єднань;
- Fast Response (FR) – вибір сервера за критерієм самої швидкої відповіді на тестовий запит від розподільника навантаження;
- Weighted Round Robbin (WRR) – при циклічному переборі кожному серверу, на відміну від статичної дисципліни RR, передається поспіль не один запит, а декілька, відповідно до ваги сервера, яка є пропорційною, наприклад, його поточному завантаженню[24].

- MC-RR (Multi Class Round Robbin) - всі можливі запити розділені за очікуваним впливом на мережу, процесор і диск: з високим навантаженням на диск; з високим навантаженням на процесор; з високим навантаженням на диск та процесор. Запити першого класу характерні для систем Web-публікацій, інші для обробки Web-транзакцій і мультимедійних Web-додатків. Запити одного типу (класу) обслуговуються згідно дисципліни RR на визначених для цього типу запитів серверах. З невеликими модифікаціями метод MC-RR відомий також як CAP(Client-Aware Policy).

Універсальний підхід до використання динамічних дисциплін було реалізовано в IBM Network Dispatcher[24,25]. Автори запровадили поняття індексу завантаження Load Metrics Index (LMI) і розділили показники якості роботи системи на три класи.

- Input – обчислюються локально в розподільнику. Наприклад, кількість з'єднань, встановлених із сервером за останні t одиниць часу.
- Host – обчислюються на кожному сервері. Наприклад, завантаження сервера.
- Forward – обчислюються шляхом мережевої взаємодії розподільника навантаження з сервером. Ці характеристики доставляють спеціальні скрипти-агенти. Наприклад, подача HTTP-запиту "GET /" і вимірювання часу відповіді.

Далі, за вектором LMI для кожного сервера обчислюються значення функції WCF (Weight Computation Function), і вибирається сервер із найбільшим її значенням.

Алгоритми є головною частиною розподільника навантаження, але окрім опису алгоритмів потрібно, для моделі, імітувати роботу серверів та генерувати потік вхідних запитів. Також з рисунка 1.3 бачимо, що розподільник є центральною ланкою у роботі ЦОД, що поєднує роботу усіх складових частин. Ці дані використаємо у наступній частині.

1.3 Балансування навантаження в мережі LTE-SON за рахунок використання властивостей самоорганізації

Технологія LTE та WiMAX створені спеціально для надання на ділянці останньої милі мобільного каналу доступу до мережі Internet із рівнем швидкості порівняним зі стаціонарними каналами доступу[17]. Ці технології ґрунтуються на модуляції сигналу OFDM і мають дуже високу ефективність використання частотного ресурсу (5-10 біт/с/Гц). Але у самих системах закладений ще один спосіб підвищення ефективності використання ресурсів мережі – побудова мережі на засадах самоорганізації. Self Organization Network (SON) легко інтегрується в LTE завдяки існуючим інтерфейсам, модулям та протоколам керування мережею[2].

Існують три основні складові частини концепції SON, які критичні для створення мережі[28]:

- **самоконфігурація** – plug-and-play, автоматичне створення базовою станцією переліку сусідніх станцій (ANR – Automated Neighbour Relation), автоматичне призначення CellID та налаштування радіопараметрів: частоти, що використовується, контролю інтерференцій, потужності сигналу випромінювання та кута нахилу антени.

- **самооптимізація** – автоматичне визначення параметрів базових станцій, які забезпечують найвищу можливу якість обслуговування абонентів (знаходження нових сусідніх станцій та видалення неактивних сусідів, координацію рівнів потужності для зменшення інтерференції між сусідніми базовими станціями, оптимізацію процесів хендоверу і конфігурації QoS).

- **самовідновлення**, яке включає до себе автоматичне визначення відмов у роботі обладнання і відповідну реконфігурацію мережі у разі відмови будь-якої базової станції.

При розподіленні потужностей в стандартах 2G/3G кожна БС має фіксовану зону покриття. Однак абоненти розподілені по території нерівномірно та постійно пересуваються по площі - через це навантаження на БС також розподілено

нерівномірно. Деякі сучасні системи дозволяють частково перерозподілити ресурси за допомогою так званих «дихаючих стільників», але таке рішення також не є досконалим, тому що не враховує режим роботи сусідніх стільників. Тому найбільш вдалим і є рішення на базі концепції мереж, що самоорганізуються.

Завдяки обміну інформацією між сусідніми БС – кожна сусідня станція може узгодити з сусідами потужність випромінювання свого приймально-передавального пристрою, таким чином змінити свою зону покриття. Це дозволить плавно перерозподілити навантаження (потоки) між усіма ділянками мережі (рисунок 1.4). Так само БС передають повідомлення центральному ядру управління, яке може керувати змінами відразу на декількох ділянках й узгодити їх між собою. Саме така гібридна схема взаємодії між базовими станціями і отримала розвиток в концепції LTE-SON.

Тобто в мережі LTE-SON можна виділити декілька головних елементів: головне ядро, яке керує сегментом мережі, базові та мобільні станції. Також важливими параметрами є пропускна спроможність БС та пропускна спроможність каналу, що потребує кожна МС. Змінення потужності БС призводить до перерозподілу МС між БС з урахуванням критерію рівномірності завантаження.

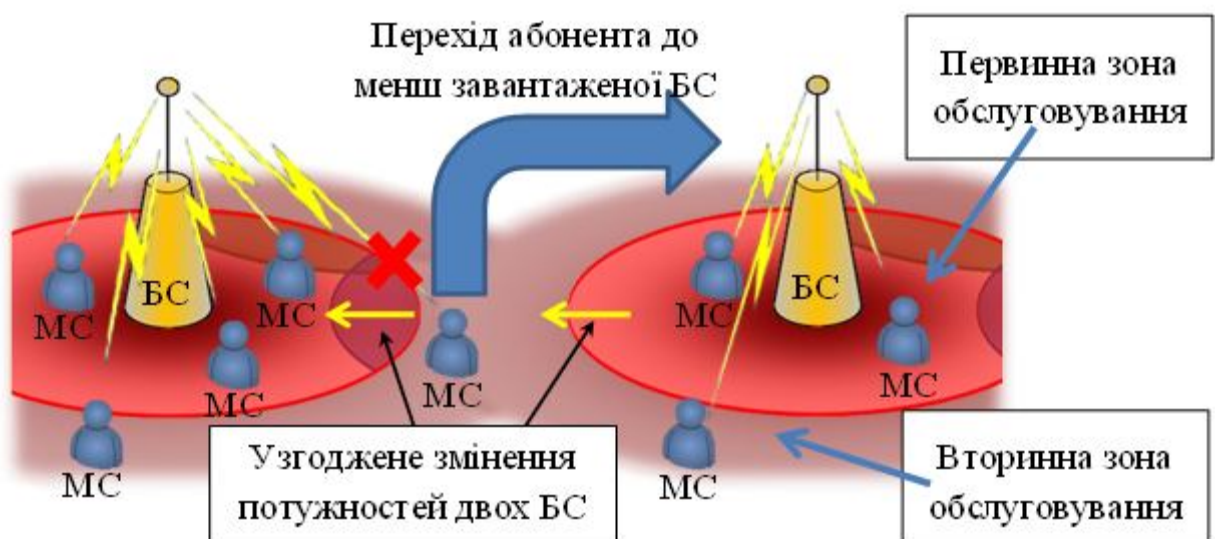


Рисунок 1.4 Перерозподіл абонентів (потоків) між БС в LTE-SON

1.4 Висновки

Проаналізувавши тенденції розвитку послуг зв'язку, треба відзначити, що найбільш актуальною послугою «завтрашнього дня» буде – широкопasmовий доступ до мережі Internet (ШСД). Ця послуга дозволяє реалізувати значний пакет інших послуг, який найбільш ёмко характеризує маркетингова програма Triple Play. Для надання на високому рівні вище означеної послуги потрібно будувати новітні мережі зв'язку, що засновані на новітніх технологіях.

