МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Харківський національний університет імені В.Н. Каразіна
Факультет математики і інформатики
Кафедра теоретичної та прикладної інформатики

Кваліфікаційна робота

бакалавр

на тему «Моделювання real-time трафіку в комп'ютерних мережах»

Виконав: студент 4 курсу, групи МФ-41
спеціальність 122 «Комп'ютерні науки»
освітньо-професійна програма
«Інформатика»

Зайцев Ю.А.
(прізвище та ініціали)
Керівник Руккас К.М.
(прізвище та ініціали)
Рецензент
(прізвище та ініціали)

Харків – 2023 рік

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

Real-time трафік – трафік реального часу

IoT – Internet of Things («Інтернет Речей»)

QoS – Quality of Service («Якість Обслуговування»)

IP – Internet Protocol

TCP – Transmission Control Protocol

UDP – User Datagram Protocol

ICMP – Internet Control Message Protocol

MPLS – Multiprotocol Label Switching

BGP-TE – Border Gateway Protocol Traffic Engineering

SDN – Software-Defined Network

LSP – Label Switched Path

DTMCs – Discrete-Time Markov Chains

ARMA – Autoregressive Moving Average

ARIMA – Autoregressive Integrated Moving Average

MES – Mean Squared Error

MAE – Mean Absolute Error

RMSE – Root Mean Square Error

ECMP – Equal-Cost MultiPath

IS-IS – Intermediate System to Intermediate System

MQTT – Message Queuing Telemetry Transport

RTP – Real-time Transport Protocol

MTU – Maximum Transmission Unit

ВСТУП

Актуальність теми дослідження. В останні роки питання наявності та доступності якісних способів зв’язку у мережі Інтернет стало гострим, особливо для громадян України: пандемія COVID-19 призвела до вимушеної ізоляції значної кількості людей задля безпеки один одного, багато за яких не мали змоги регулярно спілкуватися за межами своїх домівок, а війна, у свою чергу, розлучила багато сімей, члени яких наразі проживають у різних містах та країнах. Для того, щоб продовжувати спілкування, люди звертаються до сучасних способів Інтернет-зв’язку: популярні месенджери, програми для організації відеоконференцій та дзвінків. Названі вище типи засобів зв’язку об’єднує використання технологій передачі медіа-даних у реальному часі. Як наслідок, у рази зросла потреба у розвитку та розробці таких технологій та програм, і першим етапом їх розробки є проектування, що включає у себе моделювання трафіку системи.

Крім цього, є інші чисельні приклади застосування систем реального часу у комп’ютерних мережах: від фінансових систем та систем промислової безпеки до прямих відео-ефірів та онлайн-ігор. Також системи реального часу у комп’ютерних мережах займають окреме місце у IoT, суть якого полягає в об’єднанні будь-яких пристроїв, що мають сенсори або інші способи зчитувати показники зовнішньої середи, в одну мережу, щоб у реальному часі реагувати на зміни. У системах такого типу (особливо у промислових) зазвичай є три компоненти: це сенсори, що збирають інформацію, система управління, що у реальному часі приймає інформацію та обробляє її, і пристрої контролю, що отримують команди від системи управління та вносять корективи до технологічних процесів [1].

Моделювання мережевого трафіку в режимі реального часу відіграє важливу роль у розумінні поведінки мережі, передбаченні її продуктивності та розробці ефективних механізмів керування мережею. У міру зростання складності комп'ютерних мереж, точний прогноз та моделювання поведінки мережевого трафіку в реальному часі стають важливими для розуміння та контролю мережевих ресурсів. Однією з ключових проблем при моделюванні реального часу мережевого трафіку є відтворення складнощів та варіацій у патернах трафіку, які можуть бути пов'язані з поведінкою користувачів, характеристиками додатків та умовами мережі [2]. Існуюча література досліджувала широкий спектр підходів для вирішення цих проблем, включаючи статистичне моделювання, техніки машинного навчання та симуляції [3].

Метою дослідження було розкрити сутність поняття трафік реального часу та з'ясувати методи його моделювання.

Об'єктом дослідження є процес моделювання трафіку у системах реального часу в комп'ютерних мережах.

Предметом дослідження є сучасні методи моделювання трафіку реального часу.

Для досягнення мети необхідно розв'язати такі **задачі дослідження**:

1. Вивчити та проаналізувати наукову літературу, розкрити зміст понять «трафік у комп'ютерних мережах», «трафік реального часу» та «модель трафіку в комп'ютерних мережах».
2. Дослідити сучасну методологію моделювання трафіку комп'ютерних мереж.
3. Реалізувати методом симуляції модель трафіку реального часу за допомогою програмного забезпечення.

Робота складається зі вступу, двох розділів, загальних висновків, списку використаної літератури. Зміст роботи висвітлено на 44 сторінках основного тексту і містить 15 рисунків та 6 таблиць.

РОЗДІЛ 1. НАУКОВО-ТЕОРЕТИЧНІ ЗАСАДИ МОДЕЛЮВАННЯ ТРАФІКУ У КОМП'ЮТЕРНИХ МЕРЕЖАХ

1.1. Визначення поняття «трафік реального часу» та його характеристики.

Мережевий трафік – це потік даних, що передаються між взаємопов'язаними пристроями в комп'ютерній мережі. Цей обмін даними здійснюється за допомогою різних протоколів зв'язку, які визначають правила та формати обміну інформацією між пристроями [4]. Мережевий трафік може бути створений численними джерелами, такими як додатки користувачів, мережеві служби та фонові процеси [2].

Розуміння характеристик мережевого трафіку є важливим для розробки точних моделей трафіку та ефективного управління ресурсами мережі. Деякі ключові характеристики мережевого трафіку включають:

1. Об'єм трафіку та розподіл: об'єм мережевого трафіку означає кількість даних, переданих по мережі протягом певного періоду часу. Розподіл обсягу трафіку може змінюватися в залежності від факторів, таких як час доби, поведінка користувачів та інфраструктура мережі [3].
2. Шаблони (або патерни) трафіку: мережевий трафік проявляє різноманітні шаблони, від періодичних та передбачуваних до пульсуючих (burst) та аномальних. На ці шаблони можуть впливати характер додатків, умови мережі та поведінка користувачів [2].
3. Типи трафіку: мережевий трафік можна широко класифікувати на різні типи на основі основних протоколів, таких як TCP, UDP та ICMP. Кожен тип трафіку має різні характеристики, які можуть впливати на продуктивність мережі та управління ресурсами.

4. Вимоги щодо якості обслуговування (QoS): різні типи мережевого трафіку мають різні вимоги щодо QoS, такі як пропускна здатність, затримка та втрата пакетів, які потрібно враховувати при управлінні ресурсами мережі [5].

Зрозумівши ці характеристики, дослідники та адміністратори мережі можуть розробляти більш точні моделі трафіку, проектувати ефективну мережеву інфраструктуру та реалізовувати ефективні стратегії управління ресурсами (рис. 1).

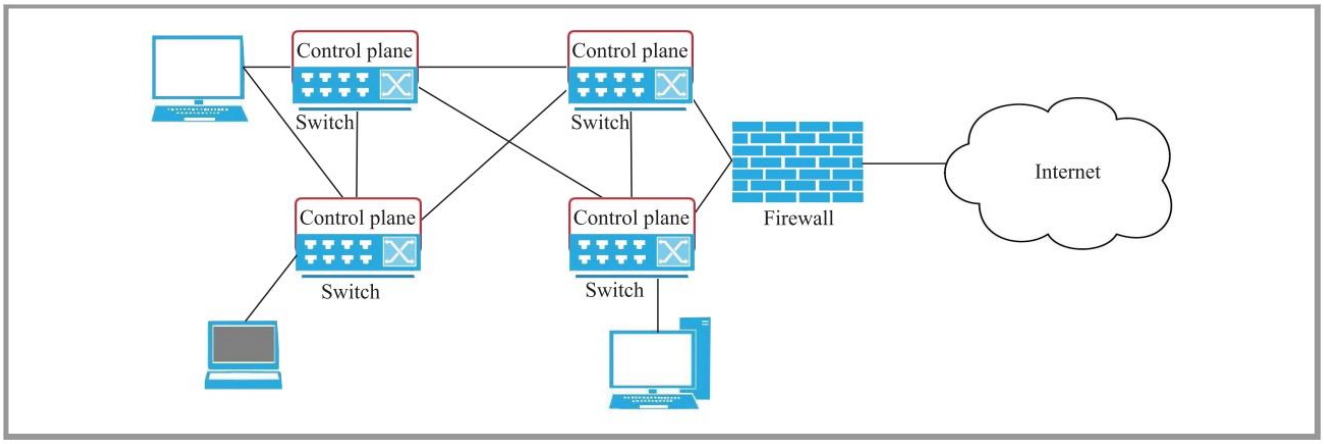

Рисунок 1. Традиційна архітектура мережі [11].

У свою чергу, мережевий трафік реального часу конкретно означає потік даних, який створюють додатки та служби, що вимагають суворих обмежень часу, низької затримки та мінімальної втрати пакетів. Це безпосередньо відповідає такій характеристиці трафіку як QoS.

Таким чином, основними вимогами до трафіку реального часу у комп’ютерних мережах є:

1. Затримка – час, необхідний для того, щоб пакет пройшов шлях від джерела до місця призначення, має не перевищувати певну величину для забезпечення плавного та безперебійного зв’язку.
2. Втрата пакетів – відсоток пакетів, які втрачаються під час передачі має не перевищувати певний відсоток. Для додатків у режимі реального часу втрата пакетів може спричинити затримки, спотворення або навіть повну втрату даних. Тому важливо мінімізувати втрату пакетів, наскільки це можливо.
3. Джитер (або джітер) – коливання затримки між пакетами, велике значення якого може призвести до того, що медіа-дані стануть переривчастими або спотвореними. Тому важливо звести джитер до мінімуму.
4. Пропускна здатність – обсяг даних, який може бути переданий через мережу за певний проміжок часу. Необхідна достатня пропускна здатність, щоб гарантувати, що дані можуть передаватися в режимі реального часу без буферизації або затримок.

Жоден сервіс у реальному часі не може працювати коректно, якщо клієнт не вказує разом із вимогами характеристики очікуваного вхідного трафіку. Опис вхідного трафіку та всіх різних вимог вимагає багато роботи з боку клієнта – збір необхідної інформації та її введення можуть зайняти дуже багато часу. Добре спроектована послуга зв’язку в реальному часі мінімізує зусилля, які має витратити клієнт.

Залежно від застосування, клієнти можуть вказати свої вимоги до затримки різними способами. Звичайно, затримки, з якими має справу клієнт, будуть пов'язані з повідомленнями, спрямованими на додатки; наприклад, затримка між початком передачі кадру відео, файлу або невідкладного дейтаграми на рівні клієнта і закінченням прийому цього ж кадру, файлу або невідкладного дейтаграми на рівні клієнта. У випадках (наприклад, у деяких розподілених системах реального часу), де замість затримок повідомлень призначаються дедлайни повідомлень, ми завжди можемо обчислити затримку на основі знання про дедлайни і часи надсилання, таким чином зводячи задачу до обмеження затримки (рис. 2) [6].

