

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Харківський національний університет радіоелектроніки

Центр післядипломної освіти
Кафедра Програмної інженерії

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

другий (магістерський)

(рівень вищої освіти)

Дослідження моделей управління якістю потокового відео у реальному часі
в залежності від сцени

Виконав:

студент 2 курсу групи ППЗдм-21-2

Ігнатов Д.О.

(прізвище, ініціали)

Спеціальність 121 – Інженерія програмного

забезпечення

(код і повна назва спеціальності)

Тип програми Освітньо-наукова

Керівник доц. Вечур О.В.

(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри, проф.

Дудар З.В.

2023 р.

Харківський національний університет радіоелектроніки

Факультет _____ Центр післядипломної освіти _____
 Кафедра _____ Програмної Інженерії _____
 Рівень вищої освіти _____ другий (магістерський) _____
 Спеціальність _____ 121 – Інженерія програмного забезпечення _____
 (код і повна назва спеціальності)
 Тип програми _____ Освітньо-наукова _____
 Освітня програма _____ Інженерія програмного забезпечення _____

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

« _____ » _____ 2023 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

студента _____ Ігнатов Дмитро Олегович _____

1. Тема роботи «Дослідження моделей управління якістю потокового відео у реальному часі в залежності від сцени» _____

затверджена наказом по університету від « 03 » квітня 2023 р. № 83Стз

2. Термін подання студентом роботи до екзаменаційної комісії «22» травня 2023р.

3. Вихідні дані до роботи _аналіз систем протоколів та апаратного забезпечення, аналіз методів кодування для онлайн трансляцій, організація відеотрансляції через інформаційні мережі з високою візуальною якістю _____

4. Перелік питань, що потрібно опрацювати в роботі мета роботи, аналіз предметної галузі та постановка задачі, дослідження існуючих технологій обробки зображення, формування вимог до програмної системи, рекомендації стосовно актуальності використання сучасних технологій _____

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз предметної галузі	26.01.2023	виконано
2	Здійснення огляду існуючих методів розробки	28.01.2023	виконано
3	Постановка задачі	05.02.2023	виконано
4	Виділення критеріїв порівняння, обґрунтування вибору	15.02.2023	виконано
5	Розробка підходів та створення формул для проведення обчислень	22.02.2023	виконано
6	Проведення експериментального дослідження, документування результатів	10.04.2023	виконано
7	Підготовка пояснювальної записки	27.04.2023	виконано
8	Перевірка роботи на антиплагіат	23.05.2023	виконано
9	Нормоконтроль	23.05.2023	виконано
10	Рецензування	23.05.2023	виконано
11	Підготовка презентації та доповіді	23.05.2023	виконано
12	Попередній захист	23.05.2023	виконано
13	Занесення роботи в електронний архів	23.05.2023	виконано
14	Допуск до захисту у зав. кафедри	23.05.2023	виконано

Дата видачі завдання _____ 202__ р.

Студент _____

(підпис)

Керівник роботи _____ доц. Вечур О.В. _____

(підпис)

РЕФЕРАТ / ABSTRACT

Кваліфікаційна робота магістра містить: 91 с., 20 рис., 2 табл., 17 джерел.

АРХІТЕКТУРА Р2Р, БІТРЕЙТ, ВІДЕОКОНТЕНТ, ОРГАНІЗАЦІЯ ВІДЕОТРАНСЛЯЦІЇ, ПЕРЕДАВАННЯ ДАНИХ, ПОТОКОВИЙ СЕРВЕР, ПРОТОКОЛИ ПРЯМИХ ВІДЕОТРАНСЛЯЦІЙ, H.264, LIVE STREAMING, MPEG-4, UDP

Актуальність цієї роботи полягає в тому, що потокові медіа відкривають зовсім нові можливості для поширення інформації, освіти та повідомлень. Використання Інтернету дозволяє знизити витрати на поширення значно нижче, ніж у традиційних засобах масової інформації.

Аудиторія Інтернету є досить великою і з кожним днем зростає, тому контент повинен бути привабливим, щоб забезпечити високі показники перегляду, що забезпечать повернення вкладених коштів у трансляцію.

Мета та завдання даного дослідження полягають в аналізі сучасних систем протоколів та апаратного забезпечення, які дозволяють організації відеотрансляції через інформаційні мережі з високою візуальною якістю, порівнювально з системами цифрового телебачення. Для досягнення цієї мети необхідно виконати такі завдання: проаналізувати методи кодування для онлайн трансляцій, проаналізувати архітектуру мережі, здійснити практичне порівняння сучасних технологій організації відеотрансляцій та платформ для прямих відеотрансляцій, а також сформулювати рекомендації стосовно актуальності використання сучасних технологій та платформ.

Об'єктом дослідження є технології передачі відеосигналу за допомогою інформаційних мереж.

Предметом дослідження є вплив обраних параметрів для онлайн відеотрансляцій та каналів передачі на якісні показники відеозображення.

Методи дослідження включають теоретичне вивчення наявних технологій та платформ для передачі відео та аудіо в реальному часі, порівняння їх та аналіз

запропонованих рішень для проведення прямих трансляцій та вебінарів як натурних експериментів.

Новизною отриманих результатів є пропозиція використовувати спеціальні технології для підвищення надійності передачі відеоконтенту за допомогою інформаційних мереж, що дозволяє поліпшити якість сприйняття аудіо та відео контенту користувачем і покращувати перспективи застосування Інтернет-мовлення в освіті.

Також пропонується використовувати окремі платформи та пакети програм під час передачі відео в реальному часі для поліпшення якості відеотрансляцій за допомогою інформаційних мереж.

Отримані результати мають практичне значення. Запропонований підхід до організації відеотрансляцій може бути корисним для навчальних закладів та корпорацій, які бажають використовувати його для навчання студентів, інформування співробітників про нові технології, а також для організації навчальних та виробничих процесів.

P2P ARCHITECTURE, BITRATE, VIDEO CONTENT, VIDEO ORGANIZATION, DATA TRANSFER, STREAMING SERVER, LIVE VIDEO PROTOCOLS, H.264, LIVE STREAMING, MPEG-4, UDP,

The relevance of this work lies in the fact that streaming media opens up entirely new possibilities for the dissemination of information, education and messages. The use of the Internet makes it possible to reduce the costs of distribution significantly lower than in traditional mass media.

The audience of the Internet is quite large and growing every day, so the content must be attractive to ensure high viewing figures, which will ensure the return of the invested funds in the broadcast.

The purpose and objectives of this study are to analyze modern protocol and hardware systems that allow the organization of video broadcasting through information networks with high visual quality, comparable to digital television systems. To achieve

this goal, it is necessary to perform the following tasks: study coding methods for online broadcasts, analyze the network architecture, make a practical comparison of modern technologies for organizing video broadcasts and platforms for live video broadcasts, as well as formulate recommendations regarding the relevance of using modern technologies and platforms.

The object of research is video signal transmission technology using information networks.

The subject of the study is the influence of the selected parameters for online video broadcasts and transmission channels on the quality indicators of the video image.

Research methods include a theoretical study of existing technologies and platforms for real-time video and audio transmission, their comparison, and analysis of proposed solutions for conducting live broadcasts and webinars as field experiments.

The novelty of the obtained results is the proposal to use special technologies to increase the reliability of video content transmission using information networks, which allows to improve the quality of perception of audio and video content by the user and improve the prospects of using Internet broadcasting in education.

It is also suggested to use separate platforms and software packages during real-time video transmission to improve the quality of video broadcasts using information networks.

The obtained results are of practical importance. The proposed approach to the organization of video broadcasts can be useful for educational institutions and corporations that wish to use it for training students, informing employees about new technologies, as well as for organizing educational and production processes.

Я, Ігнатов Дмитро Олегович, студент групи ІПЗзДМ-21-2, здобувач вищої освіти на другому (магістерському) рівні кафедра програмної інженерії, заявляю: моя кваліфікаційна робота на тему ДОСЛІДЖЕННЯ МОДЕЛЕЙ УПРАВЛІННЯ ЯКІСТЮ ПОТОКОВОГО ВІДЕО У РЕАЛЬНОМУ ЧАСІ В ЗАЛЕЖНОСТІ ВІД

СЦЕНИ, що буде представлена до ЕК для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в електронному архіві відкритого доступу EIArKhNURE. Всі запозичення з друкованих та електронних джерел мають відповідні посилання. Я ознайомлений з діючим положенням «Про протидію академічному плагіату в ХНУРЕ», згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ

ТБ – Телебачення;

ААС – Advanced Audio Coding;

АВР – Адаптивний бітрейт;

АVI – Audio Video Interleave (медіаконтейнер, вперше виконаний Microsoft);

СВР – скорочення від «constant bit rate», в перекладі – «постійний; потік даних», він же «сібїер», він же «констант бітрейт», він же «постійний бітрейт»;

СДN – це розподілена архітектура, в якій дані розподіляються по Різному на сервері;

DVD – Digital Versatile Disc (цифровий багатоцільовий диск);

HLS (HTTP Live Streaming) – комунікаційний протокол для потокової передачі медіа на основі HTTP;

HTTP – HyperText Transfer Protocol;

IPТV – «Internet Protocol Television» це доставка телевізійного контенту по мережах Інтернет-протоколу (IP);

IQA – Початкове адаптування якості;

MPEG – Moving Picture Experts Group (стандарт цифрового стиснення відео та аудіо);

NDI – Network device interface (інтерфейс мережного пристрою);

OTT – Over-the-top (сервіс потокового мультимедіа);

PQA – Прогресивна адаптація якості;

P2P – Peer-to-peer (Однорангові обчислення);

RTP – Real Time Protocol (транспортний протокол реального часу);

TCP – Transmission Control Protocol (протокол керування передачею);

VoD – Video on Demand (відео на вимогу);

VPN – Virtual Private Network (віртуальна приватна мережа);

ЗМІСТ

РЕФЕРАТ / ABSTRACT	4
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ	8
ВСТУП.....	12
1 МЕТОДИ КОДУВАННЯ ДЛЯ ПРЯМИХ ВІДЕО ТРАНСЛЯЦІЙ	14
1. Класифікація потокового відео	14
1.1. VoD	14
1.2. Live streaming	15
2. Архітектура	17
2.1. Архітектура розподіленої мережі вмісту (CDN).....	17
2.2. Peer-to-peer (P2P).....	18
2.3. Бітрейт	19
2.3.1. CBR кодування.....	19
2.3.2. ABR кодування	21
2.4. Кодування відео	22
2.5. Захоплення відео	23
2.6. MPEG-1.....	24
2.7. MPEG-2.....	25
2.8. MPEG-4.....	28
2.9. Розширений відеокодек H.264	29
2.10. Стиснення звуку.....	32
Висновки до розділу	35
2 ТЕХНОЛОГІЇ ОРГАНІЗАЦІЇ ВІДЕОТРАНСЛЯЦІЙ	37
1. Адаптивна мультимедійна трансляція	37
1.1. Адаптивне планування	37
1.2. Адаптивний вибір швидкості	41
2. Протоколи транспортування	43
2.1. HyperText Transport Protocol (HTTP)	43
2.2. UDP	44
2.3. TCP	47
2.4. RTP	51