В усіх технологіях можна виділити ряд загальних тенденцій - вони побудовані на базі All-IP мереж, тісно інтегровані з QoS, підтримують високі швидкості передачі (більше 10 Мбіт на одного абонента). Це досягається головним чином за рахунок систем (алгоритмів, методів) перерозподілу потоків. Вони дозволяють надавати послуги різного роду з такою ж якістю, як і в комутованих системах, при цьому не втрачаючи головної переваги пакетних мереж – високої щільності завантаження каналу зв'язку.

Тобто актуальною є задача аналізу, дослідження та розробки систем розподілення потоків. Були розглянуті дві різнопланові системи, де використовують алгоритми балансування. Загальний аналіз таких систем допоміг розібратися у складових елементах систем, принципах організації методів балансування, зібрати матеріал для створення «універсальної» моделі розподільника навантаження (потоків).

У наступній частині робиться акцент на створення зазначеної «універсальної» моделі (програмного ядра) за знайденими даними.

2 РОЗРОБКА УНІВЕРСАЛЬНОЇ МОДЕЛІ

2.1 Розробка аналітичної універсальної моделі для системи з розподільником потоків

Основною особливістю моделі, що розробляється, є її універсальність, тобто можливість досліджувати й аналізувати велику кількість методів розподілу потоків. Тому першочерговим етапом створення програмного пакету - було створення аналітичної моделі, що могла поєднувати в собі рішення для різних систем.

Розгляньмо систему розподілу потоків LTE-SON. Головним ресурсом, що перерозподіляється, є загальна пропускна спроможність кожної БС. Цей ресурс перерозподіляється виходячи з існуючих тарифних планів, вимог до якості обслуговування, місцеположення МС та інше. Приведемо структурну схему, що відображує вище сказане – рисунок 2.1.

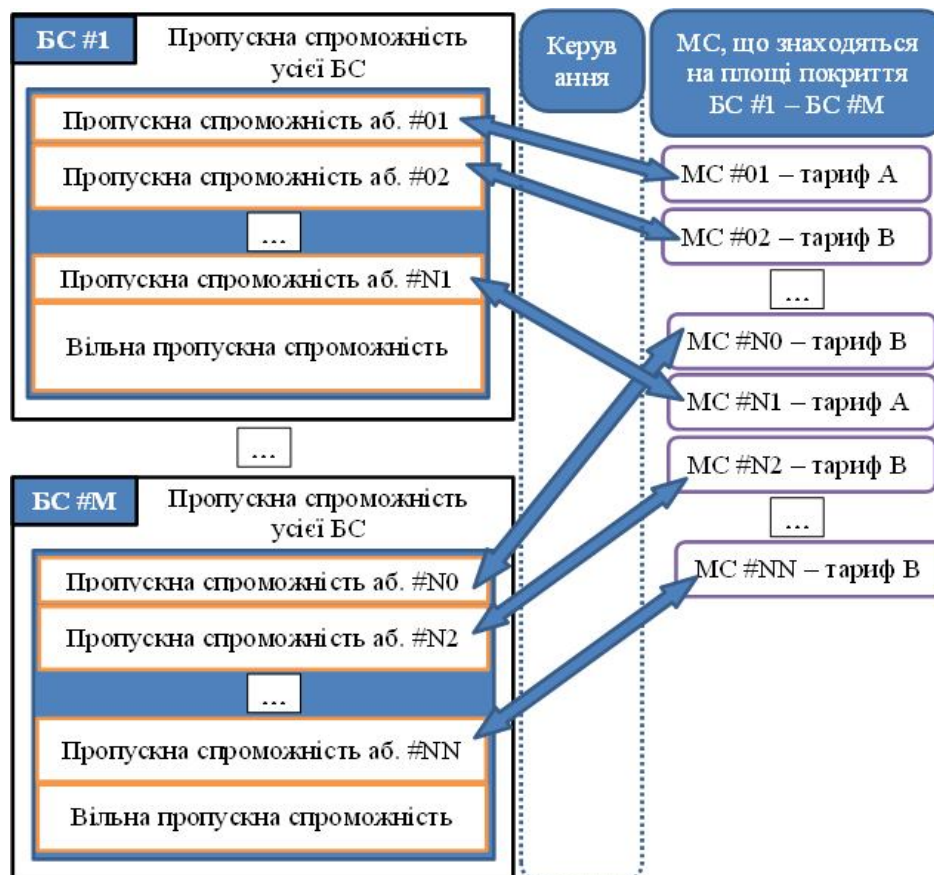


Рисунок 2.1 Структурна схема розподілення потоків в LTE-SON

У ЦОД використовується багато різних алгоритмів розподілення, але відтворимо узагальнену структурну схему. Під час роботи системи перерозподіляються ресурси процесора та пам'яті кожного серверу, що входить до кластеру. Кожен запит, що належить вхідному потоку, має свій тип, ціль, розмір, час обслуговування – завдяки чому несе якусь «вагу» для ресурсів кластера на свою обробку. Крім того, кожен сервер по своїм можливостям (швидкість відповіді, швидкість обслуговування конкретного типу запиту, багатопоточність та інше) є особливою одиницею. Розподільник навантаження у цьому випадку керує потоками, виходячи із вільних ресурсів у кластеру – рисунок 2.2.

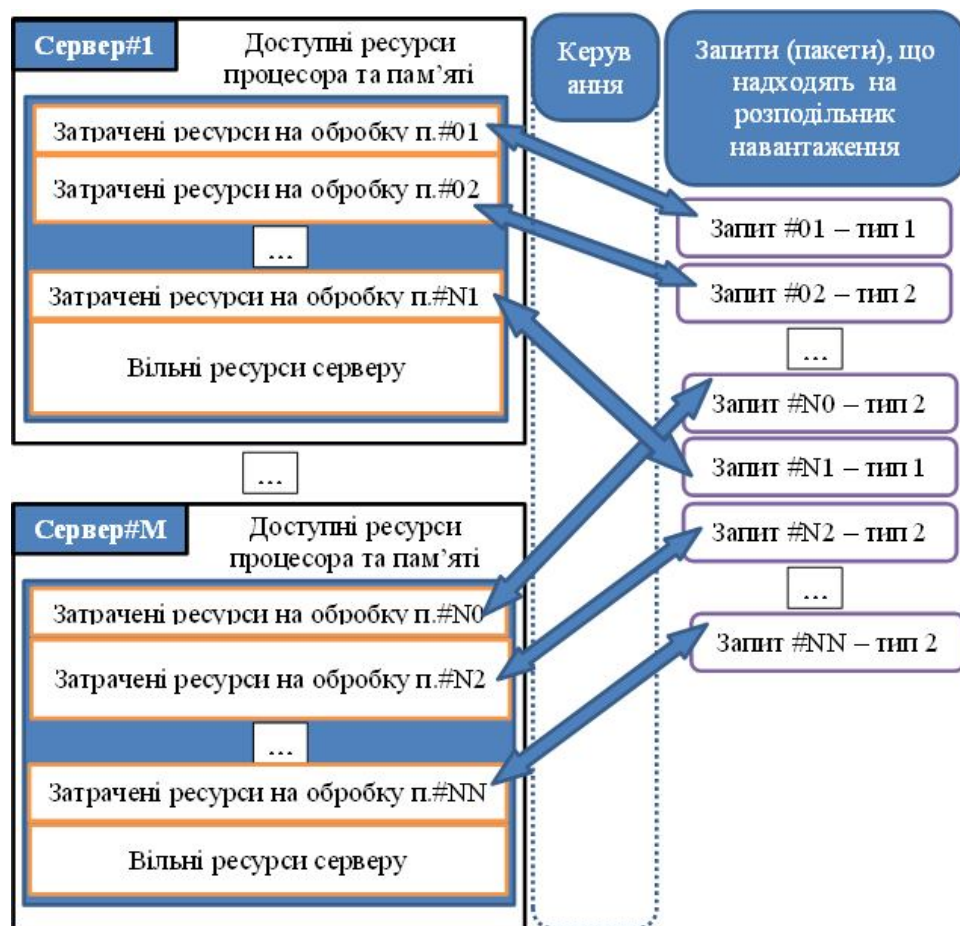


Рисунок 2.2 Структурна схема розподілення потоків в ЦОД

Системи розподілення та призначення цих систем зовсім різні, але структурно процес балансування однаковий для цих систем. Це дає змогу

сформулювати спочатку «універсальну» аналітичну модель, а взагалі створити «універсальну» програмну модель.