Додаток		Чутливість				
		Втрата	Затримка	Джитер	Пропускна здатність	Безпека
Трафік даних	Електронна пошта	Висока	Низька	Низька	Низька	Низька
	Конфіденційна ел. пошта	Висока	Низька	Низька	Низька	Висока
	Передача файлів	Висока	Низька	Низька	Низька, Середня, Висока	Середня
	Переказ грошей	Висока	Низька	Низька	Низька	Висока
Трафік реального часу	Аудіо за вимогою (AOD)	Низька	Низька	Висока	Середня	Низька
	Відео за вимогою (VOD)	Низька	Низька	Висока	Висока	Низька
	Телефонія	Низька	Висока	Висока	Низька	Низька
	Відеоконференції	Низька	Висока	Висока	Висока	Низька
	Конфіденційні відеоконференції	Низька	Висока	Висока	Висока	Висока

Рисунок 2. Приклад поширених додатків та їх чутливості до певних характеристик QoS [19].

Важливим є питання пріоритизації трафіку різних типів. На транспортному рівні моделі OSI основними протоколами є TCP, UDP та SCTP (останній нечасто використовується у відкритому інтернеті).

Зазвичай TCP використовується для додатків, де надійність є пріоритетом, а певний рівень затримки може бути прийнятний. Це

пояснюється тим, що TCP використовує механізми, такі як перевірка на помилки та підтвердження для надійної передачі даних. Приклади таких додатків – це веб-перегляд, електронна пошта, передача файлів тощо. Але існують і «легкі» протоколи обміну повідомленнями, що можуть базуватися на стеку TCP/IP, такі як MQTT, AMQP, WebSocket.

З іншого боку, UDP часто використовується для додатків у реальному часі, де важливий низький рівень затримки, і можна терпіти випадкові втрати пакетів. Приклади включають VoIP (голос по IP), онлайн ігри та потокове відео. У таких випадках може бути розумно надавати пріоритет трафіку UDP, щоб забезпечити належне функціонування цих додатків, чутливих до затримки.

Однак це не означає, що трафік UDP завжди повинен мати пріоритет над TCP. У сценарії, коли мережу в основному використовують для додатків на основі TCP, таких як веб-перегляд та електронна пошта, пріоритет TCP може бути більш корисним. Важливо також враховувати, що механізми керування заторами TCP можуть змусити трафік TCP поступатися іншому трафіку під час періодів високого мережевого навантаження. Занадто велика кількість трафіку UDP, який не має цих механізмів керування заторами, може призвести до зниження продуктивності TCP [7].

До інших вимог, що забезпечують якість зв’язку, входять послідовність, відсутність дублікатів, відновлення після відмови та час налаштування сервісу. Вимоги в цих галузях, як правило, також мають помітний вплив на продуктивність.

У випадку програм, що включають потоки повідомлень (а не окремі дейтаграми), може бути необхідним або бажаним, щоб повідомлення доставлялися у послідовності, навіть якщо послідовність не є повною. Якщо послідовність не забезпечується всіма серверами на всіх рівнях, додаток може

толерувати неправильну послідовність повідомлень, якщо їх кількість невелика, або якщо обмеження затримки настільки незначне, що дуже мало неправильних повідомлень потрібно відкидати через те, що вони спізнилися. Більшість тезисів про послідовність так само стосується дублювання повідомлень. Однак видалення дублікатів є простішим і швидшим, ніж корекція послідовності, за умови, що якийсь рівень відстежує номери послідовності вже отриманих повідомлень. Вказівка на обмеження може бути необхідною лише в разі дуже частого дублювання, але це було б ознакою серйозної несправності мережі і не повинно вирішуватися так само, як вирішуються затримки чи втрата повідомлень. Зазначені спостереження, звичайно, не стосуються випадку навмисного дублювання для більшої надійності.

У системі реального часу слід вказати, що станеться у разі відмови сервера. Ідеальною з боку зору клієнта буде така ситуація, коли відмови повністю маскуються, а обслуговування є повністю відмовостійким. Різні застосунки мають різні вимоги до відновлення після відмови. В разі невідкладних дейтаграм або потоків невідкладних повідомлень у більшості розподілених систем реального часу, ймовірно, немає великої користі від відновлення, якщо це не може бути здійснено настільки швидко, щоб виконати жорсткі терміни, принаймні у деяких випадках. У випадку передачі відео або аудіо, своєчасне відновлення обслуговування, як правило, є дуже корисним або навіть необхідним; тому клієнтам можуть знадобитися гарантії щодо верхніх меж середнього або максимального часу відновлення; це також може стосуватися інших застосунків, в яких терміни не є настільки значимими, або де основний акцент зроблений на пропускну здатність і/або надійність, а не на затримку [6].

1.2. Методи забезпечення якісного трафіку реального часу в комп'ютерних мережах

Одними з методів забезпечення QoS є пріоритизація пакетів, формування трафіку (з англ. traffic shaping) та впровадження політики трафіку (з англ. traffic policing).

Суть пріоритизації пакетів полягає у наданні одним типам трафіку більш високого пріоритету, на основі їх терміновості та важливості. Для цього пакетам задають пріоритет, зазвичай у вигляді поля в заголовку пакета, який потім використовується такими пристроями як маршрутизатори та комутатори. Це сприяє достатній пропускній здатності, яка необхідна для систем реального часу. Цей механізм може бути запроваджений на різних рівнях мережи, наприклад на мережевому, транспортному чи прикладному, для того, щоб впевнитись, що на трафік реального часу не впливає трафік іншого типу, що не так сильно покладається на часові обмеження [8].

Формування трафіку полягає у обмежуванні або пріоритезації трафіку, що посилається або приймається, залежно від його типу або джерела. У цьому методі частина пакетів затримуються щоб згладити сплески трафіку та забезпечити рівномірне навантаження. Зазвичай реалізується цей метод алгоритмом «token bucket», при використанні якого кожен пакет маркується токеном, що відображає певну кількість даних, і пакет може бути відправлений тільки тоді, коли достатньо вільних токенів. Як і при використанні пріоритизації пакетів, покращується пропускна здатність трафіку реального часу за рахунок обмеження її для інших типів трафіку [9]. Формування трафіку передбачає наявність черги та достатнього місця для буферизації пакетів.

Впровадження політики трафіку має схожу мету: стабілізувати трафік, уникаючи перевищення його встановленої норми на одиницю часу. Але реалізовується цей метод шляхом «викидання» або повторного маркування надлишкових пакетів.

Таким чином, обидва методи служать одній меті, але призводять до різних результатів (рис. 3) [10].

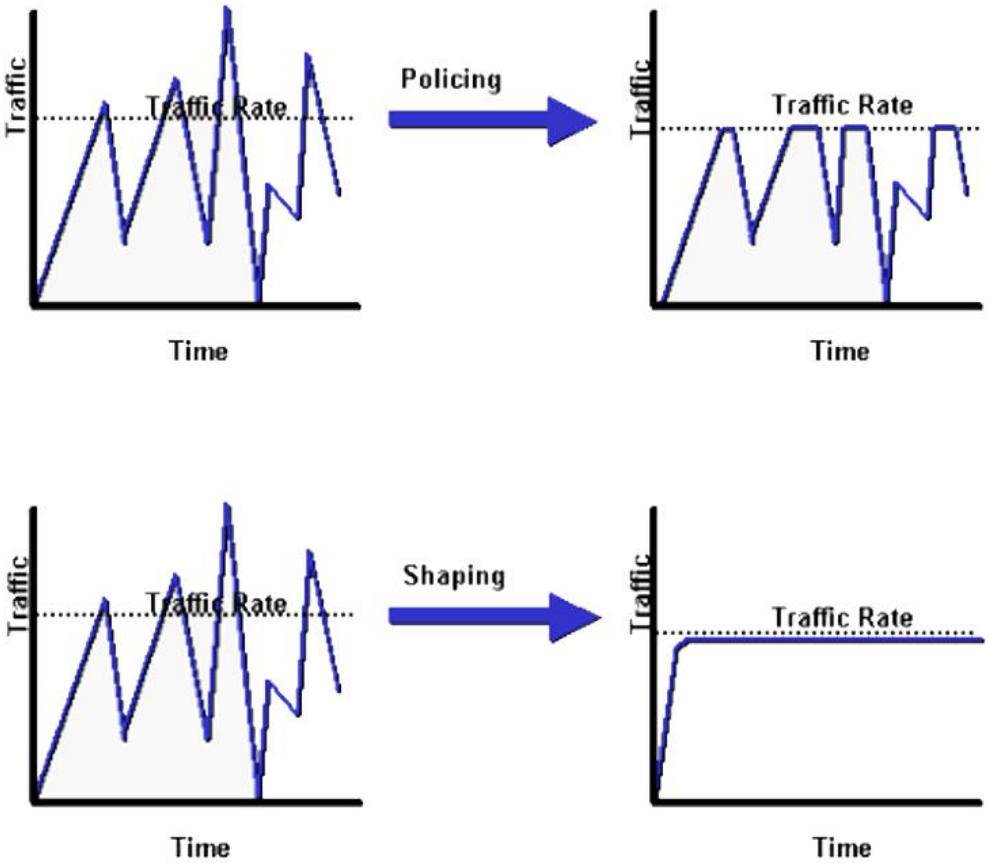

Рисунок 3. Зображення трафіку при впровадження політики трафіку (зверху) та при використанні формування трафіку (знизу) [10].

Вищезгадані методи можуть охоплювати такі прийоми контролю трафіку як: фільтрація пакетів, класифікація трафіку, резервація ресурсів, контроль перенавантаження (англ. congestion control).

На рівні топології мережі може застосовуватися техніка вибору шляху (з англ. path selection). Ця техніка базується на виборі найкращого шляху для трафіку реального часу, враховуючи доступність, провідну здатність та надійність можливих маршрутів. Метою цього методу є забезпечення QoS, зводячи до мінімуму вплив на трафік інших типів. Вибір шляху може бути реалізований за допомогою таких технологій як: MPLS, BGP-TE, SDN.

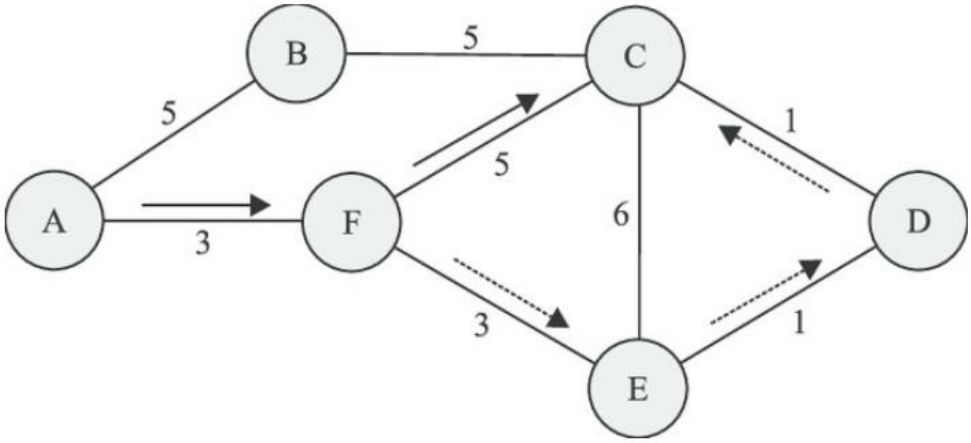

Рисунок 4. Приклад ECMP: існує два маршрути с однаковою вартістю до вузлу призначення C, (A, F, C) та (A, F, E, D, C) [11].