	11
3. Потоковий сервер.....	54
4. Попередня обробка	56
5. Пост-обробка.....	58
Висновки до розділу	60
3 ПРАКТИЧНЕ ПОРІВНЯННЯ СУЧАСНИХ ІНСТРУМЕНТІВ ДЛЯ ПРОВЕДЕННЯ ПРЯМИХ ВІДЕОТРАНСЛЯЦІЇ	61
1. Практична реалізація відеотрансляцій за допомогою HLS (HTTP Live Streaming)	61
1.1. Як працює HLS	61
1.2. Переваги та недоліки HLS.....	62
2. Практична реалізація відеотрансляцій за допомогою RTMP (Real-Time Messaging Protocol)	63
2.2. Переваги та недоліки RTMP.....	64
Висновки до розділу	65
4 ЕКСПЕРИМЕНТАЛЬНА ПЕРЕВІРКА РЕЗУЛЬТАТІВ АНАЛІТИЧНИХ ДОСЛІДЖЕНЬ	67
1. Реалізація коду	67
2. Вимір бітрейту	70
3. Висновки до розділу	73
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	75
ДОДАТОК А ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ ЗА НАУКОВИМИ НАПРЯМАМИ КЕРІВНИКА ТА НАУКОВЦІВ КАФЕДРИ ПРОГРАМНОЇ ІНЖЕНЕРІЇ	77
ДОДАТОК Б ПРИКЛАД КОДУ СЕРВЕРА	78
ДОДАТОК В Стаття з I Міжнародної науково-практичної конференції	81
ДОДАТОК Г ЗВІТ РЕЗУЛЬТАТІВ ПЕРЕВІРКИ НА УНІКАЛЬНІСТЬ ТЕКСТУ .	84
ДОДАТОК Д СДАЙДИ ПРЕЗЕНТАЦІЇ	85
ДОДАТОК Е ЕКСПЕРТНИЙ ВИСНОВОК РЕЗУЛЬТАТІВ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ НА ВІДПОВІДНІСТЬ ОФОРМЛЕННЯ «ВИМОГИ ДСТУ 3008:2015»	91

ВСТУП

Потокові медіа – цікаве доповнення до арсеналу інструментів, які використовуються виробниками мультимедійних матеріалів. Так само, як телебачення витіснило кіно та радіо як головне комунікаційне середовище, потокове передавання призначене для того, щоб змінити Інтернет.

Хоча оригінальні стандарти Інтернету були розширені далеко за межі початкової функціональності основних протоколів для включення зображень та анімації, відео та аудіо стали найбільш природними способами комунікації. Завдяки телевізійному досвіду ми тепер очікуємо від відео бути головним засобом розповсюдження знань та розваг. Це спричинило постійний розвиток, що дозволяє відео транслюватися через Інтернет як прямий ефір. Багато людей розглядають потокове передавання як альтернативний канал доставки звичайного радіо та телебачення – відео через IP.

Проте, це занадто обмежена точка зору, оскільки потокове передавання може бути найбільш переконливим, коли використовуються його особливі переваги.

У рамках інтерактивної мультимедійної презентації, воно стає цілком новим комунікаційним каналом, який може конкурувати з друком, радіо та телебаченням.

Мета та завдання цього дослідження полягають у вивченні сучасних моделей управління якістю потокового відео, які дозволяють організувати відеотрансляцію через інформаційні мережі з високою візуальною якістю, порівнянною з системами цифрового телебачення. Для досягнення цієї мети поставлено такі завдання: вивчити методи кодування для онлайн трансляцій, проаналізувати архітектуру мережі, порівняти сучасні технології організації відеотрансляції та платформи для прямих відеотрансляцій на практиці, і дати рекомендації щодо використання актуальних технологій та платформ.

Об'єктом дослідження є передавання відеосигналу через інформаційні мережі.

Предметом дослідження є вплив обраних параметрів онлайн відеотрансляції та каналів передавання на якісні показники відеозображення. Методи дослідження

включають теоретичне аналіз наявних технологій та платформ для реального часу передавання відео та аудіо, порівняння їх ефективності, аналіз пропозицій щодо прямих трансляцій та проведення вебінарів як натурних експериментів.

Новизна отриманих результатів полягає у пропозиції використання спеціальних технологій для підвищення надійності передавання відеоконтенту через інформаційні мережі. Це дозволить покращити якість сприйняття аудіо та відеоконтенту користувачем та розширити можливості використання Інтернет-мовлення в освіті.

Також було запропоновано використання окремих платформ та програмних пакетів для покращення якості відеотрансляцій через інформаційні мережі у реальному часі.

Практичне значення отриманих результатів полягає в тому, що запропонований підхід до організації відеотрансляцій може бути використаний навчальними закладами та корпораціями для навчання та інформування співробітників про нові технології, організацію навчального та виробничого процесів тощо.

1 МЕТОДИ КОДУВАННЯ ДЛЯ ПРЯМИХ ВІДЕО ТРАНСЛЯЦІЙ

1. Класифікація потокового відео

Мультимедіа потоки бувають двох видів: за запитом (VoD) або живими (Live streaming). Потоки інформації, що викликаються на запит користувача зберігаються на серверах тривалий період. Живі потоки доступні короткий період часу, наприклад, при передачі відео зі спортивних змагань.[А.1]

1.1. VoD

Відео на вимогу (VoD) – це новітня технологія, що використовується в галузі зберігання та передачі інформації. За останнє десятиліття розвиток розваг приніс кілька позитивних змін, а однією з них є відео на вимогу. Ця технологія поєднує потреби користувачів, комп'ютерні мережі, зв'язок та відеотехнології. Відео на вимогу функціонує подібно до кабельного телебачення, де вміст передбачено заздалегідь і зберігається на серверах. Сервер може знаходитися в одному фізичному місці або розподілятися на кілька місць. Вміст буферизується або завантажується перед відтворенням.

Відео на вимогу не є настільки важливим у порівнянні з прямою трансляцією. Це означає, що відео може бути призупинене або відновлене під час його відтворення. Кожен користувач може вибирати програму, яку хоче переглянути, відповідно до своїх потреб. Це означає, що програми транслюються залежно від бажань користувача. Інший користувач може запитувати різні послуги. Також користувач може переглядати програми в зручний для нього час, будь то в будні дні чи вихідні. Крім того, відео на вимогу також має деякі спеціальні функції. Користувачі можуть перемотувати відео назад або вперед залежно від своїх потреб. Відео на вимогу дозволяє клієнтам завантажувати відео та шукати нові позиції відтворення згідно з їхніми інтересами. Так як ця послуга є новою, вона ще не має універсальної стандартизації. На рисунку 1.1 зображено простий приклад відеозапису на замовлення [1]. Після запису події у прямому ефірі з камери, відео кодується. Потім закодоване відео зберігається для майбутнього використання, і

через сервер потокового передавання воно постачається рівноправним користувачам А, В і С, які це запитали.

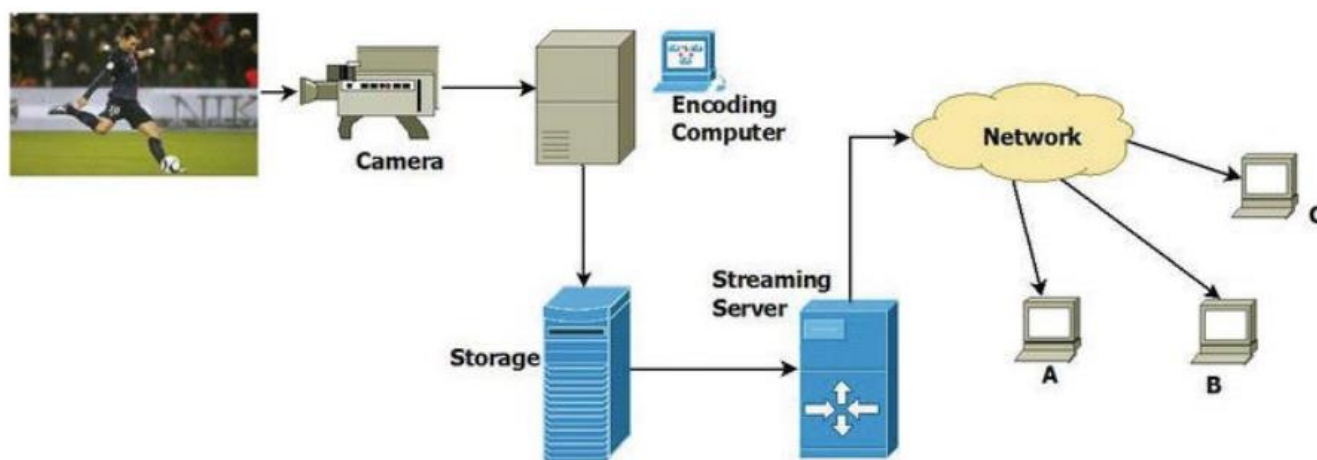


Рисунок 1.1 – Схема передавання потокового відео

1.2. Live streaming

Пряма трансляція – це платформа для перегляду відео в режимі реального часу. Вона дозволяє користувачам переглядати відеоматеріали через Інтернет з використанням аудіо/відеозв'язку. Прямий ефір є формою потокового передавання, яка дозволяє глядачам одночасно переглядати та транслювати відео за допомогою камери або аудіозапису, використовуючи Інтернет. Пряму трансляцію можна описати як трансляцію події в режимі реального часу, що відбувається в будь-якому місці через Інтернет. Оскільки неможливо бути одночасно присутнім на багатьох заходах, що відбуваються в різних місцях, пряма трансляція стає важливим засобом розваги та інформування, дозволяючи людям відстежувати події в реальному часі. Наприклад, популярним прикладом прямих трансляцій є матчі з футболу, що проводяться в різних місцях, і їх трансляція задовольняє потреби глядачів.

Передача в реальному часі відео потоку має важливе значення порівняно з відео на вимогу, оскільки вона враховує затримки. Користувач не може обирати програму у зручний для нього час, оскільки подія відбувається в прямому ефірі і передається всім користувачам. Тому користувачі не мають можливості перемотувати вперед або уповільнити подію в реальному часі. Розмір відеопакетів,

які надсилаються з сервера до глядачів, не визначений, що ускладнює пряму трансляцію порівняно з відео на вимогу. Передача відео є критично важливим аспектом у потоковому відео, оскільки кадри повинні бути доставлені до пункту призначення вчасно. Якщо кадри не досягають своєї мети вчасно, вони стають непотрібними. Тому необхідно перш за все передавати невідкладні кадри, щоб уникнути перерви у потоці відео. У відео на вимогу відео спочатку зберігається, а потім розподіляється відповідно до запиту користувача. Тому реалізація відео на вимогу менш складна, ніж пряма трансляція. На рисунку 1.2 показано простий приклад прямої трансляції [1]. Після зйомки події в прямому ефірі відбувається кодування, а потім закодоване відео передається на потоковий сервер. Потоковий сервер далі передає відео запитуючим клієнтам А, В і С, які хочуть переглянути трансляцію.