Розподільник керується при розподілі вхідним потоком вимог. Тобто головне не розподілити самі елементи, а розподілити максимально ефективно вимоги, що ставить елемент. Вимоги елементів співпадають з наявними ресурсами обладнання, що дозволяє внести абстрактне поняття «ресурс» замість конкретного опису параметрів обладнання та вимог. Виходячи з цього метою розподільника навантаження є керування напрямком потоків ресурсів (тобто керування системою обладнання) згідно з потоком вимог, що надходить. Залишилось відзначити головну функцію розподільника – він не лише організатор роботи системи, а ще й оптимізатор, тобто повинен вирішувати задачі не лише розподілу потоків, а й перерозподілу потоків згідно з критерієм оптимізації.

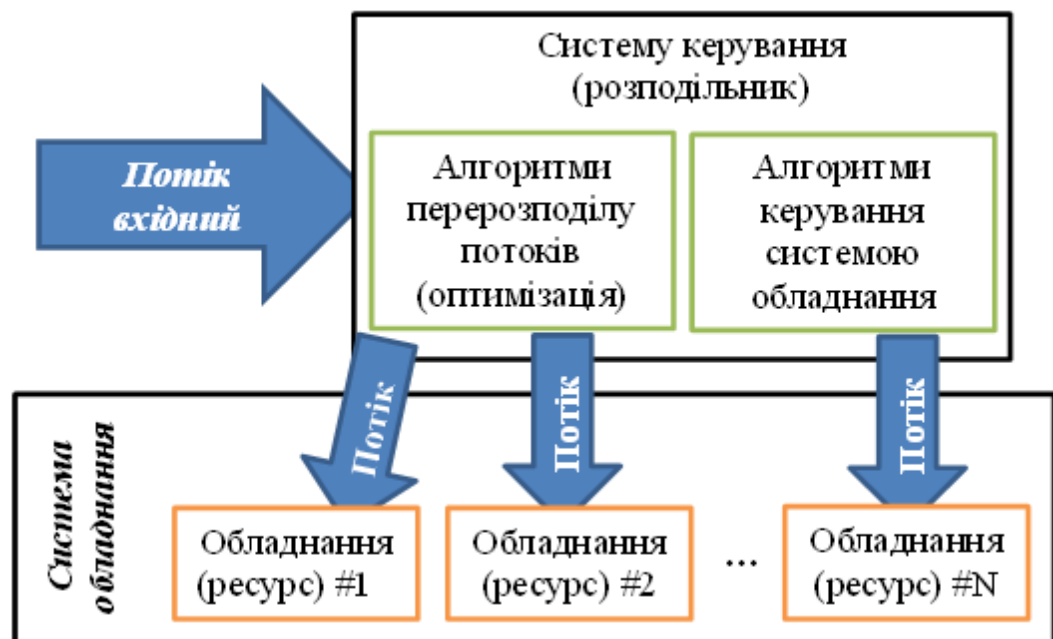


Рисунок 2.3 Аналітична структура «універсальної» моделі

2.2 Розробка програмної універсальної моделі для системи з розподільником потоків

Загальний скелет програмної моделі задає аналітична модель, але окрім того нам будуть потрібні й інші структури, що описані вище. Це обумовлено тим, що програмна модель (framework) повинна не лише описати процеси, що відбуваються, а ще дати гнучкі механізми для моделювання, дослідження та керування самою моделлю.

Найтипівішим елементом в усіх структурах є потік чогось: запитів, пакетів, миттєвих пропускних спроможностей. Тож, найпершим виділено об'єкт – **«Потік»**. Він являє з себе *багатовимірний масив*. Де в стовпці перше значення є деяким *унікальним ключем* (відлік часу для запитів, UID для мобільних станцій). А інші значення – *набір параметрів, що характеризують конкретний елемент* (тип запиту, адреса одержувача, необхідна мінімальна та бажана пропускна спроможність та ін.). Також в цьому об'єкті актуально реалізувати ще ряд методів: визначення параметра Херста, визначення пачечности потоку, визначення його статистичних характеристик та ін.

Для кожного випадку потік потрібно задавати згідно своїх правил. Наприклад, для LTE/SON це буде набір усіх мобільних станцій у сегменті мережі та пропускні спроможності, які вони вимагатимуть, а для WEB-проекту – послідовність запитів та їх тип за певний проміжок часу. Тому необхідно створити об'єкт - **«Генератор початкового потоку»**. Він повинен зчитувати необхідний потік із заданого джерела, або сам його генерувати за наданими принципами.

Так само у всіх структурах є **«Робочий елемент»**. Це власне ті елементи, що вносять у систему поняття обмеженості ресурсів: базові станції, сервер у кластері, смуга пропускної спроможності каналу в IP/MPLS. Цей об'єкт дуже специфічний для кожної задачі. Але його можна описати в глобальному масштабі через такі параметри: *робочий ресурс* (потужність процесора, виділений діапазон частот і потужність випромінювання, максимально можлива пропускна спроможність),

стан (усталені робочі параметри), *зміна стану* (нове завдання, поломка) та *ступінь завантаження* (процесора, частотного ресурсу, каналу зв'язку та ін.). Так як у нас завжди є група **«робочих елементів»**, що працюють спільно, то необхідно створити інтегруючий об'єкт - **«Система»**, який завжди буде описувати роботу елементів у сукупності. Його параметрами будуть: *масив «робочих елементів»*, *стан системи*, *зміна стану системи*, *загальносистемні та середні показники завантаження*.

Останній основний об'єкт - **«Розподільник»** - об'єкт, який здійснює через **«потоки»** взаємодію між об'єктами **«генератор початкового потоку»** та **«система»**, вирішуючи при цьому поставлену задачу розподілу потоків. Його параметрами є цілий ряд об'єктів, що вирішують різні типові задачі. Перелічимо їх нижче.

Найважливішим об'єктом-параметром є об'єкт **«Керування»**, який вирішує задачі розподілу вхідного (вихідного) потоку в **«Розподільник»** між вихідними (вхідними) потоками. Враховуючи, що керування в багатьох алгоритмах не можливо без методів оптимізації – виділяється черговий об'єкт-параметр **«Оптимізація»**. Методи оптимізації можуть накопичувати інформацію минулих рішень, початкових даних й уточнювати прийняті рішення виходячи з цього обсягу інформації. За дану функцію відповідає об'єкт-параметр - **«Пам'ять»**. Можливості застосування прогностичного контролю диктують появу об'єкта-параметра **«Прогноз»**. Всі ці об'єкти описують загальні принципи взаємодії моделі з задачами прогнозу, оптимізації та ін. А реальні методи прогнозу, оптимізації та ін. задаються в успадкованих об'єктах, що дозволяє легко змінювати вживаний метод, не змінюючи структуру всієї моделі.

Дана «універсальна» програмна модель була реалізована програмними засобами динамічної об'єктно-орієнтованої мови програмування Ruby. Загальний вигляд програмної моделі представлений нижче за допомогою UML-діаграми – рисунок 2.4. Слід відзначити, що кольори стрілок не несуть додаткової інформації, а лише полегшують розрізнення зв'язків між різними елементами.