ECMP – це техніка маршрутизації, яка використовується в комп'ютерних мережах для розподілу трафіку по кількох маршрутах з однаковою вартістю. Це часто застосовується в протоколах маршрутизації, таких як OSPF і IS-IS, для досягнення балансування навантаження і покращення продуктивності мережі.

Основним принципом ECMP є рівномірне розподілення трафіку по кільком маршрутам з однаковою вартістю. Коли маршрутизатор отримує пакет, адресований до певного пункту призначення, він переглядає свою

таблицю маршрутизації і визначає кілька варіантів наступного кроку з однаковою метрикою вартості. Потім маршрутизатор використовує хеш-алгоритм для визначення відповідного вихідного інтерфейсу для пакета на основі початкової та кінцевої адреси, номерів портів або інших полів пакета. Це забезпечує постійну маршрутизацію пакетів для одного потоку по одному й тому ж шляху (рис. 4).

Варто зазначити, що ECMP має свої особливості і обмеження. Наприклад, без додаткових механізмів ECMP не гарантує оптимального балансування навантаження, якщо доступні шляхи мають різні характеристики або містять ланки з різною пропускною здатністю. Крім того, асиметричні шляхи можуть призвести до доставки пакетів не в порядку, що може вплинути на деякі програми або протоколи, які потребують строгого збереження послідовності.

MPLS широко використовується в мережах реального часу. Ця технологія дозволяє операторам мережі задавати LSP (маршрути, якими будуть подорожувати пакети) з певними характеристиками, такими як пропускна здатність, затримка та рівень втрати пакетів.

BGP-TE використовує такі атрибути як Autonomous System Path (AS_PATH) та Multi-Exit Discriminator (MED) для впливу на вибір та обирає багато шляхів між джерелом та місцем призначення, щоб у результаті получити більш рівномірне навантаження та зробити доступним аварійне переключання.

Технологія SDN («Програмно-визначена мережа») з’явилася відносно недавно та походить з аналогічної системи у телефонній мережі загального користування (PSTN). Підхід SDN пропонує гнучкий підхід до вибору шляху, відокремлюючи рівень керування від рівня даних, що дозволяє операторам мережі динамічно контролювати поведінку мережі та її топологію за

допомогою централізованого контролера. За допомогою SDN керування трафіком стає більш гнучким і динамічним. Трафік можна централізовано визначати та контролювати на основі певних критеріїв, таких як типи додатків, вимоги до якості обслуговування (QoS) або врахування вимог безпеки. Трафік можна направляти по конкретним шляхам або піддавати спеціальній обробці в реальному часі (рис. 5).

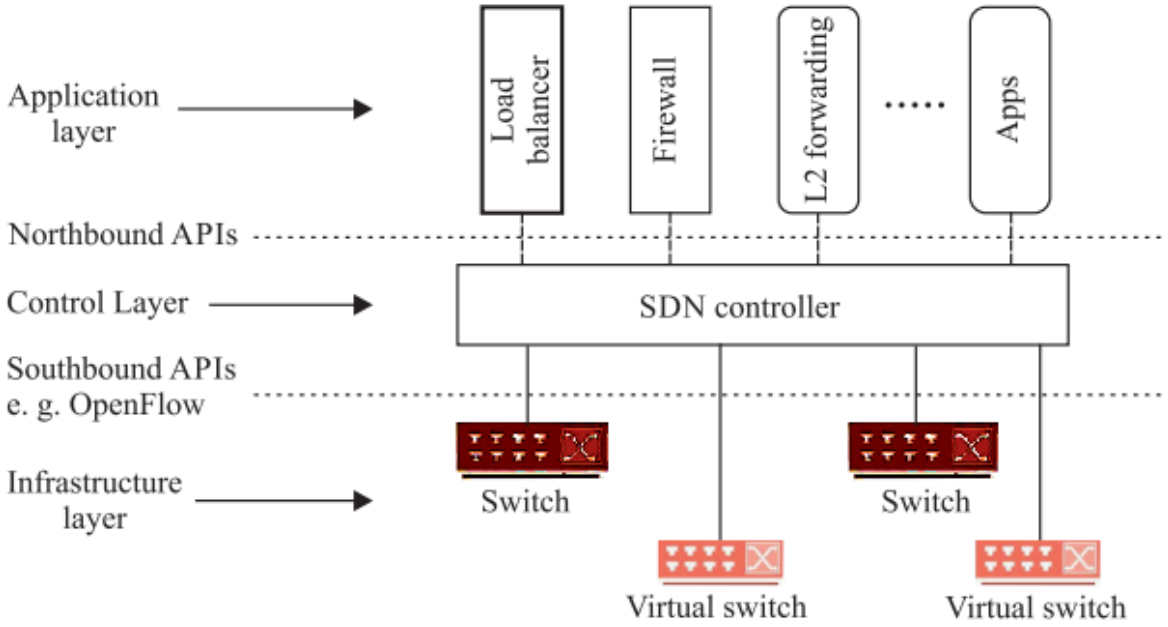

Рисунок 5. Приклад архітектури SDN, що показує розподіл системи на шари [11].

SDN також сприяє балансуванню навантаження в мережі, розподіляючи трафік розумно між кількома шляхами або ресурсами мережі. Централізовані контролери можуть динамічно контролювати навантаження, рівень перенавантаження та використання ланок для прийняття обґрунтованих рішень щодо балансування навантаження. Це забезпечує оптимальне використання мережевих ресурсів та уникнення точок перенавантаження.

У стезі QoS, SDN надає розширені можливості, дозволяючи адміністраторам визначати та контролювати політики для задоволення вимог конкретних програм. Трафік можна класифікувати, встановлювати йому пріоритет та обробляти по-різному залежно від параметрів QoS, таких як затримка, втрати пакетів та пропускна здатність. Контролери SDN можуть динамічно адаптувати політики QoS у відповідь на змінні умови мережі. Крім цього, SDN сприяє поліпшенню моніторингу та аналізу трафіку: оскільки план керування централізований, він надає глобальний огляд мережі, що полегшує збір статистики трафіку, відстеження продуктивності та виявлення аномалій або загроз безпеці. Інструменти моніторингу трафіку можуть бути інтегровані з контролером SDN для забезпечення аналізу трафіку в реальному часі та прийняття рішень [11].

1.3. Огляд типів методів моделювання трафіку в комп’ютерних мережах.

Методи моделювання трафіку за типом класифікують на аналітичні, статистичні, симуляційні, моделювання на основі сліду (англ. trace-based modeling) та гібридні. В кожного методу є свої переваги при моделюванні трафіку мережі певного масштабу, конфігурації та устрою.

Важливу роль у моделювання трафіку грають моделі Маркова. Марковські методи – це математичні техніки, які використовуються для моделювання та аналізу систем з властивістю Маркова. Властивість Маркова стверджує, що ймовірність переходу до певного стану залежить тільки від поточного стану і не залежить від минулої історії системи. У контексті моделювання мережевого трафіку в реальному часі марковські методи застосовуються для відтворення ймовірних взаємозв’язків та залежностей між

різними подіями мережі. Ці методи можуть надати цінні уявлення про патерни трафіку, метрики продуктивності та поведінку системи [3].

Марковські ланцюги – це стохастичні процеси, що включають послідовність випадкових змінних, де перехід з одного стану в інший визначається ймовірностями. Ці ймовірності часто представляються у вигляді матриці переходів, яка вказує ймовірності переходу з одного стану в інший за один крок. Саме властивість Маркова дозволяє марковським ланцюгам не запам’ятовувати події минулого і спрощує їх аналіз. Ланцюг Маркова, що приймає кінцеву кількість станів, можна задати так:

$$Pr(X_n+1=x \mid X_n=x_n)$$

Тобто ймовірність того, що наступний стан (n+1) випадкової змінної X є певною величиною, x, що залежить від поточного стану випадкової величини, $X_n = x_n$ (рис. 6).

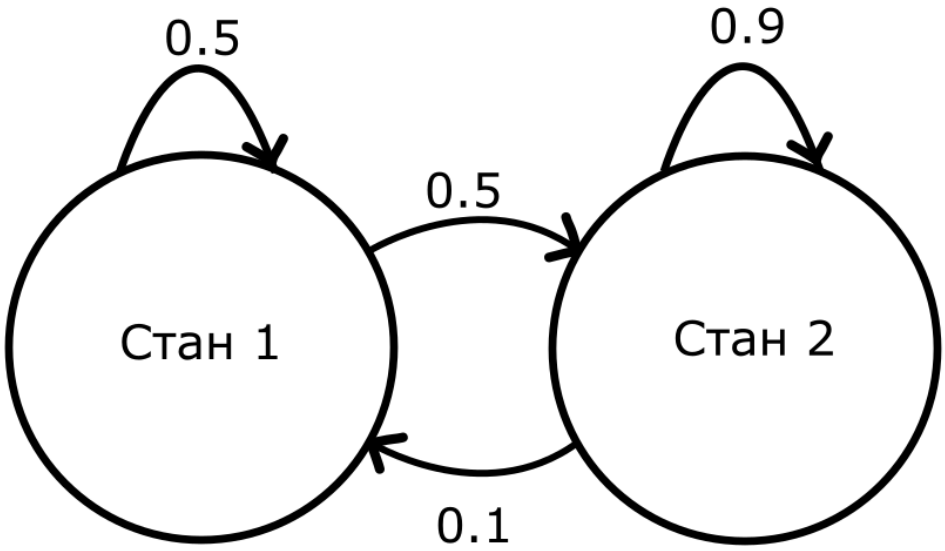

Рисунок 6. Граф переходів ланцюга Маркова з двома станами («Стан 1» та «Стан 2»).

Одним з поширених типів ланцюга Маркова, який використовується у моделюванні мережі, є процес народження-смерті. У процесі народження-смерті система може перейти від стану «i» до стану «i+1» («народження») або до стану «i-1» («смерть»), або вона може залишитися в тому ж стані. Це можна використовувати для моделювання, наприклад, кількості пакетів у черзі: прибуття нового пакету – це «народження», а обслуговування пакету та його видалення з черги – це «смерть».

Марковські ланцюги можуть мати різні властивості, залежно від їх станів та ймовірностей переходів. Деякі загальні властивості включають неподільність (кожен стан може бути досяжним з будь-якого іншого стану), аперіодичність (відсутність фіксованого шаблону для часу повернення до стану) та ергодичність (ланцюг нарешті досягає стаціонарного розподілу).