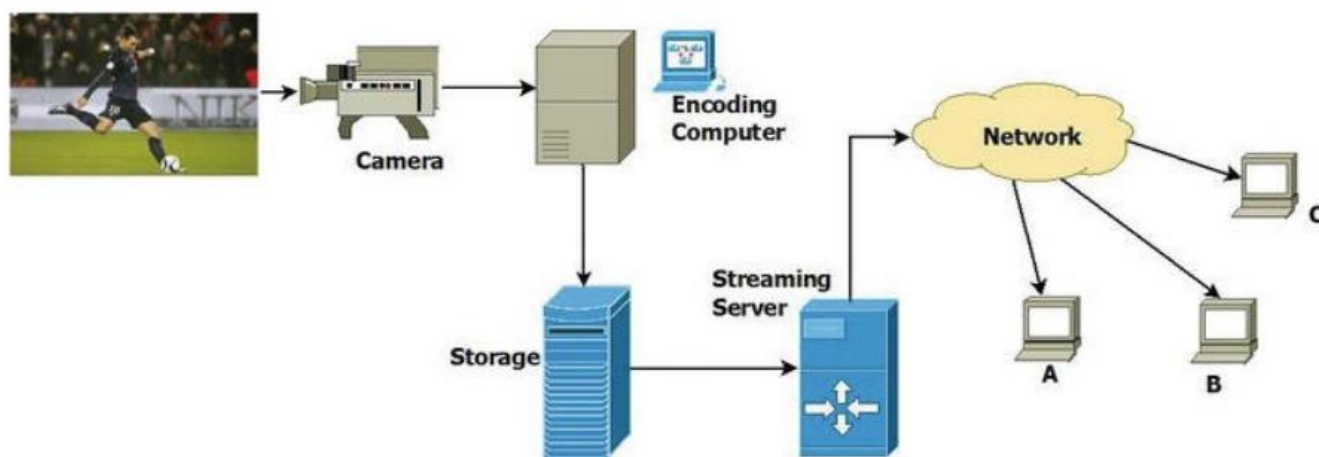


Рисунок 1.2 – Схема прямої трансляції

2. Архітектура

Ефективна робота будь-якої системи залежить від використання відповідної архітектури. Це також стосується систем потокового відео, для яких необхідна добре збалансована архітектура. Архітектури, що використовуються в таких системах, мають підтримувати широку буферизацію та безперервне передавання аудіо, відео та відповідних даних. Клієнтський сервер є популярним рішенням як для послуги відео по запиті (VoD), так і для прямого ефіру. Однак, з урахуванням різних параметрів масштабованості та якості обслуговування, ця архітектура може не відповідати вимогам. Тому дослідницьке співтовариство активно шукає кращі архітектурні рішення, зокрема, розподілену мережу вмісту (CDN) та мережу Peer to Peer (P2P), про які йтиметься далі.

2.1. Архітектура розподіленої мережі вмісту (CDN)

Архітектура CDN – це розподілена система, в якій дані розміщуються на різних серверах замість зберігання на оригінальному сервері. Ця система включає ієрархію серверів, розташованих у мережі [4]. Коли клієнт хоче взаємодіяти, він звертається до найближчого крайового сервера, а не до оригінального сервера. Якщо крайовий сервер вільний, він надає клієнту відповідь на основі доступних даних. При позитивній відповіді клієнт може переглядати відео в прямому ефірі. У разі негативної відповіді клієнт повинен спробувати інший сервер і повторити процес, поки не отримає позитивну відповідь. Сервер може обробляти кілька програм одночасно для кількох клієнтів в межах своїх можливостей. Для подолання проблеми флеш-навантаження у клієнт-серверній архітектурі розглядається новий підхід – кооперативна мережа (Coop Net). Однак, недоліком Coop Net є обмежений час роботи клієнта. Coop Net використовує розподілене потокове передавання та розподілене кодування для передачі потокового відео в прямому ефірі, що збільшує надійність системи. Управління на стороні клієнта мінімальне і просте завдяки централізованому серверному контролю. Розподілені мережеві ресурси забезпечують надійність [1]. Використання ресурсів клієнта мінімізується в

архітектурі клієнт-сервер. Проблема масштабованості контент-розподіленої мережі є основним завданням, оскільки збільшення мережі при збільшенні однорангових вузлів призводить до збільшення вартості на стороні сервера. Однорангова мережа привертає увагу дослідників через проблеми з масштабованістю, вартістю та використанням ресурсів.

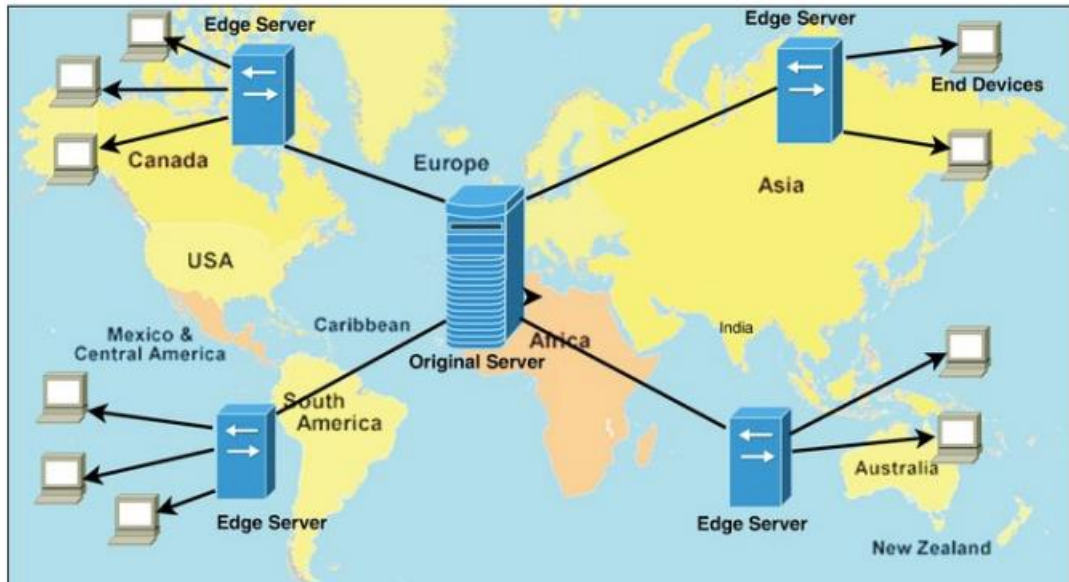


Рисунок 1.3 – Мережа розповсюдження вмісту

2.2. Peer-to-peer (P2P)

Однорангові обчислення (P2P) або обчислення в мережі – це розподілена архітектура, яка ділить завдання або призначення між рівноправними учасниками. Рівні учасники також отримують перевагу, будучи рівноцінними учасниками програми [5]. Основна увага мережі P2P полягає у справедливому розповсюдженні контенту без будь-яких виняткових умов від мережі. Метою мережі P2P є зниження накладних витрат із сервера та використання пропускної спроможності користувачів для завантаження.

На рисунку 1.4 показано деревоподібне накладення однорангової мережі з 13 одноранговими серверами та потоковим сервером. Рівні 1 та 2 отримують контент безпосередньо від сервера, тоді як інші рівноправні учасники отримують його від своїх безпосередніх батьків, а не безпосередньо від сервера. Таким чином, пропускна здатність рівноправних учасників також використовується для

подальшого завантаження контенту в мережу. Порівняння P2P та CDN наведено в таблиці.

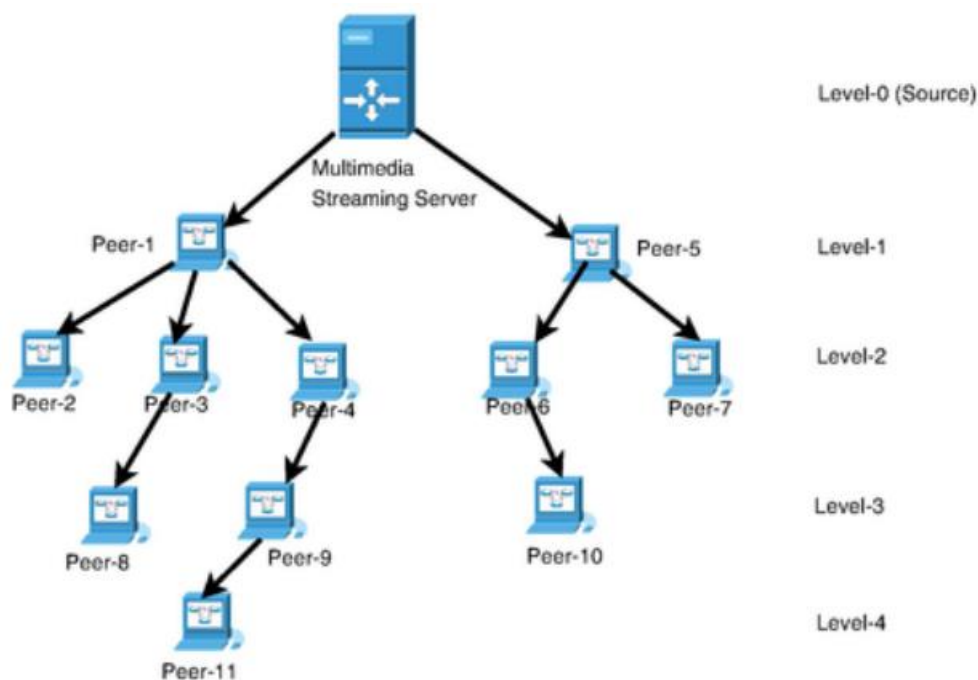


Рисунок 1.4 – Мережа P2P

Концепція робочої потокової системи P2P відображає стабільні відношення батьків-дочірніми однолітками з набору сусідніх рівних осіб.

2.3. Бітрейт

Існують різні режими кодування відеопотоку у форматі H.264, що можуть впливати на розмір вихідних потоків і ефективність роботи відеокомплексу. У даній статті пояснюється значення двох найпоширеніших режимів – CBR і ABR.

2.3.1. CBR кодування

CBR, або «постійний бітрейт», означає, що камера видає відеопотік з однаковою швидкістю, незалежно від інших параметрів, зі значним коливанням в межах $\pm 10\%$. Це значення встановлюється в налаштуваннях камери, а за замовчуванням різні виробники обмежують потік для CBR в районі 2-4 Мбіт/с для різних лінійок камер.

CBR є зручним, оскільки його легко використовувати для розрахунку дискового простору та вибору комутаторів [3]. Однак, важливо вирішити, яка

якість відео відповідатиме потребам користувача. Наприклад, 10 к/с з середнім розміром кадру в 100 КБ та 25 к/с з середнім розміром 40 КБ можуть мати однаковий потік. Вирішення про те, який варіант кращий, залежить від вимог до якості відео та того, якість буде задовольняти потреби користувача. Щоб визначити відповідний варіант для камери, слід звернути увагу на опцію пріоритету (Priority), пов'язану з режимом CBR. Залежно від вибраного пріоритету, можна встановити такі сценарії роботи:

Пріоритет швидкості (speed або rate). У випадку, коли з'являються ускладнення на зображенні (наприклад, люди переходять, автомобіль проїжджає, з'являються перешкоди або шуми через зниження освітленості), камера буде намагатися зберегти задану швидкість. При досягненні заданого потоку даних камера буде збільшувати ступінь стиснення зображення, що одночасно призведе до погіршення якості. Погіршення може бути досить серйозним, включаючи появу грубих артефактів та нерозбірливість зображення.