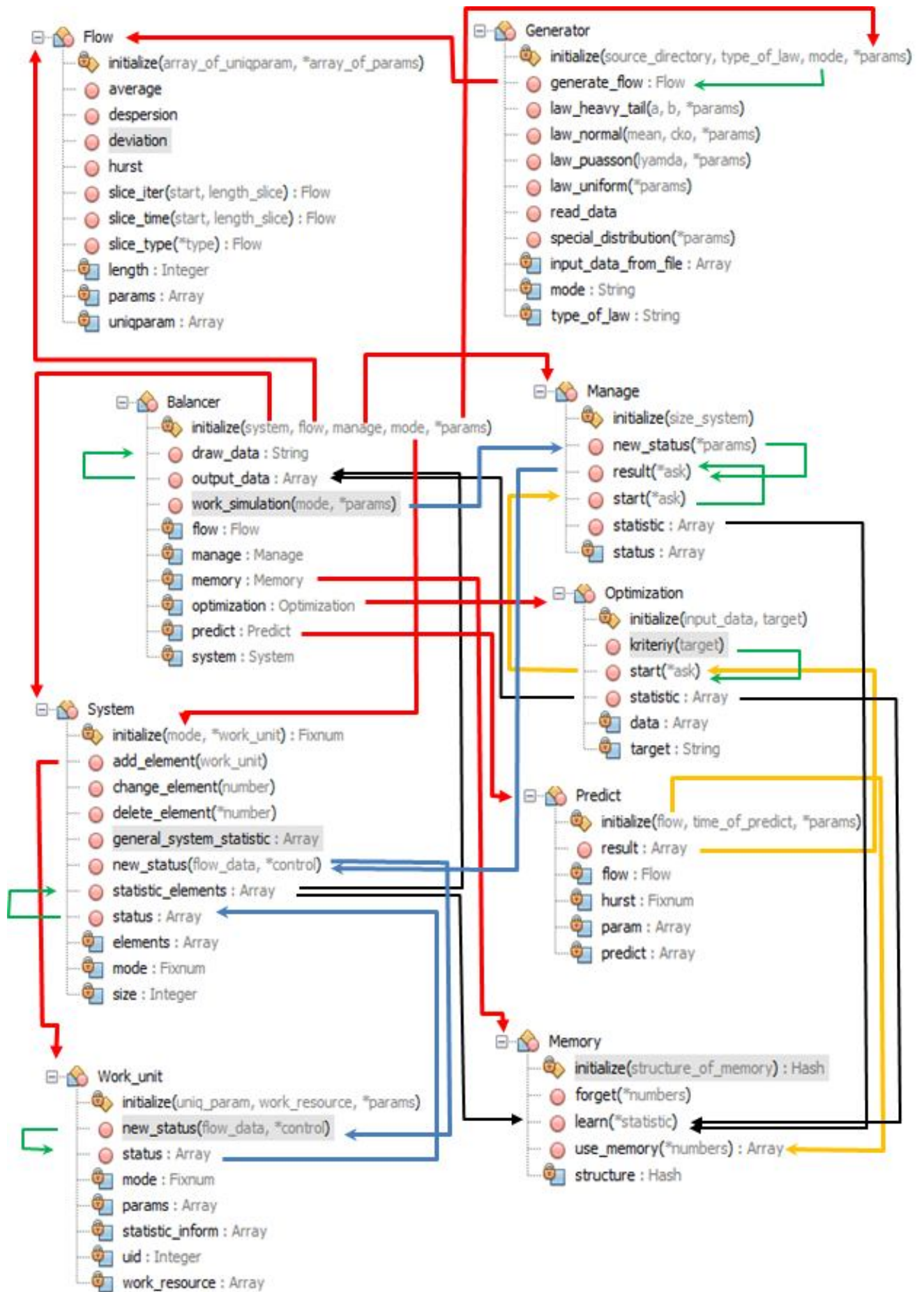


Рисунок 2.4 UML-діаграма універсальної програмної моделі

2.3 Висновки

У даному розділі було проведено структурування даних отриманих у першій частині. На підставі цього було розроблено «універсальну» аналітичну модель. Ця модель, та структурні об'єкти, що були визначені при її розробці, стали фундаментом для програмної моделі.

Була розроблена та реалізована програмна модель (framework) для дослідження задач розподілу потоків. Дана модель дає спільний інструментарій для створення конкретних моделей дослідження, полегшуючи:

- використання однакових алгоритмів та структурних одиниць у різних методах і системах,
- опис взаємодії між елементами,
- акцентування уваги на реалізації математичного апарату, а не програмних реалізацій,
- можливість додавання різного функціоналу, за рахунок написання коротких додаткових пакетів.

У наступній частині буде розглянута адаптація моделі до вже зазначених систем розподілу потоків. На базі цих адаптувань були створені програми для дослідження поставлених нижче задач. Перша задача (статична) – дослідження перерозподілу навантаження мережі LTE-SON у порівнянні з LTE. Друга задача (динамічна) – оцінка ефективності різних алгоритмів балансування навантаження в кластерних мережах.

3 АДАПТАЦІЯ МОДЕЛІ ДО КОНКРЕТНИХ СИСТЕМ БАЛАНСУВАННЯ НАВАНТАЖЕННЯ

3.1 Адаптація моделі до системи перерозподілу потоків в мережі LTE-SON

Мережі, що самоорганізуються, – є дуже складною структурою, яка дозволяє значно зменшити операційні витрати та керування мережею. Задача дослідження можливостей мереж, що самоорганізуються, дуже актуальна, тому що зараз будуються перші комерційні мережі, що засновані на цьому принципі.

Адаптуємо нашу «універсальну» модель до вирішення задач самоорганізації в рамках мережі LTE. В об'єкті «Потік» - ми задаємося набором всіх мобільних станцій (МС), що діють на керованій території. Крім унікального ідентифікатора, кожній МС присвоюються координати її розташування, тип зони в яку вона потрапила (спочатку - 0), пріоритет обслуговування, необхідна пропускна спроможність. У якості «Робочого елемента» задається - базова станція. Крім унікального ідентифікатора, кожній БС присвоюються координати її розташування, діапазон частот, що використовується, частотна ефективність (залежить від кількості антен та застосування технології MIMO). Відповідно об'єкт «Система» трансформується в об'єкт «Мережа», в даному об'єкті реалізована функція «побудова зон» - визначення первинних і вторинних зон покриття. Первина зона визначається як та, у якій співвідношення сигнал/шум «опорної» станції у 5 разів вище будь-якої іншої. У цьому алгоритмі для визначення потужності сигналу базової станції у заданій точці та, відповідно, встановлення меж зон покриття використовується модель розповсюдження радіосигналу COST231[30]:

$$PL = 46.3 + 33.9 \lg f - 13.82 \lg h_b - a h_m + 44.9 - 6.55 \lg h_b \lg d + c_m; \quad (3.1)$$

де d – відстань від БС до абонента,

f – робоча частота,

h_b – висота підйому антени,

c_m та ah_m – коефіцієнти затухання, що залежать від типу місцевості та забудівлі.

Далі адаптується об'єкт «Управління», котрий виходячи з географічних координат МС і зон покриття БС - визначає приналежність кожної МС конкретної БС. Процес оптимізації, який реалізовано у розробленій моделі, відбувається в два етапи: перерозподіл абонентів по базових станціях (який відбувається лише у SON) та можливе зниження швидкості для абонентів з низьким пріоритетом.

При моделюванні будь-якої мережі мобільного зв'язку зазвичай використовуються методи теорії телетрафіка. Але математичний опис процесів, які відбуваються у мережі LTE, побудованій із застосуванням засобів, що реалізують концепцію SON, має свою специфіку. Зокрема, при використанні розподіленого та гібридного SON базові станції мережі самостійно змінюють деякі свої параметри під впливом сигналів від інших станцій. Такі процеси дуже подібні за характером до тих, що відбуваються у нейронних мережах, тобто БС можна розглядати як нейрон. Враховуючи це, доцільно застосувати методи імітаційного моделювання поведінки масових популяцій. У даному випадку ми розглядаємо абонентів як членів такої популяції. Найдоцільнішим є використання методу м'якої оптимізації мурашиними колоніями (ACO)[29]. При цьому модель, заснована на ідеях мімікрії, враховує можливість «маркування шляху» так званим феромоном, який виступає у ролі своєрідного зворотнього зв'язку.

При перерозподілі абонентів по базових станціях враховується, що кожна БС має дві зони обслуговування – основну, у якій потужність сигналу від цієї станції набагато перевищує потужність всіх інших сигналів, та периферійну, у якій абоненти можуть обслуговуватися однією з декількох БС. Для абонентів, які знаходяться у периферійній зоні, алгоритм вибору БС базується на масиві пріоритетів базових станцій, що розраховуються за формулою:

$$P_{ij} = \frac{v_{ij}}{\sum_j v_j} ; \quad (3.2)$$

де P_{ji} – пріоритет j -ї БС для i -го абонента,

τ_{ji} – потужність сигналу від j -ї БС до i -го абонента,

Y_j – завантаженість j -ї БС.