Марковські ланцюги знаходять застосування в різних галузях. У контексті моделювання мережевого трафіку в реальному часі марковські ланцюги можуть представляти, наприклад, кількість пакетів у черзі або стан (зайнятий або вільний) сервера, тож можуть бути використані для аналізу та прогнозування патернів трафіку, оцінки метрик продуктивності системи, оцінки протоколів маршрутизації та моделювання поведінки мережі в різних сценаріях [12].

Дискретні часові марковські ланцюги (DTMCs) є типом марковських ланцюгів, де час розбивається на дискретні інтервали, і спостереження та переходи відбуваються в ці точки часу. DTMCs широко використовуються для моделювання та аналізу різних стохастичних процесів, включаючи моделювання мережевого трафіку в реальному часі.

У DTMC система переходить з одного стану в інший за дискретний проміжок часу, і ймовірності переходу між станами залишаються постійними протягом цих проміжків часу. Ймовірність переходу до певного стану

наступного кроку залежить тільки від поточного стану і не залежить від минулої історії системи. Ця властивість «безпам'ятності» робить DTMCs актуальними для моделювання динамічних систем, де майбутня поведінка залежить лише від поточного стану.

У контексті моделювання мережевого трафіку в реальному часі, DTMCs можуть відображати ймовірнісні залежності та зв'язки між різними подіями та станами мережі. Шляхом побудови моделі DTMC на основі спостережень даних мережевого трафіку, дослідники можуть отримати уявлення про патерни трафіку, оцінити показники продуктивності, такі як втрати пакетів або затримки, оцінити протоколи мережі та моделювати поведінку мережі в різних сценаріях [3].

Ще однією причиною використання марковських моделей є їх здатність вказувати на аномалії. Марковські методи можуть бути ефективними у виявленні аномалій, оскільки вони враховують ймовірнісні залежності та патерни в спостережуваних даних.

Один з підходів до виявлення аномалій за допомогою марковських методів – це навчання марковської моделі на нормальній або очікуваній поведінці даних. Марковська модель вивчає ймовірності переходу між різними станами на основі навчальних даних. Після навчання модель може бути використана для передбачення найбільш ймовірної послідовності станів або подій на основі нових спостережень. Якщо спостережена послідовність значно відрізняється від очікуваної послідовності, передбаченої моделлю, це вказує на наявність аномалії.

Існують різні типи марковських моделей, які можна використовувати для виявлення аномалій. Широко використовуються «Приховані Моделі Маркова» (ПММ), де базові стани, що генерують спостережувані дані, є невидимими, і доступні тільки спостережувані дані для аналізу. Аномалії можуть бути

виявлені шляхом порівняння ймовірності спостережуваної послідовності з ймовірністю очікуваної послідовності на основі ПММ.

Інший підхід – використання Марковських Ланцюгів або Марковських Процесів Прийняття Рішень (MDP) для моделювання поведінки системи або процесу. Аналізуючи переходи між станами та їх ймовірності, можна виявляти аномалії, коли спостережені переходи суттєво відрізняються від очікуваної поведінки.

Перевагою використання марковських методів для виявлення аномалій є їх здатність враховувати часові залежності та контекст у даних. Вони можуть обробляти послідовні дані, такі як часові ряди або послідовності подій, та надавати уявлення про динаміку та еволюцію системи [13].

Ще одним статистичним методом моделювання є аналіз часових рядів (Time Series Analysis), який використовується для аналізу та моделювання даних, що збираються послідовно у часі. Він спрямований на розуміння закономірностей, тенденцій та залежностей, присутніх у впорядкованих за часом даних, для здійснення прогнозів та передбачень майбутніх значень.

Основними аспектами, пов’язаними з аналізом часових рядів, є:

1. Компоненти часового ряду:
   1.1. Тренд: довгострокова тенденція зміни ряду.
   1.2. Сезонність: регулярні та повторювані патерни, що відбуваються в певний період часу.
   1.3. Циклічність: довгострокові патерни, які не є такими регулярними, як сезонність.
   1.4. Випадковість: флуктуації або шум у даних.
2. Стаціонарність: часовий ряд вважається стаціонарним, коли його статистичні властивості, такі як середнє значення та дисперсія,

залишаються сталими у часі. Стаціонарність часто вважається бажаною при аналізі часових рядів, оскільки спрощує моделювання та прогнозування.

3. Автокореляція: автокореляція вимірює залежність між спостереженнями на різних моментах часу у межах часового ряду. Вона допомагає виявити патерни та залежності, такі як запізнені ефекти, де поточне значення залежить від попередніх значень.
4. Моделі часових рядів:
   4.1. Авторегресійні моделі (AR): ці моделі використовують минулі значення часового ряду для прогнозування майбутніх значень. Порядок такої моделі визначає кількість попередніх значень, що враховуються.
   4.2. Моделі рухомого (або ковзного) середнього (MA): ці моделі використовують залишкові помилки попередніх прогнозів для прогнозування майбутніх значень. Порядок моделі рухомого середнього визначає кількість попередніх залишкових помилок, що враховуються.
   4.3. Моделі авторегресійного ковзного середнього (ARMA): ці моделі поєднують обидва компоненти двох попередніх моделей для врахування основних патернів у часовому ряді (рис. 7).
   4.4. Авторегресійні інтегровані моделі ковзного середнього (ARIMA): ці моделі включають процедуру диференціювання для забезпечення стаціонарності часового ряду перед застосуванням моделей ARMA. Вони враховують тенденції та сезонність шляхом диференціювання даних.
   4.5. Сезонні ARIMA моделі: розширюють моделі ARIMA для роботи з сезонними патернами в даних.
   4.6. Подвійне експоненційне згладжування (DES). Експоненційне згладжування надає експоненційно меншу вагу старішим спостереженням. Просте експоненційне згладжування не дуже добре

працює, коли в даних є тенденція (тенденція означає, що середнє значення часового ряду збільшується або зменшується з часом) [14].

5. Прогнозування: Аналіз часових рядів дозволяє прогнозувати майбутні значення на основі історичних даних та виявлених патернів. Техніки, такі як експоненційне згладжування та моделі простору станів, можуть використовуватися для прогнозування.
6. Оцінка моделей: Різні метрики, такі як середньоквадратична помилка (MSE), середня абсолютна помилка (MAE) та корінь середньоквадратичної помилки (RMSE), часто використовуються для оцінки точності моделей часових рядів.

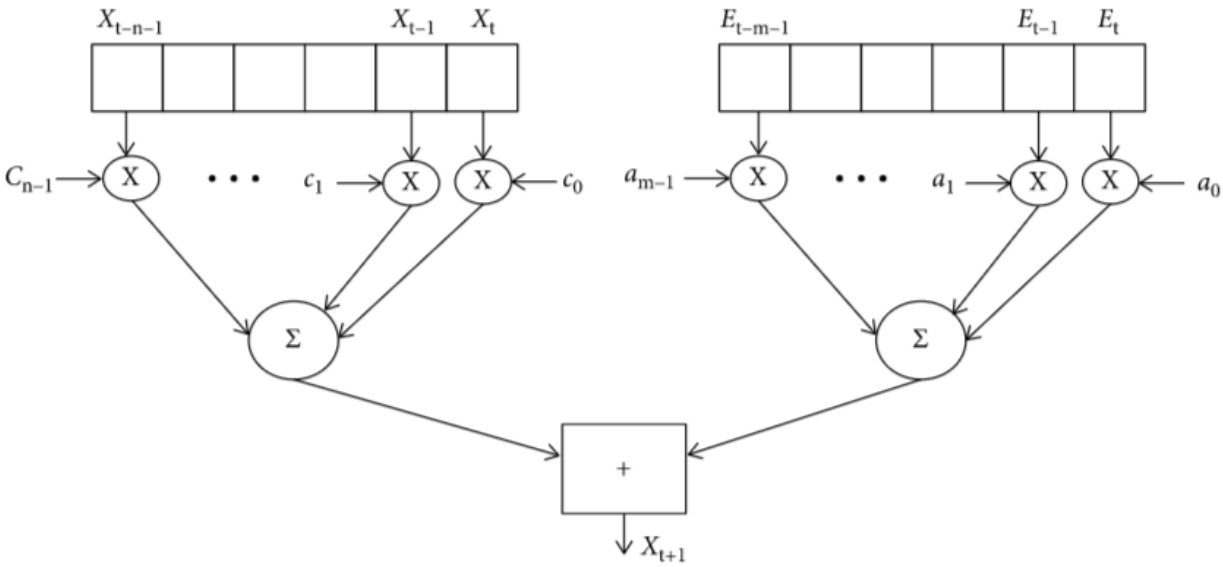

Рисунок 7. Приклад прогнозування ARMA моделлю. Окремо розраховуються AR та MA частини, кожне попереднє спостереження та помилковий член помножуються на свої відповідні коефіцієнти, після чого обидві частини додаються разом, щоб отримати кінцевий прогноз [14].

Аналіз часових рядів широко використовується в багатьох галузях, включаючи фінанси, економіку, екологічні дослідження та аналіз мережевого

трафіку, для розуміння та прогнозування часових патернів і прийняття обґрунтованих рішень [15].

Машинне навчання в моделюванні мережевого трафіку в реальному часі також відіграє важливу роль, використовуючи свою можливість аналізувати великі обсяги даних і виявляти складні закономірності та залежності. Воно надає цінні уявлення про поведінку мережі, прогнозує патерни трафіку, виявляє аномалії та покращує продуктивність та безпеку мережі.

Першим етапом використання машинного навчання є збір і попередня обробка даних: моделі машинного навчання потребують придатних та якісних даних для навчання. У контексті моделювання мережевого трафіку дані можуть бути зібрані з різних джерел, таких як мережеві пристрої, сенсори, журнали та захоплення пакетів. Ці дані часто потребують попередньої обробки, такої як фільтрація, нормалізація та вибір ознак, для підготовки до використання алгоритмів машинного навчання. Вибір ознак включає ідентифікацію найбільш відповідних та інформативних змінних або атрибутів із зібраних даних. Інженерія ознак передбачає створення нових ознак, які можуть покращити продуктивність моделей. У моделюванні мережевого трафіку ознаки можуть включати атрибути пакетів, обсяг трафіку, протоколи, час доби, IP-адреси джерела та призначення та інші [16].

Алгоритми навчання з учителем використовують марковані навчальні дані, де бажаний вихід або цільове значення відоме, для вивчення закономірностей та здійснення прогнозів. У контексті моделювання мережевого трафіку алгоритми навчання з учителем можуть використовуватися для таких задач, як класифікація трафіку (наприклад, відрізнення нормального та шкідливого трафіку) та прогнозування трафіку (наприклад, прогнозування майбутніх патернів трафіку) (рис. 8).