Пріоритет якості (quality). При ускладненні картинки, камера спрямовується на збереження заданої якості зображення, а кількість кадрів в секунду може зменшуватися, щоб не перевищувати заданий розмір CBR. Це подібно до режиму накопичення кадрів у аналоговій камері. Очевидний недолік полягає у ризику пропустити щось важливе через зниження FPS.

Рівний пріоритет (none) означає, що швидкість і якість є однаково важливими. Таким чином, при досягненні заданого порогу передачі даних обидва параметри будуть погіршуватися.

Як видно, необхідно йти на жертви. Або ж не робити їх? Якщо встановити великий потік даних, наприклад, 10 Мбіт/с, з запасом, це може призвести до надмірних витрат на мережеве обладнання (зв'язок, комутатори високої продуктивності) і відеосервери з великими дисковими сховищами, що мають високу вартість.

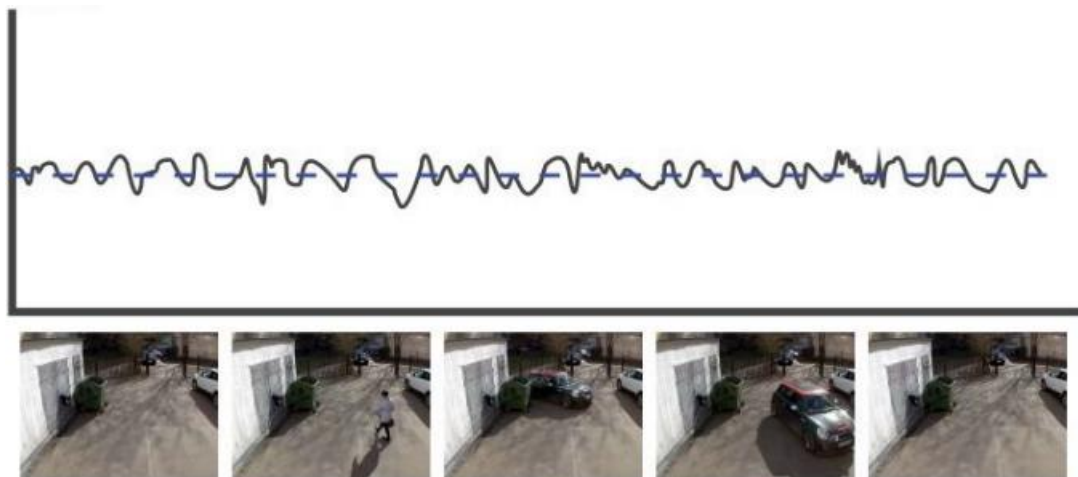


Рисунок 1.5 – Потік при кодуванні CBR

2.3.2. ABR кодування

Ще одним підходом до балансування бітової швидкості відео є технологія ABR (adaptive bit-rate), що зараз використовується багатьма сервісами VoD. У рамках ABR на рівні файлоховища відео для одного джерела (вихідного файлу) створюється множина відеопотоків різних роздільних здатностей, що мають різні рівні бітової швидкості.[A.1] ABR — це техніка потокового передавання, яка дозволяє відеопрогравачам адаптивно перемикатися між різними версіями потокового відео з бітрейтом залежно від умов мережі глядача. Регулюючи якість відео в реальному часі, ABR забезпечує плавне відтворення, мінімізуючи буферизацію та знижуючи ризик перерв.

На базі даних потоків утворюється сукупність I прошарків якості, які, у свою чергу, підлягають поділу на часові відрізки $t_{a,b}$. Для цих відрізків справедливою є умова:

$$t(1)_b = t(a)_b = t(A)_b \quad (1)$$

Під час передавання b -го відеопотоку, якщо складаються умови, коли $V(t) > B(t)$, виконується вибір $t(a-1)_b$ -го (наступного нижчого) прошарку якості за умови, що у поточний час транслюється $t(a)_b$ -й, більш високий за якістю та бітовою

швидкістю. У наслідок цього здійснюється ітераційний перехід до такого проширення якості, коли забезпечується виконання умови $V(t) \leq B(t)$.

До недоліків технології ABR можна віднести:

- відсутність алгоритмів контролю та гарантування якості відео;
- надмірне завантаження файлоховищ, що спричинюється необхідністю зберігання ряду відеофайлів різної роздільної здатності для кожного вихідного джерела;
- низька ефективність на випадок застосування для відеосервісів реального часу.

ABR — це ключова технологія потокового відео, яка пропонує значні переваги в доставці високоякісного відеовмісту через Інтернет. Дослідження підкреслює позитивний вплив ABR на якість відео, досвід користувача та використання пропускної здатності. Завдяки динамічному регулюванню бітрейту відео на основі умов мережі ABR забезпечує плавне відтворення, мінімізує буферизацію та забезпечує оптимальний перегляд. Впровадження ABR на платформах потокового відео має вирішальне значення для підвищення задоволеності користувачів і задоволення зростаючого попиту на високоякісний онлайн-відеоконтент.

2.4. Кодування відео

Кодування проходить у кілька етапів. Перший етап полягає у захопленні звичайного відео- чи телевізійного сигналу і перетворенні його в формат файлу, який може бути оброблений комп'ютерним програмним забезпеченням. Другий етап – зменшення швидкості передачі даних шляхом масштабування та стиснення до бітової швидкості, яка може передаватися через комутовані або широкосмугові схеми. Третій етап – упакування стиснутого відео у пакетний формат, який можна передавати через IP-мережу. Багато подій, що призвели до швидкого прийняття потокових медіа, відбуваються на другому етапі – стисненні. Не існує стандартного способу кодування; існує багато різних способів переходу від відео до потокового

формату. Вибраний вами шлях залежатиме від обраної апаратної платформи, необхідної пропускної здатності закодованого матеріалу та бажаної якості перегляду. Цей розділ надає опис деяких типових архітектур, але не є вичерпним посібником.

Для знаходження оптимального рішення для вашої програми, ви можете провести порівняльні тести між різними варіантами. Можна скласти власну систему з комп'ютерів, відеокарт та відповідного програмного забезпечення. Вам потрібна висока якість при низьких бітрейтах? Ви витрачаєте кілька годин на кодування щотижня, чи маєте великий архів для розміщення в Інтернеті? Усі ці фактори слід враховувати перед тим, як вирішувати, як кодувати. Загалом, ціна збільшуватиметься з підвищенням продуктивності та зручністю використання. Деякі висококласні програми надають автоматичне пакетне кодування.

2.5. Захоплення відео

Більшість стрімінгового мультимедійного контенту створюється у стандартному форматі телевізійних передач, прямо з камери або як відредагована програма на відеокасеті [2]. Трансляція у прямому ефірі – це специфічний випадок кодування, який ставить найвищі вимоги до обробного обладнання.

Перший крок полягає у перетворенні телевізійного сигналу в комп'ютерний файл, часто у форматі аудіо-відео з чергуванням (AVI). Якщо це пряма подія, запис відбувається на місці проведення. Зйомка на відеокасету, зазвичай, відбувається у центрі кодування. Також можна передавати файли безпосередньо з нелінійних систем редагування.

Перша проблема, з якою ви зіткнетесь при роботі з відео, полягає у тому, що існує багато різних форматів, які ви, можливо, захочете закодувати: аналогові або цифрові, компонентні або композитні.

Початковий матеріал навіть може бути у форматі високої чіткості, наприклад HD-CAM. Я не згадував лише аудіо-джерела. Знову ж таки, вони можуть бути аналоговими або цифровими, поставлятися на відеокасеті, відеокасеті або міні-диску, або записані на CD-ROM.

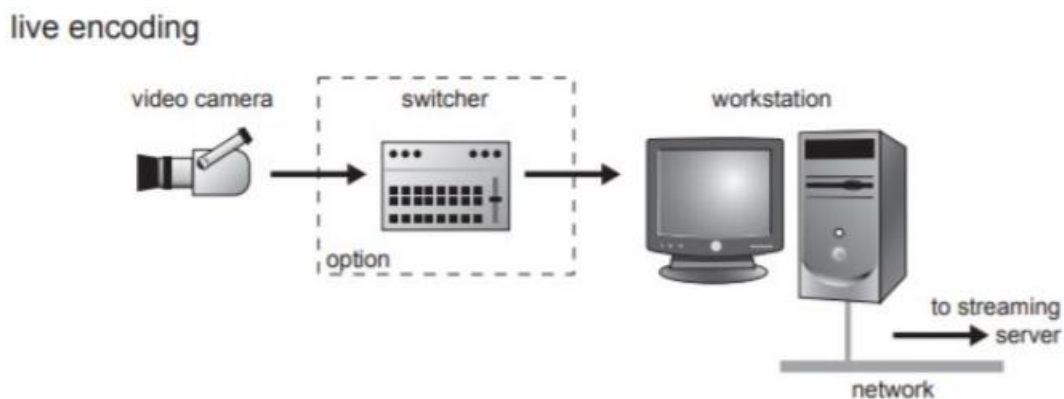


Рисунок 1.6 – Кодування в реальному часі

2.6. MPEG-1

MPEG-1 був розроблений на початку 1990-х років як перший стандарт для стиснення відео експертами Moving Pictures.

Компакт-диски, які працювали зі швидкістю приблизно 1 Мбіт / с, знайшли певну популярність в комп'ютерній мультимедіа, але ніколи не стали споживчим форматом для прокату фільмів або покупки. Незважаючи на це, MPEG-1 іноді використовується й сьогодні для деяких додатків, зокрема недорогих камер спостереження та деяких веб-відеопрограм.

MPEG-1 є підмножиною MPEG-2, тому будь-який декодер MPEG-2 може декодувати сигнал MPEG-1. Також варто зазначити, що MPEG-1 дозволений як метод стиснення відео для DVD-дисків, і багато DVD-програвачів відтворюють відео з компакт-дисків. Є доступні автономні та програмні кодери MPEG-1 за дуже розумними цінами з кількох джерел.

MPEG-1 не підтримує чергування відео або відео високої чіткості, тому його неможливо використовувати для відео PAL, NTSC, 720p або 1080i. Крім того, MPEG-1 не має деяких розширених методів кодування, які розроблені для систем MPEG-2 та MPEG-4. Ці обмеження зробили його легким у розробці програмних кодерів MPEG-1 та декодерів, які могли працювати на доступному кілька років тому апаратному забезпеченні ПК. Завдяки сучасним потужним машинам стало легше застосовувати MPEG-2 та MPEG-4 кодери та декодери, і обидві технології здатні створювати якісніші зображення з меншою пропускну здатністю. Як

результат, сьогодні системи MPEG-1 вважаються застарілими, і їх не варто розглядати як нові реалізації.