Абонент передається до тієї БС, яка має найвищий пріоритет. Після обчислення пріоритетів всіх БС і, відповідно, призначення кожного абонента до якоїсь БС, обчислюється навантаження, яке створює кожна базова станція на транспортну мережу.

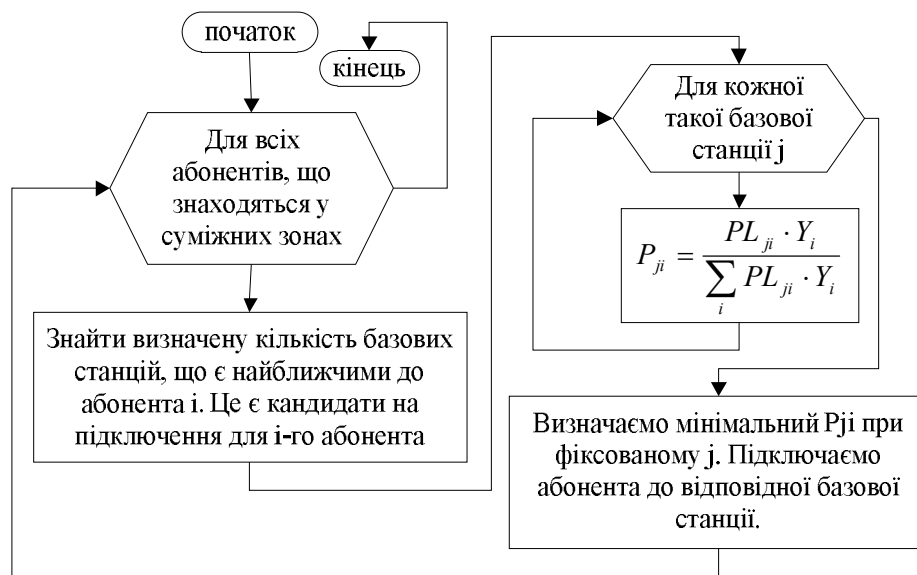


Рисунок 3.1 Алгоритм перерозподілу абонентів по БС (SON)

На базі адаптованої моделі була розроблена програма для дослідження LTE-SON на мові програмування Ruby. Програма має консольний інтерфейс.

```

C:\Ruby187\bin\ruby.exe
distances: [5.92236361026655,101.739029566811,314.58759358602]
disconnected: 190/800 users

After optimization:
Total speed: 2052.0
Overhead in LTE mode: 1860.0
Speeds of the 1 type [min,avg,max]: [1.0,1.0,1.0]
Speeds of the 2 type [min,avg,max]: [0.454545454545455,2.732,5.0]
Speeds of the 3 type [min,avg,max]: [0.0,3.0475,10.0]

distances: [5.92236361026655,112.352494084088,467.892561828155]
disconnected: 12/800 users
users zone type 1: 544
users zone type 2: 256

```

Рисунок 3.2 Текстове виведення результатів роботи програми

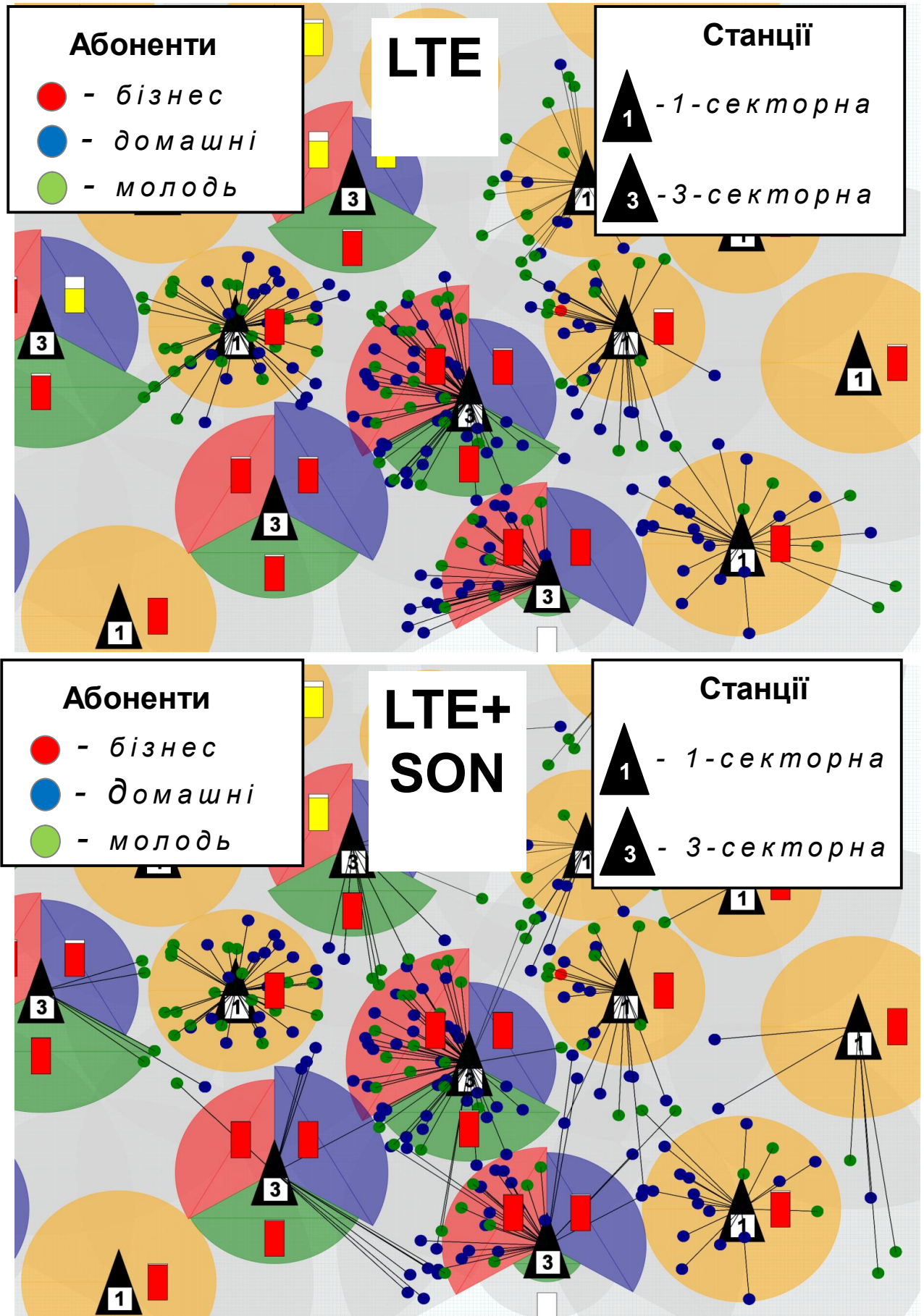


Рисунок 3.3 Графічне виведення результатів роботи програми

Вхідними даними є параметри скупчення абонентів різного типу (закон розподілу), число абонентів, число приймально-передавальних пристроїв (їх координати) і діаграма їх спрямованості, смуга частот, що використовується, частотна ефективність.

Програма моделює два стани системи. Перший – як працює мережа без SON. Другий – як перерозподіляє ресурси SON. Виведення результатів реалізується в текстовому (рисунок 3.2) та графічному виді (рисунок 3.3). Графічний від відображає перерозподіл абонентів завдяки SON, а текстовий дає оцінку підвищення якості обслуговування завдяки SON.

3.2 Застосування універсальної моделі для порівняння різних алгоритмів системи балансування навантаження в ЦОД

Ефективність кластерних мереж обробки даних безпосередньо залежить від якості балансування навантаження розподільником. Велика кількість різноманітних алгоритмів розподілу потоків ускладнює процес обрання найкращого серед них до конкретної ситуації. Застосуємо універсальну модель для порівняння їх між собою.