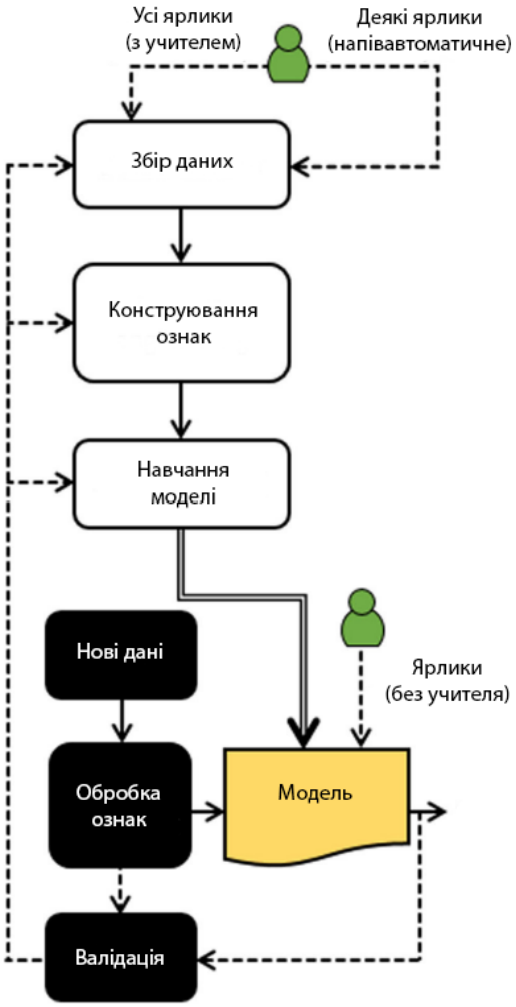

Рисунок 8. Компоненти рішень,
заснованих на машинному навчанні [16].

У свою чергу алгоритми навчання без учителя використовуються, коли навчальні дані не мають міток або відомих цільових значень. Ці алгоритми можуть виявляти приховані патерни, кластери та аномалії в даних мережевого трафіку. Техніки навчання без учителя, такі як кластеризація та виявлення аномалій, є цінними для ідентифікації незвичайної поведінки мережі або виявлення атак на мережу.

Для аналізу мережевого трафіку, що охоплює такі завдання як класифікація трафіку, виявлення вторгнень та прогнозування трафіку, використовується і глибинне навчання із застосуванням нейронних мереж. Моделювання трафіку реального часу вимагає можливості адаптації та оновлення моделей по мірі надходження нових даних. Технології навчання в режимі реального часу дозволяють проводити постійне навчання та удосконалення моделей без необхідності повного повторного навчання. Ці методи особливо корисні в динамічних мережевих середовищах, де патерни трафіку можуть змінюватися з часом.

Важливо зазначити, що машинне навчання в моделюванні мережевого трафіку в реальному часі є активною дослідницькою галуззю, і існує різноманітні підходи та алгоритми, пристосовані до конкретних завдань аналізу мережі [17].

Серед інших методів моделювання є застосування теорії черг, різних ймовірнісних моделей, модель суміші Гаусса, Баєсової мережі.

1.4. Специфіка моделювання трафіку реального часу методом симуляції та інструменти, що використовуються для цього.

Серед методів, що найбільш широко використовуються, важливим є метод симуляції. Симуляція дозволяє дослідникам та інженерам відтворювати поведінку мережі в контрольованому віртуальному середовищі. У симуляції мережевого трафіку створюється комп’ютерна модель, яка імітує роботу та взаємодію різних компонентів мережі, таких як маршрутизатори, комутатори та кінцеві вузли. Вона дозволяє аналізувати та оцінювати різні аспекти

мережевого трафіку, такі як продуктивність, завантаженість, затримки та якість обслуговування.

Загальними перевагами методу симуляції трафіку мережі вважають такі фактори:

1. Моделювання дозволяє розуміти складну динаміку трафіку мережі, включаючи ідентифікацію патернів, "вузьких місць" і потенційних точок збою. Це розуміння допомагає у плануванні, дизайні та оптимізації мережевих ресурсів
2. Симуляційні моделі можуть передбачати поведінку мережі в різних умовах, що є критично важливим у процесах прийняття рішень, таких як планування ємності, маршрутизація та балансування навантаження.
3. Моделювання забезпечує безпечне середовище для тестування змін в мережі або нових розгортань без переривання фактичної мережі. Це безризикове тестування особливо важливе для критично важливих мереж, де простій може призвести до значних фінансових або операційних втрат.
4. Симуляції можуть використовуватися для навчання адміністраторів мереж і професіоналів з кібербезпеки розумінню та пом'якшенню потенційних загроз. Вона може надати практичний досвід, що покращує навчання та розуміння устрою системи.
5. Моделювання може відтворювати поведінку дорогих, великомасштабних мереж без потреби в фізичних ресурсах. Цей ефективний підхід сприяє дизайну та плануванню модернізації та розширення мереж.

Серед недоліків та особливостей симуляції виділяють такі пункти:

1. Точність симуляційної моделі залежить від того, наскільки реалістично вона відтворює реальні динаміки трафіку. Якщо модель не точна, результати моделювання можуть бути оманливими, що може призвести до менш оптимального прийняття рішень.
2. Моделювання реального часу мережевого трафіку може бути обчислювально важким, особливо для великомасштабних мереж. Ця складність може вимагати значних обчислювальних ресурсів, що може обмежувати використання моделей моделювання.
3. Створення, валідація та калібрування моделей моделювання можуть бути трудомісткими. Зміни в поведінці мережі можуть вимагати частих оновлень моделі, збільшуючи витрати як фінансових, так і людських ресурсів на проектування системи.
4. Реальні симуляції для масштабних мереж можуть бути складними через високий рівень необхідних деталей та динамічну природу мережевого трафіку. Ця проблема може обмежувати застосовність симуляційних моделей у великомасштабних або швидко змінюваних мережах.

Процес моделювання трафіка методом симуляції починається з визначення мети симуляції трафіку мережі. Це може включати визначення оптимальної конфігурації для мережі, виявлення "вузьких місць" у потоці трафіку або перевірку нової структури мережі.

Далі виконується побудова топології. Це охоплює вказівку вузлів, зв’язків та їх властивостей, таких як пропускна здатність, затримка та розмір черг. Топологія мережі може базуватись на реальних мережах або бути розроблена спеціально для симуляції. Симуляція дозволяє налаштовувати

параметри мережі, включаючи фізичну топологію, конфігурацію мережевих пристроїв та протоколи. Це дає змогу вивчати різні сценарії та вплив змін на мережеву поведінку.

Наступним кроком є генерація трафіку. Реалістична генерація трафіку є важливою для точного моделювання складних мереж. Профіль трафіку - це опис типу трафіку, який потрібно імітувати. Це включає параметри, такі як тип протоколу (наприклад TCP, UDP), розмір пакету, швидкість пакету (пакетів в секунду), напрямок трафіку (в одну сторону, в обидва боки) та інші атрибути. У межах однієї мережі також можна використовувати різні профілі трафіку. Існує багато інструментів для генерування трафіку, включаючи Iperf, hping, tcpreplay та інші. Вибір залежить від типу трафіку, який потрібно генерувати, та від тонкості контролю, що вимагається для трафіку. Наприклад, Iperf часто використовується для тестування пропускної здатності, тоді як hping корисний для створення настроюваних пакетів. Після прийняття певного профілю трафіка, генератор трафіку треба налаштувати. Зазвичай це включає встановлення параметрів командного рядка або написання конфігураційного файлу [18].

Після запуску симуляції ведеться спостереження за роботою системи. Цей етап включає перевірку відповідності генерованого трафіку мережі очікуваному та аналіз його впливу на продуктивність мережі. Для цієї мети можуть бути корисні інструменти, такі як Wireshark, що дозволяють аналізувати пакети, що передаються через мережу. В першу чергу досліджуються такі характеристики, як використовуваний протокол, розмір пакетів та швидкість, з якою вони надсилаються.

Всі важливі результати процесу моніторингу трафіку слід документувати. Це може включати необроблені дані (наприклад, захоплення пакетів), а також будь-які графіки, діаграми або звіти, що створені на основі

цих даних. Ця документація може використовуватися для майбутнього посилання або як докази для підтвердження ваших результатів.

Висновки до розділу 1

У цьому розділу були визначені такі поняття як «трафік мережі» та «трафік реального часу», були описані деякі з основних методів моделювання трафіку та вимоги до трафіку реального часу. Трафік реального часу вимагає уваги до певних його характеристик в залежності від устрою системи. Усі методи моделювання відповідають різним призначенням та не замінюють один одного. Метод симуляції ефективно проявляє себе в моделюванні трафіку реального часу завдяки можливості передбачати різні сценарії та масштаби мережі, гнучкій конфігурації та безпеці, але вимагає точності в передачі деталей та характеристик мережі для досягнення реалістичної моделі.

РОЗДІЛ 2. МОДЕЛЮВАННЯ ТРАФІКУ РЕАЛЬНОГО ЧАСУ МЕТОДОМ СИМУЛЯЦІЇ

2.1. Побудова топології мережі та визначення її характеристик

Метою моделювання трафіку у цій роботі є демонстрація процесу використання програмного забезпечення для симуляції роботи мережі, тому реалістичністю характеристик мережі можна у певній мірі знехтувати.

Для моделювання трафіку реального часу було взято умовну схему системи індустріального контролю. Типовими для таких систем, як було зазначено у вступі до роботи, є такі вузли:

1. Набір сенсорів – пристроїв, що зчитують показники при виробництві та надсилають їх до керуючого вузлу, де ця інформація обробляється.
2. Пристрої контролю – отримують інформацію від серверу та впливають на промислові процеси.
3. Сервер – вузол, що отримує інформацію від пристроїв-сенсорів. Виконує обчислювальні задачі, приймає рішення та керує системою.

Масштаб мережі було обрано невеликий: один маршрутизатор, один комутатор, один сервер, чотири пристрою-датчика та два пристрою контролю.

Серед програмного забезпечення, що симулює роботу мережі на рівні пакетів, між NS-3, OPNET, Packet Tracer, GNS3 та інших було обрано саме GNS3 (Graphical Network Simulator-3), так як ця програма зарекомендувала себе як стандарт для освітньої та наукової роботи. Це безкоштовне та відкрите програмне забезпечення може використовуватися як у нескладних навчальних проектах, так і у комерційній діяльності. Розповсюджується на офіційному веб-сайті https://gns3.com, де можна знайти як документацію, так і платформу для спілкування спільноти людей, що користуються цією програмою.

Для побудови мережі, віртуальне обладнання MikroTik було обрано через його гнучкість у конфігурації та ліцензію, що дозволяє безкоштовно його використовувати. Варто звернути увагу на обмеження даного програмного забезпечення. При використанні образів мережевого обладнання MikroTik у віртуальних машинах, безкоштовною є така версія MikroTik CHR (Cloud Hosted Router), що встановлює ліміт швидкості інтерфейсів обладнання у 1 Мбіт/с. Для реальної мережі цього, звичайно, замало, але для демонстрації роботи мережі зі значними обмеженнями це підходить, бо спонукає використовувати методи проектування трафіку, що описані у розділі 1, пункті 1.2.

Інструментом генерації трафіку обрано утиліту iperf. Це універсальний інструмент, який може генерувати та вимірювати широкий спектр моделей трафіку, що робить його підходящим для моделювання мережі та симуляції. Це програмне забезпечення розповсюджується безкоштовно на веб-сайті https://sourceforge.net/projects/iperf2/.