2.7. MPEG-2

MPEG-2 є одним з основних стандартів для відео в даний час. Він був затверджений у 1996 році і застосовується в різних програмах, таких як цифрове телебачення, високочітке телебачення, супутникове телебачення та кабельне телебачення. Кожного дня телекомпанії з усього світу записують, обробляють і відтворюють тисячі годин відео у форматі MPEG-2. Мільйони годин записів MPEG-2 щодня продаються у формі DVD зі стандартною роздільною здатністю.

Кількість пристроїв, які підтримують відтворення MPEG-2, постійно зростає. MPEG-2 також є стандартною системою стиснення відео для цифрового телебачення (DTV) у США. Згідно з законом, всі телевізори, які продаються в США, обов'язково мають тюнер DTV та відеодекодер MPEG-2.

MPEG-2 виявив значний прогрес у порівнянні з MPEG-1. Завдяки підтримці чергування, він може підтримувати стандартні сигнали NTSC та PAL з повною роздільною здатністю. Він також підтримує різні роздільні здатності та профілі продуктивності, що дозволяє використовувати різноманітні програми, включаючи HD-телевізори з роздільною здатністю 720p або 1080i. З мультиплексуванням відео- та аудіопотоків, MPEG-2 забезпечує можливість таких програм, як багатоканальне супутникове телебачення. Крім того, MPEG-2 підтримує п'ятиканальний звук та стандарт Advanced Audio Coding (AAC), які не були доступні в MPEG-1.

Одним з ключових понять у MPEG-2 є різні профілі та рівні, що потрібно розуміти в MPEG-2, включає різні профілі та рівні, доступні для кодування. Вибір правильного профілю та рівня для конкретної програми може значно вплинути на вартість системи. Використання вищих профілів та рівнів призводить до збільшення складності для кодерів та декодерів, може вимагати більшої пропускну здатності та значно підвищувати загальну вартість системи.

Існують чотири рівні роздільної здатності, впорядковані від найнижчого до найвищого. Ось короткі визначення кожного рівня:

- низький рівень відноситься до розміру зображення до 352 пікселів по ширині на 288 рядків, як у MPEG-1;

- основний рівень відноситься до основних розмірів зображення, що використовуються у стандартному телевізійному визначенні, тобто до 720 пікселів по ширині на 576 горизонтальних рядків для сигналів PAL. Розмір для сигналів NTSC зазвичай становить 720 пікселів по ширині на 480 рядків;

- високий рівень 1440 подвоює вертикальну та горизонтальну роздільну здатність основного профілю, тому пропонує 1440 пікселів по ширині на 1152 рядків;

- високий рівень розширює високий рівень 1440 до широкоформатного високого дозволу зі співвідношенням сторін 16:9 (замість 4:3 для звичайних основних сигналів профілю). Це збільшує максимальну кількість пікселів на рядок до 1920, але кількість рядків залишається на максимальному рівні 1152, достатньо для підтримки 1080-лінійного HD-сигналу.

MPEG-2 також підтримує безліч різних профілів продуктивності, які визначають типи методів, які може використовувати кодер для стиснення відео. Зі збільшенням профілю зростають вартість і складність як відеокодера, так і декодера. Однак, зазвичай підвищується якість відео, тому потрібно знайти компроміс. Шість профілів MPEG-2 впорядковані від най простішого до найскладнішого: простий, основний, 4:2:2, SNR, просторовий і високий [3].

Ось короткі означення кожного профілю:

- простий профіль дозволяє використовувати лише I та P-кадри, не включаючи B-кадри, що спрощує кодування та декодування. Цей профіль корисний для програм з низькою затримкою, оскільки він не вимагає обчислення B-кадрів;

- основний профіль призначений для різноманітних застосувань. Він підтримує всі рівні роздільної здатності і найчастіше використовується у більшості

програм. Зазначимо, що для цього профілю використовується колірний простір 4:2:0. DVD-диски відповідають специфікаціям цього профілю основного рівня;

- профіль 4:2:2 розроблений для підтримки обробки кольорів 4:2:2 без усіх додаткових функцій, які притаманні високому профілю. Зараз він використовується у виробництві відео, постпродукції та мережевій передачі внесків;

- профіль SNR (Signal-to-Noise Ratio) впроваджує концепцію масштабованих відеопотоків для одного відеосигналу: базового потоку, який містить більшу частину даних зображення, та допоміжного потоку, який може використовуватися для поліпшення продуктивності відео (зменшення рівня шуму та артефактів стиснення). MPEG-2 використовується у програмах телевізійного мовлення, але також знайшов застосування у програмах MPEG-4 для передачі відео через Інтернет;

- просторовий профіль також використовує концепцію масштабованого потоку, проте в цьому випадку основним потоком є сигнал SD (стандартна роздільна здатність).

Загалом, MPEG-2 залишається чітко визначеною та стабільною системою стиснення, яка широко використовується по всьому світу для створення відео професійної якості. Завдяки своїй гнучкості, потужності та великій кількості встановленого обладнання, він і надалі буде зберігати свою популярність як стандарт навіть у майбутньому.

Таблиця 1.1 – Зміст підтримуваних профілів і рівнів MPEG-2

Maximum Picture Resolution		Simple Profile	Main Profile	4:2:2 Profile	SNR Profile	Spatial Profile	High Profile
	High Level 1920 x 1152		4:2:0 80 Mbps				4:2:0 or 4:2:2 100 Mbps
	High 1140 Level 1140x1152		4:2:0 60 Mbps			4:2:0 60 Mbps	4:2:0 or 4:2:2 80 Mbps
	Main Level 1720x576	4:2:0 15 Mbps No B Frames	4:2:0 15 Mbps	4:2:2 50 Mbps	4:2:0 15 Mbps		4:2:0 or 4:2:2 20 Mbps
	High Level 352x288		4:2:0 4 Mbps		4:2:0 4 Mbps		
Profile Complexity							

2.8. MPEG-4

MPEG-4 є пізнішим продуктом процесу стандартизації, його перша версія була офіційно затверджена в 2000 році. Значне оновлення, відоме як H.264 та MPEG-4 Part 10 і MPEG4 AVC (усі синоніми), стало стандартом у 2003 році. Також MPEG-4 містить широкий спектр нових технологій стиснення відео, розроблених за останнє десятиліття. Нещодавно була розроблена вся необхідна допоміжна технологія для системи MPEG-4 (наприклад, напівпровідники), що дозволяє широко розгорнути MPEG-4. Як MPEG-4, так і H.264 мали значний вплив на стиснення відео в цілому та передачу IP-відео зокрема. Кожне з цих важливих досягнень буде обговорено в цьому розділі. І MPEG-4, і H.264 використовують переваги збільшеної обчислювальної потужності, доступної в напівпровідниковій промисловості за останнє десятиліття. Обидві розробки вимагають виконання значно більших обчислень як у кодері, так і в декодері. Разом ці інновації дозволили зменшити пропускну здатність на 50 відсотків для еквівалентної якості відео порівняно з MPEG-2 і запровадили нові програми, такі як доставка HD-відео через Інтернет і через диски Blu-ray і мережі IPTV. Технологія синтетичного зображення має дві основні переваги:

– кожному користувачеві можна надати контроль над елементами, що відображаються. Наприклад, якщо користувач не хоче бачити оцінку, можна вказати декодеру не відображати табло. Або, якщо користувач бажає, а монітор дозволяє, може перемістити табло на іншу частину екрана або змінити його зовнішній вигляд;

– синтетичні сигнали вимагають набагато меншої пропускнуєї спроможності порівняно з природними. Це пов'язано насамперед із вродженою складністю природних сигналів та необхідністю точного відтворення пікселів, що становлять природне зображення.

У MPEG-4 було включено кілька інших технологічних досягнень. Розмір кожного макроблоку більше не фіксований, і маленькі блоки можуть бути використані в деталях зображених областях зображення. Це також дозволяє використовувати великі блоки та кодувати їх меншою кількістю бітів, якщо пікселі в частинах зображення дуже схожі. Іншим нововведенням є використання фрактального стиснення, що корисно для деяких типів зображень. Нарешті, MPEG-4 дозволяє використовувати В-кадри на основі інших В-кадрів, що знижує необхідність Р-кадрах і загальну вимогливість до пропускнуєї здатності.

2.9. Розширений відеокодек H.264

MPEG4 Part 10, також відомий як MPEG-4 AVC або H.264, є потужним набором методів стиснення природних зображень[3]. Використання цих методів дозволяє досягти більш ефективного стиснення, що означає меншу швидкість передачі даних при збереженні однакової якості. Нові методи включають:

– використання кількох опорних кадрів для кожного стисненого кадру. Це дозволяє кодувати різні макроблоки на основі відмінностей вихідних зображень, що корисно, коли об'єкт переднього плану приховує та пізніше виявляє частину фону, яка вже була закодована в попередньому кадрі;

– використання просторового передбачення макроблоку на основі сусідніх блоків, що сприяє більш ефективному кодуванню великих, повторюваних областей;

– використання деблокуючого фільтра, який допомагає видалити гострі краї, що іноді з'являються на кордонах між сусідніми макроблоками. Згладжуючи ці різкі переходи, можна створити приємніше зображення з меншою швидкістю передачі даних;

– застосування зваженого передбачення, яке дозволяє застосовувати коефіцієнт масштабування до ряду макроблоків, спрощуючи процес кодування сцен з широкомасштабними змінами яскравості, наприклад, переходом до чорного.

– використання логарифмічних (а не просто лінійних) розмірів кроків для квантування, що спрощує управління швидкістю передачі даних в кодері для сцен з широким діапазоном яскравості;

– удосконалені методи арифметичного стиснення, такі як CAVLC (контекстно-адаптивне кодування змінної довжини) та CABAC (контекстно-адаптивне двійкове арифметичне кодування). Обидва ці методи вимагають більше обчислювальних зусиль, ніж прості методи кодування, які були введені в початковому розділі. Однак вони дозволяють досягти значно більш стисненого бітового потоку.

В H.264 використовується багато інших методів і технологій, які є численними і загадовими для повного перерахування тут. Загалом, швидкість передачі даних значно покращується порівняно з MPEG-2, але кодери та декодери стають складнішими. Оскільки неможливо реалізувати кожну функцію для кожного додатка, в H.264 були визначені різні профілі, які обслуговують різні програми. Нижче наведено частковий перелік визначених на даний момент профілів:

– базовий профіль: має найменший обсяг навантаження на обробку; не використовує В-кадри (також відомі як зрізи) та не використовує кодування CABAC. Цей профіль підходить для використання у відеоконференціях з прогресивним скануванням та програм для мобільного телебачення. Деякі компанії визначили додаткові види базового профілю, наприклад, «низька складність», але вони не включені до чинних стандартів;

– основний профіль: призначений для споживачів та трансляцій; він став дуже популярним у деяких додатках. Основний профіль додає В-кадри та кодування САВАС до базового профілю, що вимагає більше обробної потужності та пам'яті для кодування та декодування;

– розширений профіль: спрямований на потокові програми і має додаткові функції для відновлення втрачених даних та спрощення перемикання потоку. Цей профіль додає фрагменти SI та SP до основного профілю. Вони поліпшують продуктивність при перемиканні між потоками з низькою швидкістю передачі даних, дозволяючи генерувати зображення без необхідності очікувати I-зрізу. В цьому профілі не використовується кодування САВАС;

– високопрофільний: підтримує високоякісне зберігання та трансляцію відео для професійних додатків. Цей профіль став одним з основних форматів споживання завдяки підтримці HD на Blu-ray дисках. Існують різні розширені версії, які додають додаткові дані для кожного пікселя (наприклад, 10-бітна кольорова палітра 4:2:2 або 14-бітний вибірка кольорів), а також версії, які використовують внутрішню компресію кадрів для спрощення редагування.