Опишемо наш **«робочий елемент»** - сервер. Під час обробки запитів найбільше навантаження лягає на процесор сервера, тому як робочий ресурс будемо використовувати потужність процесора. Процесор може одночасно обробляти цілий ряд процесів, а якщо не встигає, то певну частину процесів зберігає в буфері оперативної пам'яті. Уявимо стан нашого сервера, як стек завантаження процесора – процеси, що одночасно обробляються, та ресурси, що витрачаються ними, (процесорний час та потужність), а також запити, які стоять в черзі на обробку. Кожного разу, надсилаючи новий запит, ми перераховуємо стани всіх елементів кластера (дивимося, які ресурси повинні звільнитися до часу цього запиту) і визначаємо новий стан самого кластеру, роль якого грає об'єкт **«Система»**.

Залишається відкритим питання де взяти самі параметри процесора: витрати часу та потужності на обробку одного запиту заданого типу для різних конфігурацій комп'ютерів. Їх потрібно визначати експериментальним шляхом, але ми спростимо нашу задачу – задамо їх самі, таким чином, ми не будемо асоціюватися з конкретною конфігурацією та все одно зможемо проаналізувати загальні тенденції різних простих способів балансування навантаження в WEB-проектах. При цьому звичайно потрібно врахувати випадковий фактор, так як навіть запити одного типу можуть завантажувати комп'ютер по-різному. Для цього ми задамося середніми значеннями обробки одного запиту кожного типу еталонним комп'ютером - h_i , а скільки буде реально потребляти ресурсів кожен наступний запит у j -ого сервера, визначимо за формулою:

$$\tau_{i,j} = \frac{(-1.0 * h_i) * \ln(rand)}{b_j} \quad (3.3)$$

Де $\tau_{i,j}$ – це витрати процесорного часу, витрати потужності процесора, що потребується, у даній роботі обчислюються як $\tau_{i,j}/10$, і перевіряється, щоб воно було менше 1 (так як процесор може бути максимально завантажен на 100%).

$rand$ – псевдовипадкове число.

b_j – множник потужності процесора.

В роботі: $h = [0,05; 0,1; 0,07; 0,08]$, $b = [8; 5; 2; 4; 6]$.

Вхідний потік був узятий з[31]. Запити були розбиті на 4 типи в залежності від свого розміру. Таким чином, в рамках нашої універсальної моделі реалізовані об'єкти: «генератор вихідного потоку» і «система». Залишилося реалізувати об'єкт «Розподільник». Дослідимо тільки прості методи, які не використовують прогноз або складні методи оптимізації, розглянемо 3 успадкованих об'єкта класу «Керування»: алгоритми Round Robbin (DNS-сервер), Multi Class Round Robbin, Weighted Round Robbin.

Алгоритми представлені у виді блок схем на рисунку 3.4. Докладніше розглянемо останній алгоритм, так як він має певну специфіку. Алгоритм працює як Round Robin, тільки замість одного пакета кидає відразу пачку, згідно з установленою ваги для даної машини. Метод визначення коефіцієнтів (ваг) відрізняється в різних способах реалізації даного алгоритму, використана формула[24]:

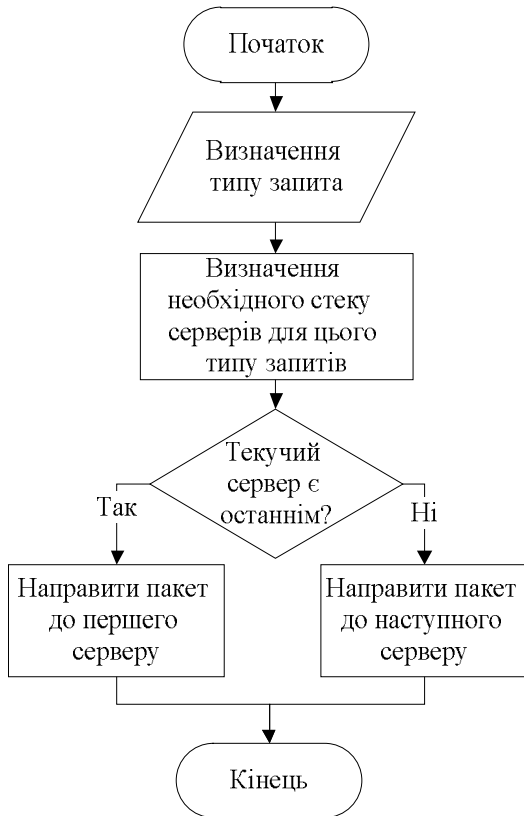
$$c_i = \frac{(1.0 - load_i) * (\max(load) - load_i)}{2} + \frac{\max(queue) - queue_i}{10} \quad (3.4)$$

де $load$ – масив миттєвих значень завантаження кожного сервера

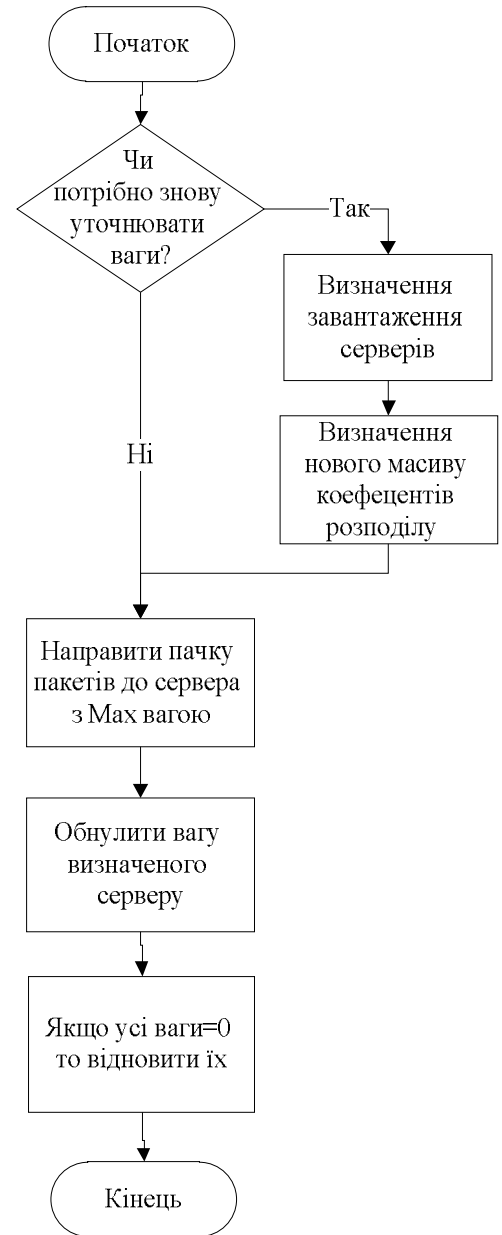
$queue$ – масив кількості запитів у черзі кожного сервера



Round Robbin



Multi Class Round Robbin



Weight Round Robbin

Рисунок 3.4 Алгоритми порівнюваних дисциплін балансування

Результатом адаптації універсальної моделі до цієї конкретної системи стала ще одна дослідницька програма – «Розподільник навантаження». Вона також має консольний інтерфейс, та вхідні дані зазначені вище. Вихідними

даними є текстовий файл, що описує усі параметри кожного стану системи для заданого алгоритму – рисунок 3.5.