Iperf виконує функції вимірювання максимально досяжної пропускної здатності в IP-мережах, а використовуючи UDP, iperf може вимірювати джитер, що особливо важливо для реальних додатків, таких як VoIP та потокові медіа, де високі значення джитеру можуть впливати на якість обслуговування. Важливо зазначити, що у контексті TCP концепція джитеру є дещо менш значущою, оскільки сам TCP надає механізми для обробки пакетів, які надходять не в порядку, повторних передач та контролю потоку, тим самим забезпечуючи стабільний і впорядкований потік даних для додатка. Іншими словами, TCP внутрішньо згладжує джитер, отже, хоча джитер може існувати на рівні IP, він не впливає безпосередньо на TCP-потік.

Для симуляції зв’язку датчиків та пристроїв контролю з сервером було вирішено використовувати протокол TCP, що буде імітувати протокол MQTT

(TCP/IP). MQTT часто використовується у системах IoT через його легкість і здатність підтримувати низьке споживання енергії, що корисно для пристроїв, які живляться від батареї. Він використовується в різних галузях, включаючи автомобільну, виробничу, телекомунікаційну, нафтогазову, та багато інших.

Винятком буде один з чотирьох пристроїв-датчиків – вузол, що грає роль IP-камери. Трафік, який цей вузол буде генерувати, буде імітувати RTP протокол, відправляючи UDP пакети.

Важливим для розуміння є умовність та приблизність усього трафіку мережі, так як симуляція по своїй суті передбачає лише імітацію навантаження на мережу.

Для збору інформації про трафік мережі та оцінку її характеристик використовувалася програма Wireshark. Це аналізатор мережевого протоколу, використовується у всьому світі для усунення неполадок в мережі, аналізу, розробки програмного забезпечення та комунікаційного протоколу, а також освіти. Wireshark відомий своїм розгорнутим набором функцій, який включає глибокий аналіз сотень протоколів, захоплення трафіку у реальному часі та офлайн-аналіз. Щодо ліцензування, Wireshark є вільним програмним забезпеченням, ліцензованим за GNU General Public License (GPL), тож може безкоштовно використовуватися як у комерційних, так и у навчальних цілях.

Першим етапом є побудова топології мережі на основі трьох вище описаних компонентів (рис. 9).

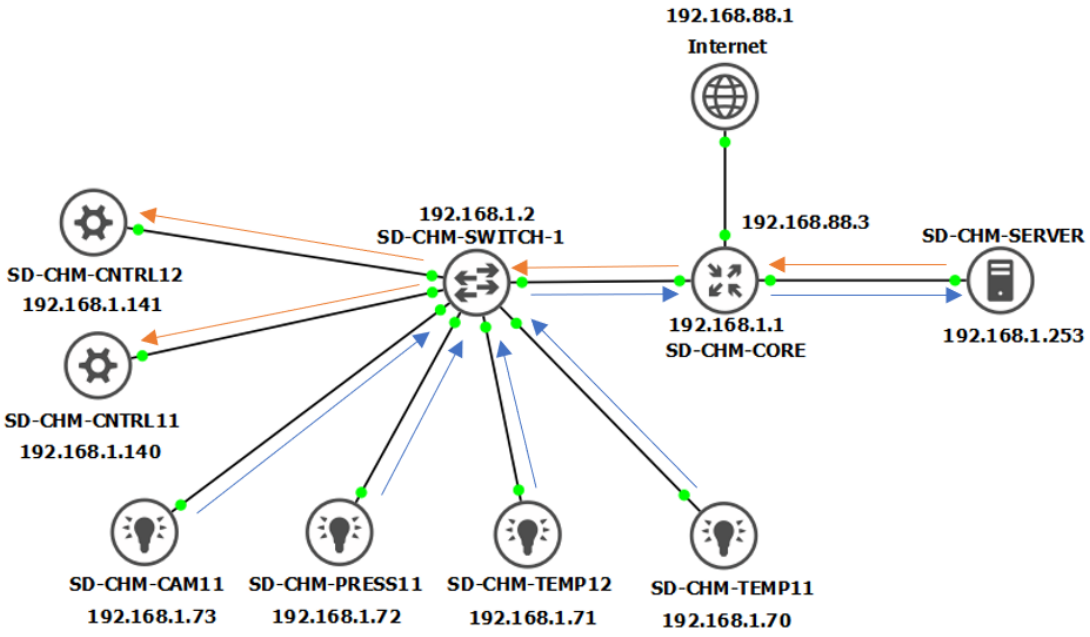

Рисунок 9. Топологія мережі.

У цій топології усі три типу пристроїв знаходяться в одній підмережі 192.168.1.0 з маскою 255.255.255.0 (/24).

Головним та єдиним маршрутизатором є SD-CHM-CORE з адресою 192.168.1.1, через нього проходить увесь трафік як у самій мережі. Використовується статична адресація через невеликий масштаб мережі та стабільний статичний набір вузлів. Щоб забезпечити зв’язок підмережі з мережею Інтернет, маршрутизатор підключений до локальної мережі комп’ютеру, на якому виконується практична робота (Internet, адрес 192.168.88.1).

Роль сервера, що збирає інформаціє, приймає рішення та керує системою, грає вузол SD-CHM-SERVER.

Набором сенсорів є вузли SD-CHM-TEMP11, SD-CHM-TEMP12, SD-CHM-PRESS11. Сюди ж можна віднести сервер, що імітує IP-камеру SD-CHM-CAM11. Ці пристрої у реальному часі (безперервно) надсилають серверу зібрані дані.

Як було зазначено вище, вузли SD-CHM-TEMP11, SD-CHM-TEMP12, SD-CHM-PRESS11 та SD-CHM-SERVER генеруватимуть TCP трафік, імітуючи протокол MQTT, може бути складним завданням, оскільки MQTT - це протокол рівня додатків зі своїми власними специфічними особливостями, такими як механізм keep-alive і його легкий заголовок пакетів. Наприклад, розмір пакета MQTT може бути від 2 байт до 256 мегабайт, проте в типовому сценарії IoT ці пакети часто досить малі, іноді не більше декількох десятків байтів. Також MQTT може мати досить випадкові інтервали між пакетами, але в цій практичній роботи розглядається саме безперервний потік даних, тому цим можна знехтувати.

2.2. Запуск симуляції. Збір метрик та аналіз якості трафіку

Отже TCP трафік спочатку був налаштований таким чином, щоб розмір пакетів був близьким до 64 байт. (рис 10). Але розмір TCP пакета (або сегмента) зазвичай керується стеком TCP у операційній системі, і додатки не можуть безпосередньо контролювати його. Фактори, такі як розмір MTU інтерфейсу мережі та механізм відкриття шляху MTU TCP, впливатимуть на дійсний розмір пакетів TCP в мережі. Тому точно та явно задати розмір TCP пакетів через iperf не є можливим.

Source	Destination	Protocol	Length
192.168.1.70	192.168.1.253	TCP	66
192.168.1.253	192.168.1.70	TCP	67

Рисунок 10. Захоплення пакетів TCP.

UDP трафік в даній мережі генерується лише одним вузлом - SD-CHM-CAM11, що імітує IP-камеру та протокол RTP. Так як пропускна здатність обладнання обмежена значенням в 1 мегабіт/с, заздалегідь можна передбачити проблеми з якістю сервісу UDP трафіку. Розмір UDP пакетів – приблизно 1200 байтів (рис 11).

Source	Destination	Protocol	Length	Info
192.168.1.73	192.168.1.253	UDP	1242	33833 → 5004 Len=1200
192.168.1.73	192.168.1.253	UDP	1242	33833 → 5004 Len=1200

Рисунок 11. Захоплення пакетів UDP.

Наступним кроком є спостереження за роботою мережі та збір інформації. Так як саме TCP трафік симулює основні функції системи (а «чистий» UDP протокол має надто просту структуру заголовків, що ускладнює його діагностику), має сенс в першу чергу звертати увагу на показники TCP трафіку (таблиця 1-2).

	Загальна кількість пакетів	Втрата пакетів, кількість	**Втрата пакетів, %**	**Середня затримка, мс**	Дублікати пакетів, кількість	**Дублікати пакетів, %**
TCP	3249	73	**2.25**	**265.4**	48	**1.48**
	13403	367	**2.74**	**227**	217	**1.62**
	41030	1159	**2.82**	**51.4**	754	**1.84**

Таблиця 1. Результати симуляції трафіку: втрата пакетів, затримка для дублювання пакетів.

	Адреса відправника	Адреса отримувача	Швидкість, кілобайт/с
TCP	192.168.1.70	192.168.1.253	113.59
	192.168.1.71	192.168.1.253	109.19
	192.168.1.72	192.168.1.253	107.74
	192.168.1.253	192.168.1.140	331.75
	192.168.1.253	192.168.1.141	335.51

Таблиця 2. Результати аналізу трафіку: швидкість усіх TCP потоків даних.

Затримку TCP пакетів можна виміряти кількома способами, і один з ефективних методів – використовувати значення «відносного початку» («Relative Start») у сеансах TCP. Цей метричний параметр в Wireshark ефективно позначає час між початком сеансу TCP та отриманням кожного пакета. Це дозволяє аналізувати час, який потрібен для проходження пакетів мережею після ініціації сеансу TCP. Відносний початок також дозволяє нам

порівняти затримку між різними пакетами в одному сеансі, що може допомогти визначити, чи є затримки постійними протягом сеансу, чи значно вони варіюються в різний час. Більш того, відносні початкові часи для пакетів TCP можуть надати уявлення про те, як працюють алгоритми управління перевантаженням TCP. Якщо ми бачимо постійне збільшення відносних початкових часів, це може вказувати на те, що включається механізм уникнення перевантаження TCP, що сповільнює швидкість передачі, щоб уникнути втрати пакетів. Та навпаки, якщо ми бачимо зменшення відносних початкових часів, це може свідчити про те, що вікно перевантаження зростає, дозволяючи більшій кількості пакетів проходити мережею одночасно.