У листопаді 2007 року до H.264 були додані масштабовані профілі на базовому, високому та високому внутрішньорівневому рівнях. Основною особливістю цих потоків є їх внутрішня структура, яка дозволяє декодеру обробляти лише частину поточкових даних для створення зображень зі зниженою роздільною здатністю, кадровою частотою або якістю порівняно з оригінальним потоком.

На момент написання цього тексту H.264 має 20 різних рівнів продуктивності, які можна порівняти з різними рівнями в MPEG-2 (низький, основний, високий 1440 та високий). Рівні продуктивності H.264 варіюються від 1 (64 до 256 Кбіт/с) до 6.2 (від 800 Мбіт/с). Це дуже багато рівнів для обговорення в цьому контексті.

Загалом, MPEG-4 є захоплюючою колекцією технологій, яка обіцяє значно збільшити обсяг відеоінформації, яку можна стиснути в заданому обмеженні

пропускної здатності мережі. Завдяки MPEG-4 AVC стало можливим набагато ефективніше кодування відео, а різноманітність доступних типів об'єктів робить інтеграцію з комп'ютерною графікою простою і надзвичайно ефективною з точки зору пропускної здатності.

2.10. Стиснення звуку

Так само, як відеокомпресія, MPEG має багато варіантів аудіокомпресії. Існують три рівні MPEG звуку (позначені як Рівень I, II і III), а також дві версії нового стандарту аудіокомпресії під назвою Advanced Audio Coding (AAC) [3]. В цьому розділі ми коротко розглянемо кожен з них. Варто зазначити, що всі ці методи аудіокомпресії працюють з будь-яким типом відеокомпресії MPEG, за винятком потоків MPEG-1, які не підтримують AAC аудіо.

Перед тим, як перейти до відмінностей між цими варіантами, давайте розглянемо їх спільні характеристики. Всі MPEG аудіокодеки втрачають інформацію, що міститься в звуковому сигналі, тобто вони є «збитковими». Крім того, всі вони є «перцептивними кодеками», що означає, що компресія здійснюється на основі того, що людська слухова система може або не може сприйняти. Також всі MPEG кодеки використовують цифрові аудіосигнали з однією з трьох можливих частот дискретизації: 32 кГц, 44,1 кГц або 48 кГц. Крім того, кожен з декодерів (крім AAC) має правильно працювати з потоками, які були вже закодовані для нижчих рівнів, тобто декодер рівня III повинен приймати потік рівня I або II.

Перцептивне аудіокодування може значно зменшити розмір файлу. Вухом насправді більш чутливе до помітних змін, спричинених компресією, ніж око. Однак, оскільки вухо не є ідеальним, деяка інформація може бути відхилена або несприйнята звуком. Наприклад, вухо не сприймає сигнали, які тривають менше мілісекунди (мс). Гучний сигнал може заглушити або приховати більш тихий сигнал, що безпосередньо передує йому або йде за ним. Також існують обмеження частотного діапазону, який вухо може сприйняти. Крім того, гучний сигнал на одній частоті буде заглушати більш тихі звуки на інших частотах, що знаходяться

поруч, та вони будуть звучати значно тихіше на віддалених частотах (тому ви, наприклад, не можете почути шум «шипіння» на аудіокасетах, за винятком тихих періодів). Всі ці обмеження аудіосистеми були враховані при розробці методів аудіокомпресії MPEG.

MPEG audio Layer I – це найпростіша система стиснення, в якій використовуються 384 вхідні вибірки на кожен цикл стиснення. Це відповідає 8 мілісекундам аудіоматеріалу з дискретизацією 48 кГц. Layer I забезпечує постійну швидкість передачі бітів при ступені стиснення 4:1. Це означає, що аудіосигнал CD-якості зі швидкістю 1,4 Мбіт/с може бути стиснутий до потоку 384 Кбіт/с без помітної втрати якості. Стиснення при швидкостях 192 або 128 Кбіт/с призводить до помітної втрати якості.

MPEG audio Layer II використовує 1152 зразки на кожен цикл стиснення, що відповідає 24мсек аудіо з дискретизацією 48 кГц. Це дозволяє більш точно визначати частоти. Рівень II також зменшує деяку надмірність рівня I, досягаючи стиснення до 8:1. Таким чином, звук якості CD може бути досягнутий зі швидкістю потоку 192 Кбіт/с.

MPEG audio Layer III використовує таку ж кількість зразків, як рівень II, але використовує їх ефективніше. Layer III має режим спільного стерео, який використовує сильну схожість між лівим та правим каналами стереопрограми.

Також використовується кодування змінної довжини для упаковки стисненого звуку, щоб коефіцієнти були ефективніше передані в потік. Як результат, кодери рівня III можуть упаковувати звук якості CD в невеликі потоки 128 Кбіт/с, досягаючи ступеня стиснення до 12:1. Зазначимо, що аудіофайли, стиснуті за допомогою MPEG Layer III, часто мають розширення .mp3 і є популярними для завантаження музики, обміну файлами та використання на портативних програвачах.

AAC, або MPEG Advanced Audio Coding, доступний лише для відеопотоків MPEG-2 або MPEG-4. Він підтримує до 48 аудіоканалів, включаючи 5.1 звук, і містить безліч інструментів, які можуть використовуватися складним кодером для

створення високоякісного аудіопотоку. Один з варіантів – це режим стиснення без втрат, який підходить для додатків, що не потребують абсолютно високого стиснення, але вимагають високої точності. Для потоку MPEG-4 можуть використовуватися різні аудіооб'єкти, подібні до відео MPEG-4. Ці об'єкти можуть бути природними або синтетичними. Аудіодекодер збирає всі аудіооб'єкти і видає остаточний комбінований вихід. Сигнали AAC часто використовуються для потокового передавання аудіо та відео програм і підтримуються на портативних пристроях, таких як iPod від Apple.

Загалом, аудіо MPEG є гнучким і не вимагає майже такої ж потужності обробки, як відео MPEG. Зі зростанням рівня складності як кодера, так і декодера, зростає ступінь стиску (таблиця 1.2), який узагальнює та порівнює основні особливості різних рівнів звуку в цьому розділі. Програмні декодери рівня III можуть працювати безперебійно на широкому спектрі персональних комп'ютерів, включаючи настільні та портативні системи. Декодери AAC також стали дуже популярними, враховуючи підтримку від Apple. При виборі методу аудіокодування варто пам'ятати, що загальна пропускна здатність транспорту повинна бути достатньо високою, щоб переносити відеосигнал, звуковий сигнал та деякі накладні витрати, щоб потоки працювали належним чином. Використання методів аудіокодування з вищим ступенем стиснення дозволить збільшити пропускну здатність для передачі відеосигналів без втрати якості звуку.

Узагальнюючи, AAC є гнучким аудіоформатом, який підтримує до 48 аудіоканалів і надає високу якість звуку. Він може бути використаний як для стиснення з втратами, так і для стиснення без втрат в залежності від вимог застосування. Декодери AAC широко підтримуються і доступні на різних пристроях, включаючи портативні пристрої від Apple. При виборі методу кодування аудіо необхідно забезпечити достатню пропускну здатність для передачі як звуку, так і відео, забезпечуючи належну роботу потоків і збереження якості сигналу.

Таблиця 2.2 – Порівняння стандартів аудіо MPEG

Compression Method	Samples per Block	Compression Ratio	CD-Quality Bit Rate (kbps)
Layer 1	384	4:1	384
Layer 2	1152	8:1	192
Layer 3	1152	12:1	128
AAC	1152	16:1	96
HE AACv2	2048	32:1	48

Висновки до розділу

При передачі відеосигналів через IP-мережу зазвичай застосовують стиснення, щоб зменшити кількість бітів, необхідних для представлення відео зображення. Користувачі можуть самостійно вибирати, чи використовувати стиснення для своїх відеосигналів, але потрібно розуміти, що вибір методу стиснення може вплинути на успішність мережевого проекту.

У цьому розділі описано процеси захоплення та стиснення відео. Процес кодування може бути простим, але необхідність кодування для різних форматів та розширень може зробити його складним. Інструменти, такі як Discreet Cleaner або Anystream Agility, можуть допомогти скоротити частину цієї складності. Вибір параметрів для сеансу кодування є компромісом між якістю та швидкістю передачі даних. Вибір кодеків залежить від різних факторів, таких як архітектура, яку використовуєте, необхідна якість відео, затримка обробки, ресурси центрального процесора і вартість. Кодування тісно пов'язане з передвідборкою відеоматеріалу, що дозволяє очистити сигнал перед стисненням, видаляючи артефакти, такі як шум. Шум знижує пропускну здатність даних, що може ускладнити стиснення, тому попередній обробки є необхідним етапом перед кодуванням.

Кодеки стиснення, такі як MPEG-1, MPEG-2 та MPEG-4, є необхідними для потокової передачі відеоінформації. Існує багато ефективних систем, які можуть конкурувати зі старими форматами розповсюдження відео. MPEG-4 є продовженням розробки та має ряд профілів кодування, які підходять для різних застосувань, включаючи мобільні пристрої з низькою швидкістю передачі даних і

бездротове відео високої чіткості. Завдяки цим кодекам можлива передача великого обсягу інформації в ефективному форматі.

Таким чином, кодеки стиснення є необхідними компонентами для передачі відео через IP-мережу. Вони дозволяють зменшити розмір файлу і забезпечують оптимальну якість передачі відео, залежно від вимог і обмежень проекту. Особливо важливим є вибір правильного кодеку і параметрів стиснення, оскільки це може вплинути на успіх і ефективність відеомережі.

2 ТЕХНОЛОГІЇ ОРГАНІЗАЦІЇ ВІДЕОТРАСЛЯЦІЙ

1. Адаптивна мультимедійна трансляція

Модель потокового мультимедіа може бути розглянута як модель відповіді на запит, при цьому попит кожного клієнта не є постійним, і мережева конфігурація може динамічно змінюватися. Тому статичний механізм попиту не здатний вирішити проблему потокової передачі мультимедіа. Адаптивний характер є важливим аспектом мультимедійної потокової моделі, існують різні методи адаптації, які будуть детальніше розглянуті нижче.