| Time | Duration | Class | <1> | <2> | <3> | Pr-<1> | Pr-<2> | Pr-<3> | Q-<1> | Q-<2> | Q-<3> |
|----------|----------|-------|-----|-----|-----|--------|--------|--------|-------|-------|-------|
| 0.000000 | 0.059416 | 0 | + | - | - | 1.49 | 0.00 | 0.00 | 0 | 0 | 0 |
| 0.019919 | 0.498252 | 0 | - | + | - | 1.49 | 3.32 | 0.00 | 0 | 0 | 0 |
| 0.025557 | 0.206365 | 0 | - | - | + | 1.49 | 3.32 | 5.16 | 0 | 0 | 0 |
| 0.032469 | 0.409455 | 0 | + | - | - | 11.72 | 3.32 | 5.16 | 0 | 0 | 0 |
| 0.037691 | 0.992607 | 1 | - | + | - | 11.72 | 9.94 | 5.16 | 0 | 0 | 0 |
| 0.039901 | 0.014764 | 0 | - | - | + | 11.72 | 9.94 | 5.53 | 0 | 0 | 0 |
| 0.045638 | 0.141978 | 0 | + | - | - | 15.27 | 9.94 | 5.53 | 0 | 0 | 0 |
| 0.053581 | 0.330086 | 1 | - | + | - | 15.27 | 12.14 | 5.53 | 0 | 0 | 0 |
| 0.056531 | 0.452420 | 0 | - | - | + | 15.27 | 12.14 | 16.47 | 0 | 0 | 0 |
| 0.057368 | 0.065115 | 0 | + | - | - | 16.90 | 12.14 | 16.47 | 0 | 0 | 0 |
| 0.058109 | 0.184409 | 0 | - | + | - | 16.90 | 13.37 | 16.47 | 0 | 0 | 0 |
| 0.060047 | 0.375335 | 0 | - | - | + | 15.41 | 13.37 | 25.85 | 0 | 0 | 0 |
| 0.060727 | 0.063115 | 0 | + | - | - | 16.99 | 13.37 | 25.85 | 0 | 0 | 0 |
| 0.061313 | 0.047148 | 0 | - | + | - | 16.99 | 13.68 | 25.85 | 0 | 0 | 0 |
| 0.061872 | 0.015710 | 0 | - | - | + | 16.99 | 13.68 | 26.25 | 0 | 0 | 0 |
| 0.068106 | 0.298698 | 2 | + | - | - | 24.46 | 13.68 | 26.25 | 0 | 0 | 0 |
| 0.079941 | 0.058667 | 0 | - | + | - | 24.46 | 14.07 | 25.85 | 0 | 0 | 0 |
| 0.097338 | 0.295861 | 1 | - | - | + | 24.46 | 14.07 | 33.25 | 0 | 0 | 0 |
| 0.099916 | 0.900000 | 0 | + | - | - | 46.96 | 14.07 | 33.25 | 0 | 0 | 0 |
| 0.105307 | 0.900000 | 0 | - | + | - | 46.96 | 20.07 | 33.25 | 0 | 0 | 0 |
| 0.107461 | 0.298717 | 0 | - | - | + | 46.96 | 20.07 | 40.72 | 0 | 0 | 0 |
| 0.119941 | 0.083979 | 0 | + | - | - | 49.06 | 19.76 | 40.72 | 0 | 0 | 0 |
| 0.122226 | 0.026648 | 0 | - | + | - | 49.06 | 19.94 | 40.72 | 0 | 0 | 0 |
| 0.137233 | 0.322159 | 0 | - | - | + | 45.85 | 19.94 | 48.77 | 0 | 0 | 0 |

Рисунок 3.5 Виведення результату роботи дисципліни Round Robin для 3-х серверів

За допомогою програми промодельємо описану вище систему. Результат роботи зобразимо графічно, відображаючи ступінь завантаження кожного серверу за увесь час обробки заданої послідовності. Там же відобразимо середню завантаженість системи в цілому – рисунок 3.6.

На рисунках чим менш відхиляється крива серверу від середнього значення по системі тим краще працює алгоритм балансування. Як бачимо, найбільш гарно працює Weighted Round Robin. Для характеристики якості збалансованості навантаження динамічної системи застосуємо оцінку:

$$k = \sum_1^{N_{\text{запросов}}} \left(\sqrt{\frac{\sum_{j=1}^{N_{\text{серверов}}} (\overline{\text{load}} - \text{load}_j)^2}{N_{\text{серверов}}}} \right) \quad (3.5)$$

Оцінка краще всього показує ефективність зробленого балансування, оскільки враховує як розбіжність щодо середнього значення, так і різкі зриви балансу. Тому є зручною характеристикою для порівняння різних алгоритмів.

Окрім того характеристикою роботи системи є кількість пакетів, що були оброблені кожним сервером, та число втрачених пакетів.

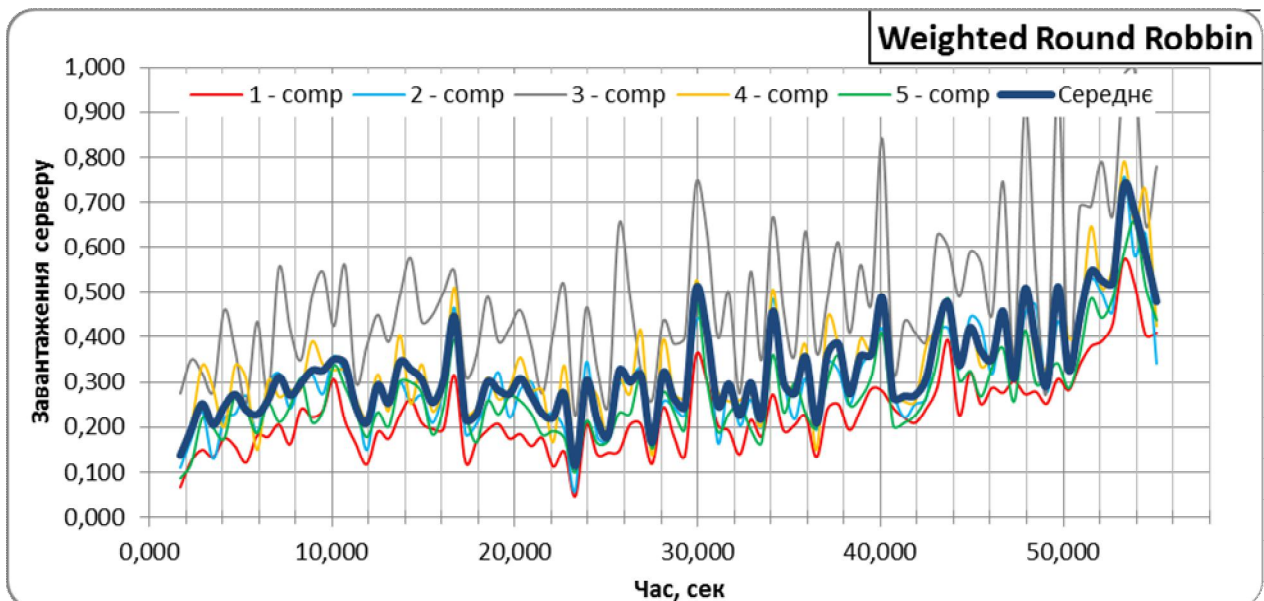
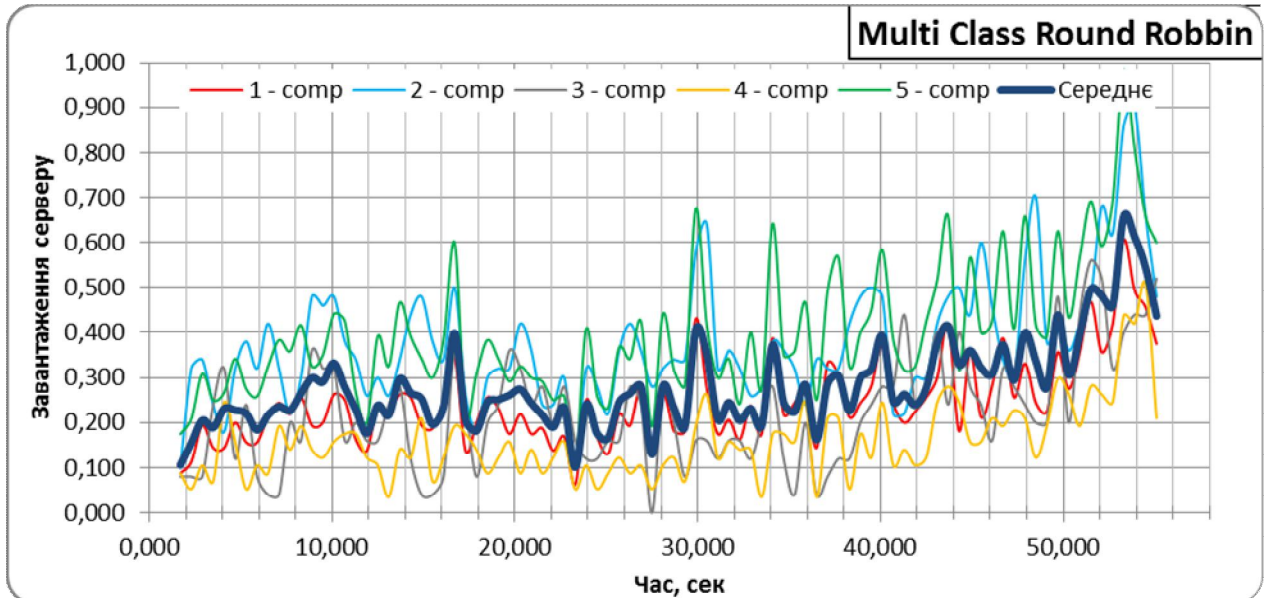
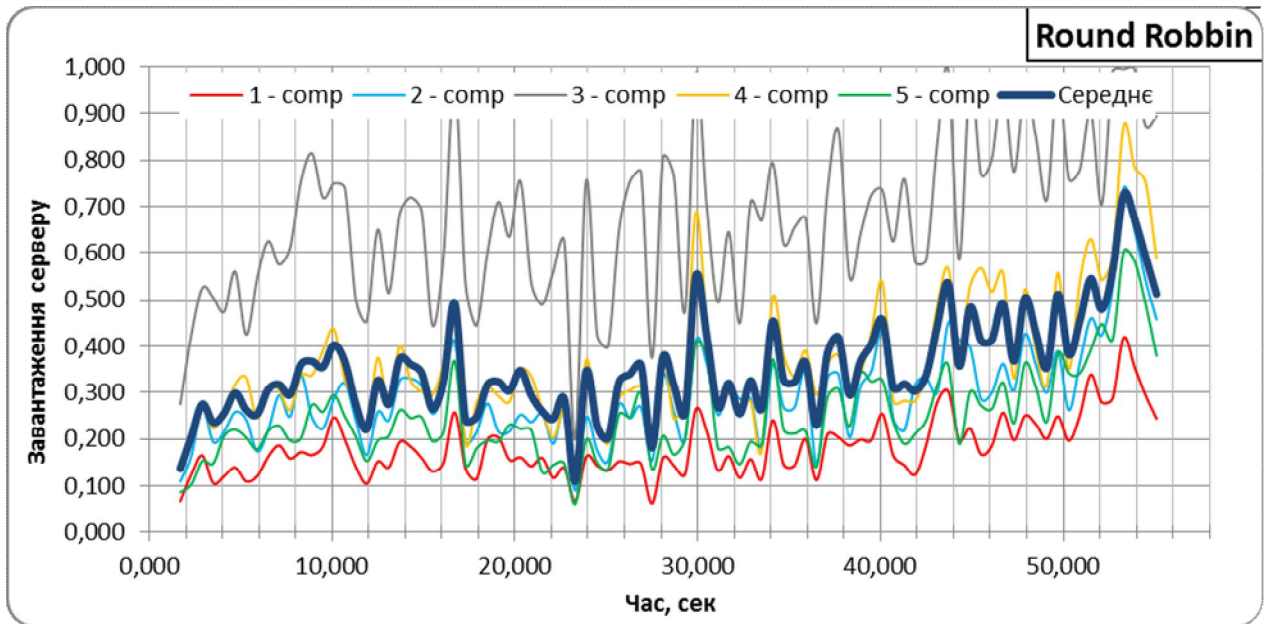