Рівень втрати пакетів TCP трафіку вимірявся через кількість пакетів з позначкою «Попередній сегмент(и) не був захоплений» («Previous segment(s) not captured»). Втрачені пакети вказуються, коли є ACK для номера послідовності, але відповідний пакет не був знайдений – це показує, що пакет був надісланий, але його не було захоплено, зазвичай через те, що його було втрачено десь по шляху (рис. 12).

tcp.analysis.lost_segment

No.	Time	Source	Destination	Protocol	Length	Identification	Info
1264	6.238680	192.168.1.70	192.168.1.253	TCP	1514	0x5359 (21337)	[TCP Previous segment not captured]
1289	6.354068	192.168.1.72	192.168.1.253	TCP	1514	0x54fe (21758)	[TCP Previous segment not captured]
1313	6.441018	192.168.1.70	192.168.1.253	TCP	1514	0x535d (21341)	[TCP Previous segment not captured]
1376	6.789312	192.168.1.70	192.168.1.253	TCP	1514	0x5363 (21347)	[TCP Previous segment not captured]
1391	6.856564	192.168.1.71	192.168.1.253	TCP	1514	0xa7ae (42926)	[TCP Previous segment not captured]
1406	6.930515	192.168.1.72	192.168.1.253	TCP	1514	0x5503 (21763)	[TCP Previous segment not captured]
1433	7.040278	192.168.1.71	192.168.1.253	TCP	1514	0xa7b3 (42931)	[TCP Previous segment not captured]
1569	7.718741	192.168.1.71	192.168.1.253	TCP	1514	0xa7bf (42943)	[TCP Previous segment not captured]
1677	8.202748	192.168.1.72	192.168.1.253	TCP	1514	0x5519 (21785)	[TCP Previous segment not captured]
1790	8.755666	192.168.1.72	192.168.1.253	TCP	1514	0x5527 (21799)	[TCP Previous segment not captured]
1861	9.125820	192.168.1.71	192.168.1.253	TCP	1514	0xa7d5 (42965)	[TCP Previous segment not captured]
1871	9.190980	192.168.1.72	192.168.1.253	TCP	1514	0x552d (21805)	[TCP Previous segment not captured]
1992	9.740444	192.168.1.71	192.168.1.253	TCP	1514	0xa7de (42974)	[TCP Previous segment not captured]
2043	10.010781	192.168.1.72	192.168.1.253	TCP	1514	0x5535 (21813)	[TCP Previous segment not captured]
2056	10.075906	192.168.1.71	192.168.1.253	TCP	1514	0xa7e4 (42980)	[TCP Previous segment not captured]
2127	10.359601	192.168.1.72	192.168.1.253	TCP	1514	0x5541 (21825)	[TCP Previous segment not captured]
2212	10.802450	192.168.1.71	192.168.1.253	TCP	1514	0xa7f0 (42992)	[TCP Previous segment not captured]
2273	11.078317	192.168.1.71	192.168.1.253	TCP	1514	0xa7f7 (42999)	[TCP Previous segment not captured]
2296	11.169520	192.168.1.72	192.168.1.253	TCP	1514	0x554b (21835)	[TCP Previous segment not captured]

Рисунок 12. Захоплені TCP пакети з фільтром по позначці «Попередній сегмент не був захоплений».

Але так як рівень втрачених пакетів не є вичерпним для розуміння загального рівню помилок у трафіку, також було виміряно кількість ACK-дублюючих пакетів («Duplicate ACK»). Також важливо розуміти, що трафік може містити і пакети з іншими прапорами-помилками, і вони часто пов’язані один с одним. Наприклад коли відправник отримує три дублюючі ACK (це називається потрійне дублювання ACK), він припускає, що сегмент був втрачений, і повторно передає цей конкретний сегмент до того, як сплине його таймер – цей механізм називається «Швидка Ретрансляція» («Fast Retransmission»). Тому у трафіку можна помітити пакети з позначкою «ACK Retransmission», і хоча «Дубльований ACK» часто викликає «Ретрансляцію», ці два фактори не завжди співпадають один до одного.

Якщо з приведеної таблиці визначити середні значення показників базуючись на трьох вибірках, то отримаємо 2.6% втрати пакетів, 181 мс затримки та 1.6% дублікатів. Хоча відмінності в показниках затримки при різній кількості пакетів у вибірці можуть свідчити не про високу затримку взагалі, а про великий розкид цього значення, що за визначенням можна назвати джитером.

Рівень втрат пакетів 2.6% може бути прийнятним через характер IoT-пристроїв та інфраструктури мережі, на якій вони зазвичай розгортаються.

Дублювання на рівні 1.6% вказує на те, що деякі пакети передаються більше одного разу. Це зазвичай відбувається через механізми повторної передачі TCP, що активуються, швидше за все, через втрату пакетів або високу затримку. Це додає навантаження на мережу і може сприяти збільшенню затримки та втрати пакетів. У деяких випадках це може бути прийнятним, але це зазвичай свідчить про деякий рівень проблеми мережі.

Для MQTT в IoT-сценаріях вказаний вище рівень затримки може бути прийнятними, якщо вимоги до реального часу не є строгими (наприклад, моніторинг даних сенсорів). Однак, для систем управління, де рішення повинні бути прийняті негайно (наприклад, промислові системи управління), це зазвичай поганий показник.

Так як у напрямку до серверу SD-CHM-SERVER кількість пакетів більше, швидкість TCP трафіку в ньому менша, ніж швидкість у напрямку від сервера до контролерів (SD-CHM-CNTRL-70 та SD-CHM-CNTRL-71).

UDP-трафік є менш пріоритетним у системі, тому з його характеристик була виділена лише швидкість передачі даних (таблиця 3).

	Тривалість роботи мережі, секунд	Об'єм переданої інформації, мегабайт	Швидкість передачі, кілобайт/с
UDP	23.59	2.13	92.16
	94.97	8.11	87.04
	279.73	21.87	79.87

Таблиця 3. Швидкість передачі даних UDP трафіку мережі.

Враховуючи те, що частину каналу мережі забирає на себе IP-камера, яка у цій системі є побічною, а можливості обмежені пропускною здатністю мережевого обладнання, одним з рішень може бути пріоритизація TCP трафіку за рахунок UDP. Надання пріоритету TCP перед UDP може потенційно покращити якість TCP-потоків, забезпечуючи їм більше пропускної здатності та менше конкуренції в мережі. Це може призвести до зменшення затримки,

меншої втрати пакетів та меншої кількості повторних передач (дублікатів) для TCP-трафіку.

Для цього маршрутизатор був налаштований таким чином, щоб маркувати усі пакети відповідними їх протоколу ярликами, після чого задати пакетам с різними ярликами різний пріоритет у «Queue Tree» (рис 13).

```
[admin@MikroTik] > ip firewall mangle print
Flags: X - disabled, I - invalid; D - dynamic
 0    chain=prerouting action=mark-packet new-packet-mark=tcp_packet
      passthrough=yes protocol=tcp log=yes

 1    chain=prerouting action=mark-packet new-packet-mark=udp_packet
      passthrough=yes protocol=udp log=yes
[admin@MikroTik] > queue tree print
Flags: X - disabled, I - invalid
 0   name="TCP Queue" parent=global packet-mark=tcp_packet limit-at=0
     queue=default priority=1 max-limit=0 burst-limit=0 burst-threshold=0
     burst-time=0s bucket-size=0.1

 1   name="UDP Queue" parent=global packet-mark=udp_packet limit-at=0
     queue=default priority=8 max-limit=0 burst-limit=0 burst-threshold=0
     burst-time=0s bucket-size=0.1
[admin@MikroTik] > ip firewall mangle print stats
Columns: CHAIN, ACTION, BYTES, PACKETS
# CHAIN       ACTION             BYTES  PACKETS
0 prerouting  mark-packet  627 533 687  660 090
1 prerouting  mark-packet  477 152 507  398 222
[admin@MikroTik] > queue tree print stats
Flags: X - disabled, I - invalid
 0   name="TCP Queue" parent=global packet-mark=tcp_packet rate=857696
     packet-rate=115 queued-bytes=0 queued-packets=0 bytes=596441329
     packets=662301 dropped=0

 1   name="UDP Queue" parent=global packet-mark=udp_packet rate=728624
     packet-rate=76 queued-bytes=0 queued-packets=0 bytes=451564751
     packets=376519 dropped=0
[admin@MikroTik] >
```

Рисунок 13. Конфігурація файрволу та черг маршрутизатора.

Система «Queue Tree» («Дерево Черг») в MikroTik – це потужна та гнучка функція, розроблена для управління та контролю потоку пакетів через маршрутизатор. Вона побудована на концепції ієрархії спадкування, де кожна черга може мати один батьківський «клас» та кілька дочірніх. Дерево черг може бути досить складним, з можливістю мати численні гілки, всі з власними

рівнями пріоритету та обмеженнями швидкості. Кожна черга має свої властивості, включаючи батьківську чергу, пріоритет, максимальний ліміт, обмеження швидкості, порогове обмеження, час обмеження та вказівку на батьківську чергу або інтерфейс (звідки буде братися трафік).

Судячи зі статистики пакетів у чергах, можна зробити висновок, що пропускної здатності достатньо і для TCP, і для UDP трафіку мережі, так як довго пакети в чергах не затримувались та не було потреби будь-які пакети викидати. Тоді роль черги у даній мережі зводиться саме до надання пакетам TCP більшого пріоритету без сильного перешкоджання UDP трафіку.

У результаті запуску симуляції мережі з пріоритизацією TCP пакетів та збору інформації про трафік було отримано такі результаті (таблиця 4).

	Загальна кількість пакетів	Втрата пакетів, кількість	**Втрата пакетів, %**	**Середня затримка, мс**	Дублікати пакетів, кількість	**Дублікати пакетів, %**
TCP	3321	80	**2.41**	**59.2**	64	**1.93**
	12557	337	**2.68**	**57.8**	200	**1.59**
	38110	991	**2.60**	**44.8**	700	**1.84**

Таблиця 4. Результати симуляції трафіку з пріоритезацією TCP трафіку

Рівень втрати пакетів та дублікатів залишився на колишньому рівні, в той час як затримка стала стабільно низькою. Так як основна проблема першої симуляції була саме в великому розкиду затримки, після пріоритизації затримка стала меншою та більш стабільною.

Загалом для багатьох систем реального часу затримка менше 100 мілісекунд часто вважається прийнятною. Це пов'язано з тим, що людям важко

сприймати затримки, які коротші за цей час. Однак деякі додатки IoT можуть мати набагато строгіші вимоги до затримки. Наприклад, у автоматизованих виробничих системах або робототехніці затримки можуть бути менше 10 мс або навіть 1 мс, щоб забезпечити точний контроль і своєчасну реакцію на швидко змінювані умови.

Побічним ефектом пріоритизації трафіку стало зменшення швидкості передачі даних пакетами TCP, що скоріш за все є наслідком роботи механізму компенсації внаслідок перевантаження, хоча TCP і має більший пріоритет (таблиця 5).

	Адреса відправника	Адреса отримувача	Швидкість, кілобайт/с
TCP	192.168.1.70	192.168.1.253	97.63
	192.168.1.71	192.168.1.253	99.87
	192.168.1.72	192.168.1.253	97.31
	192.168.1.253	192.168.1.140	306.48
	192.168.1.253	192.168.1.141	304.06

Таблиця 5. Швидкість передачі даних TCP пакетами.

Також цікавим наслідком пріоритизації TCP стала більш стабільна швидкість UDP-пакетів. Незважаючи на те, що UDP тепер має нижчий пріоритет, він теж виграє від меншого навантаження на мережу. Оскільки UDP не знижує швидкість як механізм компенсації при втраті пакетів, як це робить TCP, він може продовжувати відправляти пакети з однаковою швидкістю. Таким чином, при меншому перевантаженні мережі в цілому, може бути

менше втрат пакетів для UDP, що може призвести до більш стабільної швидкості (таблиця 6).

UDP	Тривалість роботи мережі, секунд	Об'єм переданої інформації, мегабайт	Швидкість передачі UDP, кілобайт/с
	25.87	2.18	86.02
	104.4	8.74	86.03
	308.78	25.55	84.99

Таблиця 6. Швидкість передачі даних UDP після пріоритизації TCP трафіку.