1.1. Адаптивне планування

Стратегія планування змінюється залежно від зміни конфігурації мережі. Таким чином, одноранговий зв'язок послідовно вибирає інший блок залежно від поточної конфігурації мережі або поточного попиту однорангової мережі. У цьому розділі розглядаються різні адаптивні параметри, які можуть впливати на стратегію планування, а також різні схеми адаптивного планування.

Пасивні параметри адаптації – це параметри, що заздалегідь визначені і не змінюються в мережі для певного рівня. У той же самий час активні параметри адаптації – це параметри, що змінюються зі зміною стану мережі. Для однорангових мереж активні параметри адаптації є динамічними, тоді як пасивні параметри адаптації мають статичний характер. Активна адаптація є динамічною і має вирішальне значення для схеми планування, при цьому різні параметри, такі як однорангова ситуація, затори та інші, впливають на планування.

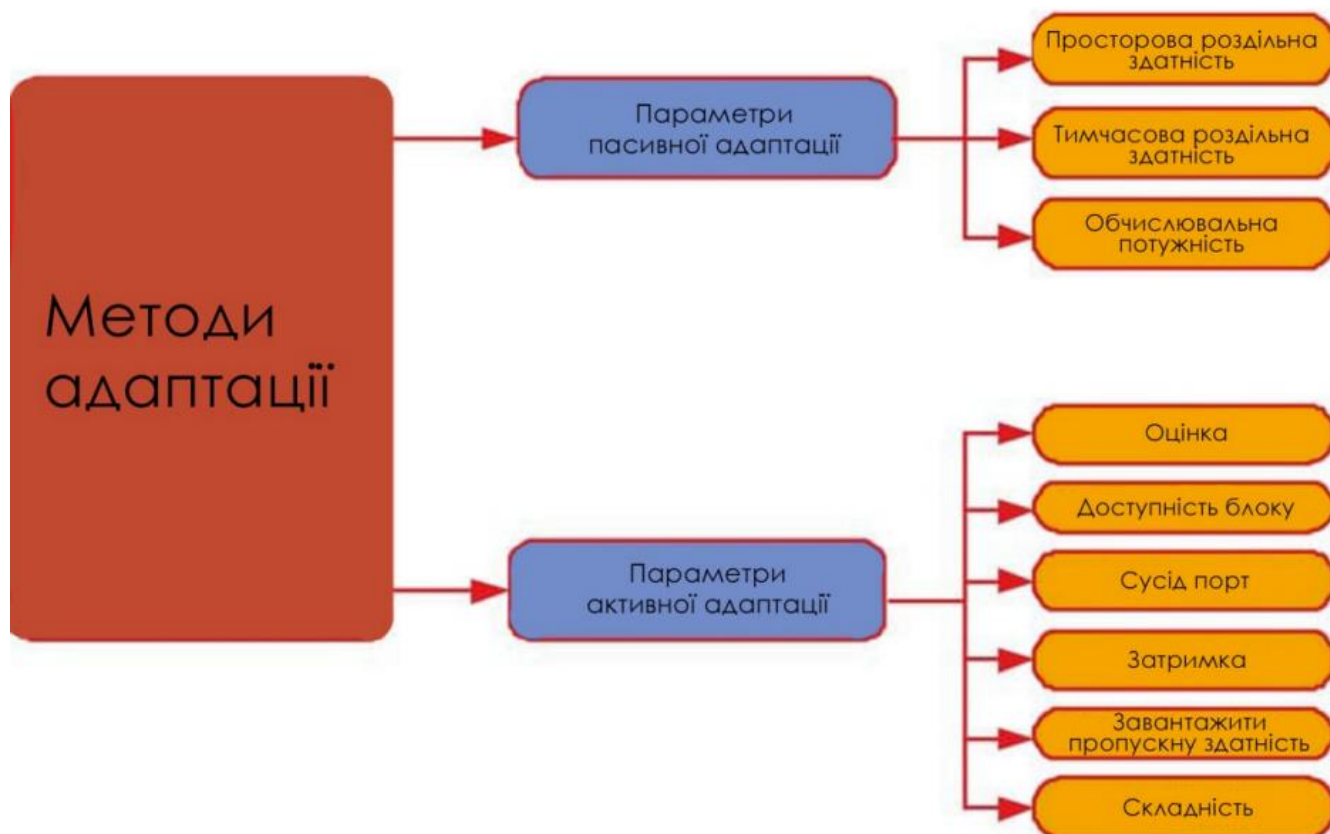


Рисунок 2.1 – Техніка адаптації

Неправильне регулювання ресурсів може погіршити відносини між відправником та одержувачем. Також впливає на продуктивність мережі недостатнє розуміння її функціонування. У статті Чжен та ін. [6] описується адаптивний підхід, який використовує схему планування на основі натискання для вирішення проблеми нестабільних зв'язків між попитом та пропозицією [6]. Цей підхід базується на прогнозуванні. Система передбачає використання ресурсів, тобто визначає, які ресурси можуть стати недостатніми, щоб забезпечити затребування в ранніх етапах, і забезпечує плавне планування. Адаптивний метод використовується для вибору рівнозначних партнерів та ресурсів "push". Рівнозначні партнери вибираються залежно від мережевих умов, а найкращі з них обираються як однолітки.

Головним фактором, що впливає на потокову передачу відео, є неоднорідність мережі та кінцевих пристроїв. Вона виявляється у різних проблемах, таких як роздільна здатність та розмір екрану, затримка, змінна

пропускна здатність завантаження та можливості обробки. З метою поліпшення потокової передачі відео для кінцевих користувачів автор пропонує адаптивний підхід. Цей підхід використовує масштабоване кодування відео (SVC) для забезпечення якісної адаптивної передачі. SVC використовує техніку кодування у вигляді різних шарів, таких як базовий шар і декілька шарів покращення. Вибір шару здійснюється двоетапним механізмом, адаптація якості забезпечується шляхом вибору відповідного шару. Перший крок – це початкова адаптація якості (IQA), а другий крок – прогресивна адаптація якості (PQA).

На рисунку 2.2 зображена архітектура потокового передавання, яка використовує якісну адаптацію.

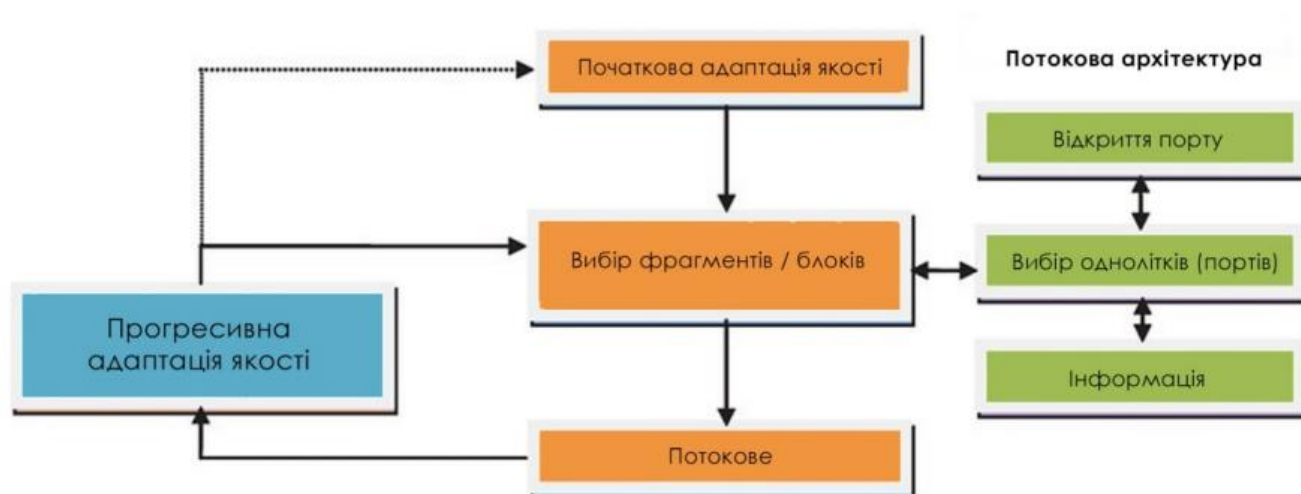


Рисунок 2.2 – Якісна адаптивна архітектура потокового P2P

Початкове адаптування якості (IQA): Як можна зрозуміти з назви модуля, це вихідне рішення щодо налаштування якості. Параметри, що впливають на рішення в IQA, включають роздільну здатність екрана, доступну пропускну здатність та обчислювальну потужність. Початковий набір якості позначається як (QS0). Значення QS0 визначається за формулою (2).

$$QS0\Delta = \{(d, t, q) : \forall d = 0 \dots D, t = 0 \dots T, q = 0 \dots Q\} \quad (2),$$

де d , t та q – загальна кількість шарів у просторовому, часовому та SNR вимірах.

Прогресивна адаптація якості (PQA) – це модуль, що регулярно виконується з метою покращення якості відео відповідно до змін у мережі. Він враховує два основні фактори у реальному часі: пропускну здатність та доступність блоків. Крім того, він враховує потужність обробки для забезпечення рівних ресурсів. За цими параметрами він обчислює адаптацію мережевого стану, адаптацію бітрейту та адаптацію складності. На завершення він надає остаточний результат адаптації [7].

Динамічна потреба однолітків задовольняється за допомогою параметрів, які є життєво важливими для кінцевого користувача. Цей підхід забезпечує результати, враховуючи динамічний ефект природи, і тому надає кращі адаптивні результати щодо доступності блоку та часу відповіді. Проблема планування вирішується шляхом пошуку рівнозначного відправника та ресурсу. У випадку мережі P2P може бути багато різних шляхів між джерелом і призначеним місцем. Тому пошук найкращого шляху між відправником та одержувачем також має велике значення. Для цього використовуються підходи, що використовують генетичний алгоритм [7]. Ці методи включають процедури мутації та кросоверу з невеликою популяцією. Ровесники отримують інформацію про фізичний шлях, яка використовується як придатна функція для прийняття рішень. Для обчислення інформації про фізичний шлях використовується команда Traceroute, яка динамічно визначає зміни в мережі. Адаптивний підхід забезпечує вибір найкращого доступного шляху між відправником та одержувачем. Використання такого підходу до вибору шляху позитивно впливає на продуктивність мережі щодо затримки. На рисунку 8 показано класифікацію методів адаптації за двома параметрами, на яких вони базуються. З-поміж цих різних параметрів, адаптивний вибір однолітків та вибір адаптивного темпу мають значний вплив на мережу P2P, тому ці два механізми адаптації детально обговорюються.

Для забезпечення повного використання пропускну спроможності доступних у конкретному регіоні однорангових станцій E. Караєр пропонує підхід до планування багатопокілних пакетів [7]. Створюються різні класи відповідно до бажаної якості обслуговування. Два наступні підходи до планування розроблені

таким чином, що перший підхід сконцентрований в основному на вирішенні однієї проблеми, що генерується, в той час як другий підхід фокусується на динамічному програмуванні. Рішення досягаються з використанням псевдополіноміального часу. Поліпшена ефективність пропускної спроможності досягається та перевіряється на основі результатів моделювання. Інша схема, яка фокусується на виборі декількох фрагментів та однорангових елементів, визначена в. Основний підхід полягає у забезпеченні корисності однорангових елементів та фрагментів відповідно до ймовірності відтворення кожного фрагмента та швидкості потокової передачі. Доступна пропускна спроможність кожного однорангового з'єднання в мережі грає вирішальну роль для схеми планування.