Рисунок 3.6 Порівняння алгоритмів балансування

Виходячи з отриманих даних, можна зазначити, що Round Robin покаже себе ефективно тільки при однакових конфігураціях всіх машин, а також при відсутності аварійних ситуацій (зник зв'язок із одним із серверів). Тому більш прийнятним варіантом є Weighted Round Robin, але він не враховує потенційного навантаження, що надходить до нього, тільки післядію від колишніх подій. З іншого боку MultiClassRoundRobin як би враховує тип навантаження, що надходить, та навіть можливості обробки цих запитів обладнанням, так як не відсилає тяжких запитів до «повільних» машин, а легкими запитами не займає ресурсів «робочої лошаки», але через відсутність знань про завантаження серверів (зворотного зв'язку) – має ті ж самі вади, що і Round Robin.

3.3 Висновки

В результаті адаптації універсальної моделі до конкретних систем балансування навантаження було створено дві дослідницькі програми: «LTE-SON» та «Розподільник навантаження».

Програма «LTE-SON» дозволяє:

- оцінити економічну ефективність використання мереж, що самоорганізуються;
- аналізувати розширення мережі з ростом клієнтської бази;
- досліджувати явища самоорганізації мереж.

За допомогою програми «Розподільник навантаження» можливо:

- порівнювати різні алгоритми балансування;
- виявляти вплив методів прогнозування, аналізу трафіку, методів оптимізації на ефективність роботи системи розподільника;
- промодельовати конкретний стан роботи групи серверів з заданими параметрами.

Крім того було проведено дослідження ефективності трьох алгоритмів балансування навантаження в кластерних мережах. Найкращим виявився алгоритм Weighted Round Robin.

Перелік використаної літератури

1. Переверзев, А.В. Моделирование в электронике: [підручник]/ А.В. Переверзев, В.І. Бойко, А.А. Зорі, В.П. Тарасюк, О.В. Вовна, В.В. Багрій–2-ге вид., доповн. і переробл. – Донецьк: ДВНЗ«ДонНТУ», 2010. –352 с.
2. Sujuan Feng, Eiko Seidel. Self –Organizing Networks (SON) in 3GPP Long-Term Evolution.– Nomor Research GmbH.–2008.–15 p.
3. Воропаева В.Я., Литвинов А. А. Методика прогнозирования параметров трафика телекоммуникационных сетей // Вестник НТУ «ХПИ». Сборник научных трудов. Тематический выпуск «Системный анализ, управление и информационные технологии». – Харьков, НТУ «ХПИ». – 2005. - № 54. – 180 с. С. 142 – 147
4. Marco Dorigo, Thomas Stuzle. Ant Colony Optimization. – Massachusetts nstitute of Technology. – 2004. – 368 p
5. Бессараб В.І., Воропаєва А.О. Особливості оптимізації транспортних мереж мобільного зв'язку третього покоління // Наукові праці Донецького національного технічного університету. Серія: Обчислювальна техніка та автоматизація. Випуск 17 (148). - Донецьк-2009. – 215 с., С. 43-48
6. Воропаєва А.О. Аналітичні моделі для дослідження динамічних процесів в телекомунікаційних мережах // Дванадцята всеукраїнська (сьома міжнародна) студентська наукова конференція з прикладної математики та інформатики СНКПМІ – 2009: Тези доповідей. Львів: ЛНУ – 2009, С. 64-65
7. В.А. Куркчи, Ю.В. Ладыженский. Метод получения оптимального баланса нагрузки для распределенного логического моделирования. Материалы 2-й международной научно-технической конференции "Моделирование и компьютерная графика - 2007". Донецк, 10 - 12 октября 2007 г. Стр. 174-181
8. Соловйов М.С., Воропаєва В.Я. Забезпечення показників якості в конвергентних мережах GSM/Wi-Fi при впровадженні нових сервісів // Наукові праці Донецького національного технічного університету. Серія: Обчислювальна техніка та автоматизація. Випуск 16 (147). - Донецьк-2009. – 248 с., С. 22-28.
9. Охріменко М.Ф., Воропаєва В.Я. Особливості використання алгоритму fair queuing algorithm для забезпечення QoS в домені пакетної комутації мобільних мереж третього покоління // IV міжнародна науково-технічна конференція «Проблеми телекомунікацій», м. Київ, 20 – 23 квітня 2010 р. // Збірник тез. К.: НТУУ «КПІ», 2010. С. 190