Якщо дослідити динаміку пакетів-помилок в трафіку до пріоритизації та після, то можна побачити різницю в патерні. Після пріоритизації трафіку середній рівень помилок зменшився, а спалахи залишилися, хоча й теж стали меншими. Якщо порівняти загальну кількість помилок до і після – 11.5% та 10.3% відповідно, то різниця становить 1.2% (рис. 14-15).

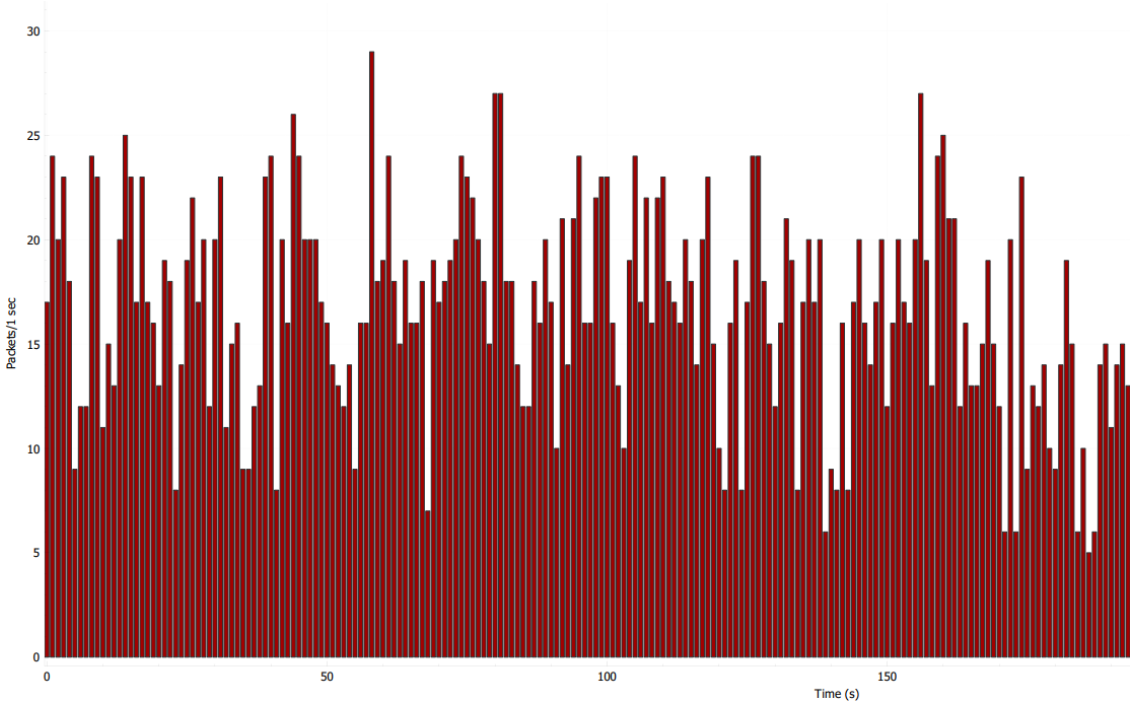

Рисунок 14. Кількість помилкових пакетів у секунду до пріоритизації TCP трафіку.

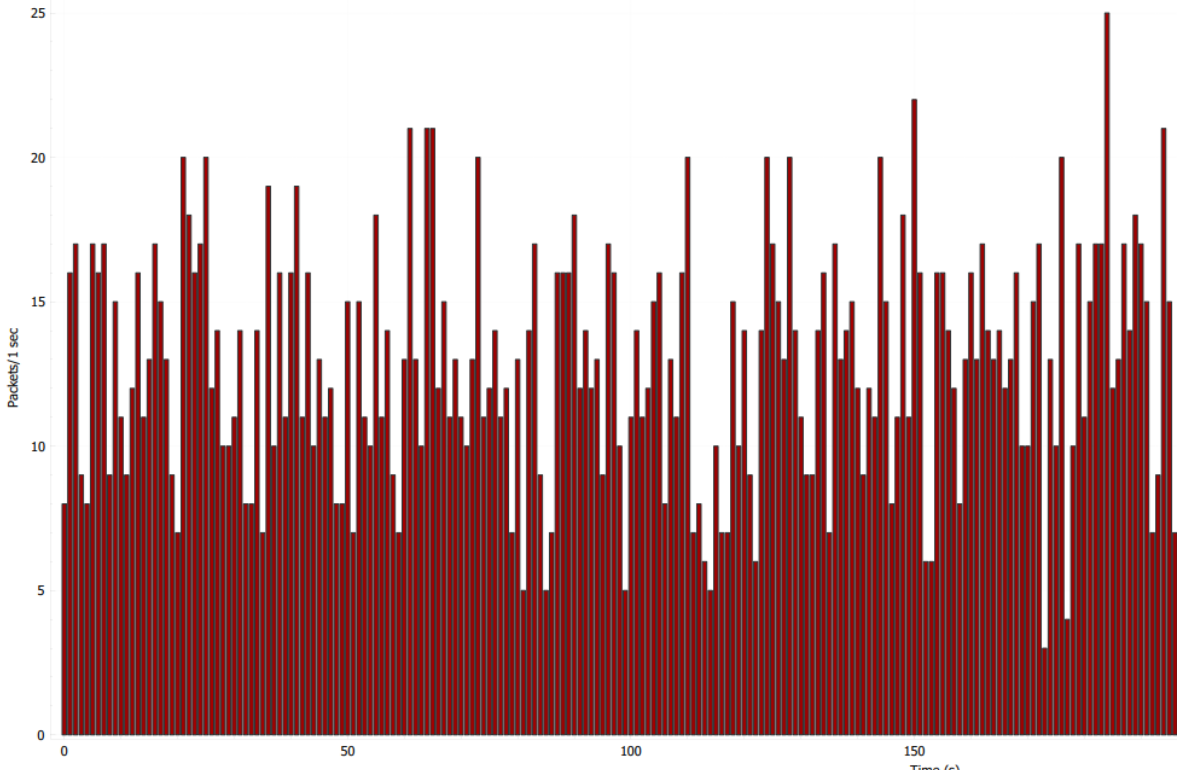

Рисунок 15. Кількість помилкових пакетів у секунду після пріоритизації TCP трафіку.

Висновки до розділу 2

У практичній роботі було продемонстровано основні етапи процесу моделювання трафіку реального часу методом симуляції, що охоплює: задання мети, побудову топології, генерацію трафіку та його подальший аналіз. Описано методи збору таких метрик як затримка, втрата та дублювання пакетів, проведено їх порівняння та динаміку характеристик трафіку з урахуванням особливостей програмного забезпечення. В цілях демонстрації методів забезпечення якості обслуговування мережі був використаний метод пріоритизації трафіку одного типу, внаслідок чого певні характеристики трафіку мережі покращилися та робота мережі стала більш стабільною, а побічні ефекти були розглянуті та пояснені.

ВИСНОВКИ

Дана робота надала огляд моделювання реального трафіку в комп'ютерних мережах, з акцентом на методи моделювання (зокрема метод симуляції) та механізми забезпечення якості обслуговування (QoS). Ця робота спрямована на освітлення складних аспектів якості реального трафіку, який є критичним аспектом управління мережею в системі реального часу в сучасному світі.

Практична частина цієї роботи закріпила теоретичні основи через моделювання умовної промислової IoT мережі, що складається з датчиків, серверу та контролерів. Застосування механізмів пріоритету трафіку служило практичним свідченням способів, за допомогою яких може бути забезпечено QoS в мережі, що призводить до більш ефективної і гладкої роботи мережі, потенційно знижуючи затримку пакетів та кількість помилок. Важливим аспектом також є передбачення побічних ефектів впровадження таких механізмів.

СПИСОК ВИКОРИСТАННИХ ДЖЕРЕЛ

[1] Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. Future Generation Computer Systems, 29(7), 1645-1660.

[2] Paxson, V. & Floyd, S. (1995). Wide-area traffic: The failure of Poisson modeling. IEEE/ACM Transactions on Networking, 3(3), 226-244.

[3] Leland, W. E., Taqqu, M. S., Willinger, W., & Wilson, D. V. (1994). On the Self-Similar Nature of Ethernet Traffic (Extended Version). IEEE/ACM Transactions on Networking, 2(1), 1-15.

[4] Tanenbaum, A. S., & Wetherall, D. J. (2010). Computer Networks (5th ed.). Prentice Hall.

[5] Roughan, M., Sen, S., Spatscheck, O., & Duffield, N. (2002). Class-of-service mapping for QoS: A statistical signature-based approach to IP traffic classification. Proceedings of the ACM SIGCOMM Internet Measurement Workshop.

[6] Ferrari, D. (1990). Client Requirements for Real-Time Communication Services. IEEE Communications Magazine, 28(11), 65-72.

[7] Bonaventure, O. (2011). Computer Networking: Principles, Protocols and Practice. CreateSpace Independent Publishing Platform..

[8] Zhang, Y., & Chen, Y. (2018). A Survey on Quality of Service for Real-Time Multimedia Applications on the Internet. IEEE Communications Surveys & Tutorials, 20(2), 1164-1193. doi: 10.1109/comst.2018.2796513.

[9] Garcia-Luna-Aceves, J. J., & Spohn, M. (1995). Traffic shaping in real-time networks: A survey. IEEE Network, 9(2), 8-22.

[10] Jimenez J, Cisco Systems, Inc. Traffic Policing and Shaping, Guide to Managing Cisco Network Traffic, 2022, URL:

https://www.cisco.com/c/en/us/support/docs/quality-of-service-qos/qos-policing/19645-policevsshape.html (дата звернення: 03/04/2023).

[11] Abbasi, M. R., Guleria, A., & Devi, M. S. (2016). Traffic Engineering in Software Defined Networks: A Survey. Journal of Telecommunications and Information Technology 4(2016):3-14.

[12] Norris, J. R. (1998). Markov chains. Cambridge University Press.

[13] Liu, F., Ting, K. M., & Zhou, Z. H. (2012). Isolation-based anomaly detection. ACM Transactions on Knowledge Discovery from Data (TKDD), 6(1), 1-39.

[14] Xu, Z., Iqbal, M. F., Zahid, M., Habib, D., & John, L. K. (2019). Efficient Prediction of Network Traffic for Real-Time Applications. Journal of Computer Networks and Communications, 2019, 4067135. https://doi.org/10.1155/2019/4067135.

[15] Chatfield, C. (2016). The Analysis of Time Series: An Introduction (6th ed.). CRC Press.

[16] Boutaba, R., Salahuddin, M. A., Limam, N., Ayoubi, S., Shahriar, N., Estrada-Solano, F., & Caicedo, O. M. (2020). A comprehensive survey on machine learning for networking: evolution, applications and research opportunities., Journal of Internet Services and Applications, Springer Nature.

[17] Ding, H., Wang, L., Li, G., & Yeung, D. Y. (2017). Deep learning for traffic prediction and classification in SDN: A survey. IEEE Communications Surveys & Tutorials, 19(4), 2596-2620.

[18] Thomas T. Chen (2010). Network Traffic: Modeling and Control. John Wiley & Sons.

[19] Alkahtani, A., Woodward, M., & Al-Begain, K. (2023). An Overview of Quality of Service (QoS) and QoS Routing in Communication Networks..