1.2. Адаптивний вибір швидкості

Для покращення мережі P2P недостатньо лише адаптивного вибору однолітків. Тому в новій схемі планування поєднуються адаптація накладання і адаптація швидкості, як описано. Регулятор швидкості застосовує адаптацію ставки на основі місцевої інформації в цій схемі. Цей підхід ефективний для побудови накладання і планування в мережах P2P, але не враховує такі властивості, як неоднорідність і затримка мережі. Фіксований канал використовує постійну швидкість передачі даних через мережу P2P. Висока швидкість передачі даних збільшує ймовірність втрати даних через обмежену пропускну здатність між пристроями на одному рівні. З іншого боку, низька швидкість передачі даних може призвести до зниження якості відео. Таким чином, швидкість передачі даних безпосередньо впливає на продуктивність потокового відео в мережі P2P. Для вирішення цієї проблеми розглядається підхід самоадаптивної мульти-ставки [1].

Архітектура підходу MoSee включає три основні компоненти: Tracker (сервер реєстрації), Broadcaster (сервер мовлення) та клієнт (Споживач даних). Клієнт може бути дротовим або бездротовим і відповідає за передачу даних згідно з вимогами клієнта. Він також реєструє канал, створений для клієнта, за допомогою трекера. Клієнт реєструється на сервері реєстрації та отримує список програм. Після вибору програми клієнт вибирає канал і приєднується до нього. У MoSee канал

розподіляється на підпотоки, кожен з яких має унікальний індекс. Залежно від стану мережі та буфера, клієнт може обрати будь-який підпотік.

Підпотік можна також адаптивно змінювати залежно від змін в мережі. На початку, коли мережа є ненадійною, клієнт обирає підпотік зі сповільненою швидкістю передачі, але з часом, коли мережа стає стабільною, клієнт обирає підпотік з високою швидкістю передачі даних. Концепція кількох підпотоків використовується для зменшення обміну даними між клієнтами, які мають різні підпотоки даних. Для підвищення рівня обміну даними між такими клієнтами використовується інший підхід. Коли клієнт з високою пропускнуою здатністю може передати дані з високою швидкістю, він також передає дані зі зниженою швидкістю для слабших клієнтів. Коли слабкому клієнтові потрібен підпотік зі зниженою швидкістю передачі, клієнт з високою швидкістю може надати йому такий підпотік. Для вибору сусіднього клієнта використовується значення часу зворотного зв'язку (RTT), при цьому різним клієнтам надаються різні ваги, і потім робиться остаточний вибір.

Різні складнощі можуть виникати в мережі P2P через різні фактори, такі як нерівномірність пропускнуої здатності смуги, великі затримки передачі та витік інформації з однорангових мереж. Однак, є перспективний підхід, який вирішує всі ці проблеми – це адаптована потокова передача відео в реальному часі. Для створення накладання між одноранговими мережами в мережі P2P використовується підхід накладання на багато дерев. За цим підходом, відео надсилається кількома однолітками відповідно до мережевих умов та доступної пропускнуої здатності. Залежно від стану мережі, потоковий сервер перемикає бітовий потік між безліччю доступних потоків. Щоб уникнути перевантаження мережі, застосовується схема формування швидкості, яка знижує швидкість передачі потоку. Формування швидкості в розподіленому багат шаровому накладанні включає два етапи. Перший етап полягає в обчисленні виділеної смуги пропускання шляхом створення списку отриманих та переданих пакетів. Залежно від залишкової пропускнуої здатності, найбільш відповідний пакет зі списку

надсилається. Другий етап включає застосування оптимізованого планувальника, який диференціює пакети залежно від їх впливу на ефективність мережі. Хоча цей підхід сприйнятливий для застосування на стороні сервера, в мережі P2P він також враховує фактичний попит між однолітками. Метод управління швидкістю відтворення відео обговорювався у двох різних випадках – номінальному та надійному. Для розрахунку швидкості використовується залишкова пропускна здатність завантажених пірів у мережі. Цей метод динамічно перевіряє залишкову пропускну здатність однорангових мереж, що активні в даний момент в мережі. Для управління швидкістю відтворення була розроблена теоретична модель та запропоновані відповідні рішення.

Задоволення запиту приймача вимагає вищої швидкості, і для цього модель черги може бути корисною. Якщо належна черга підтримується і забезпечується належний вхід в чергу, тоді однорангові вузли можуть отримувати відповідні швидкості. Для адаптивного планування розглядаються різні моделі масового обслуговування, такі як однорангова та серверна сторони. У черзі однорангової мережі дані, що надходять, класифікуються на два класи: F (пересилання) та NF (непересилання). Клас пересилання включає дані, які мають бути передані іншим одноранговим вузлам, тоді як клас непересилання містить дані, що призначені для власного вузла і не потребують подальшої передачі. Таким чином, немає необхідності передавати дані іншим вузлам.

2. Протоколи транспортування

2.1. HyperText Transport Protocol (HTTP)

HTTP був розроблений з метою доставки файлів HTML, і його використання поширене для зв'язку між браузером та веб-сервером у розподілених, спільних, гіпермедіа-інформаційних системах. Взаємодії включають запити та відповіді, і заголовки містять мету запиту. Відповіді містять інформацію, подібну до MIME, для опису формату передаваних даних.

Крім того, HTTP використовується як загальний протокол для зв'язку між користувачами і проксі-серверами або шлюзами до інших Інтернет-систем, включаючи ті, що підтримують протоколи RTSP, SMTP та FTP. Таким чином, HTTP надає базовий рівень доступу до різноманітних програм з мультимедійним вмістом.

Цей протокол є важливим для потокового передавання, оскільки використовується веб-сторінками, що посилаються на вміст, і часто є єдиним вибором для комунікаційного протоколу в простих мережевих середовищах. Іноді брандмауери компаній блокують доступ до потокових мультимедіа через протоколи реального часу, тому HTTP стає єдиним варіантом доставки медіапотоків. Цей протокол працює на прикладному рівні і зазвичай використовує з'єднання TCP/IP.

Це може вважатися недоліком HTTP, але він також має значні переваги. Тому що протокол не має стану і не зберігає зв'язку, веб-сервер не має накладних витрат на підтримку з'єднання з браузером. Ця характеристика дозволяє серверу ефективно обробляти багато користувачів одночасно.

2.2. UDP

Teradek є провідною компанією на ринку обладнання для прямих трансляцій. Сьогодні Teradek пропонує широкий спектр обладнання для передачі відеосигналів, призначеного для прямого мовлення, виробництва відео, інтернет-мовлення та виробництва фільмів. Протокол дейтаграм користувача (UDP) є беззв'язним транспортним механізмом, який може підтримувати високошвидкісні потоки даних, такі як цифрове відео та інші типи передачі даних [3]. Він часто використовується, коли накладні витрати на встановлення з'єднання (як це робить TCP) не потрібні. Наприклад, UDP часто використовується для трансляції повідомлень з одного пристрою на всі інші пристрої в сегменті мережі (наприклад, сервер друку сповіщає всіх користувачів про відсутність паперу в принтері).

UDP — це протокол без з'єднання, що означає відсутність механізму встановлення зв'язку між хостом-відправником і хостом-одержувачем. Відправник

UDP просто форматує дейтаграми з правильним сокетом призначення (IP-адресою та номером порту) і передає їх на IP для транспортування. Звичайно, це також означає, що немає координації між відправником даних UDP і приймачем даних UDP для забезпечення належної передачі даних.

На перший погляд може здатися, що такий брак координації робить UDP непридатним для передачі відеоданих, оскільки відсутність відеоданих може перешкодити здатності приймача відображати правильну послідовність повних кадрів. Однак важливо пам'ятати, що кожен кадр у відеопотоці NTSC відображається лише протягом 33 мілісекунд (40 мілісекунд для відео PAL). Якщо частина даних для кадру відсутня, ідеальний приймач повинен:

- а) визначити відсутність даних;
- б) надіслати повідомлення відправнику, щоб запитати необхідні дані для повторної передачі;
- в) отримувати та обробляти повторно передані дані;
- г) переконатися, що все це відбувається до появи відображеного зображення.

З метою полегшення завершення всіх цих кроків у вікні 33 мсек, може бути розумним уникати повторної передачі відсутніх даних. Багато форматів відеопотоків мають механізми виявлення та виправлення помилок, наприклад, транспортні потоки MPEG включають байти для прямої корекції помилок Ріда-Соломона. Декодер MPEG може виправити помилки бітів за допомогою цих байтів і іноді він може відновити втрачені пакети, що унеможлиблює необхідність повторної передачі цих пакетів. З урахуванням цих можливостей, UDP є логічним вибором для транспортного протоколу, оскільки він не додає непотрібну накладну витрату на потоки, які вже мають вбудовані функції виправлення помилок.

Переваги UDP:

- дуже низькі накладні витрати: заголовки пакетів для UDP є одними з найменших серед усіх протоколів, що використовують IP;
- проста схема адресації портів: Ця схема дозволяє будь-яким даним, що надходять до певного сокета, переадресовуватися безпосередньо до програми,

прикріпленої до цього сокета. Це також усуває необхідність використання транспортного протоколу (у цьому випадку UDP) для відстеження декількох різних підключень до одного порту;

- швидкий час налаштування: в UDP хост-відправник не повинен встановлювати з'єднання перед початком передачі даних. Це дозволяє відправнику розпочати передачу, не чекаючи встановлення з'єднання;

- низькі затримки: UDP не буферизує дані для очікування підтвердження, тому затримки через очікування підтверджень усуваються;

- гнучка архітектура: оскільки UDP не вимагає двостороннього зв'язку, він може працювати в односторонніх мережах, наприклад, супутникових трансляціях.

Крім того, UDP може використовуватися для передачі даних до багатьох призначень програм, коли одне джерело надсилає інформацію до декількох точок призначення. Широка доступність: UDP підтримується більшістю обладнання, що використовується в IP-мережах.

Недоліки UDP:

- ненадійний транспорт: UDP не має вбудованого механізму для автоматичного повторення пошкоджених або втрачених даних;

- блокування брандмауером: Деякі адміністратори брандмауерів можуть конфігурувати свої системи так, щоб блокувати UDP-трафік, через можливість введення хакером зловмисних пакетів даних;

- відсутність «зворотного каналу» зв'язку між відправником і отримувачем: Це означає, що відправник не отримує автоматичного зворотного зв'язку від отримувача;

- відсутність автоматичного регулювання швидкості: Відправник несе відповідальність за контроль швидкості вихідного потоку даних, щоб не перевищувати пропускну здатність транспортного зв'язку.

Загалом, UDP є простим транспортним протоколом з низькими накладними витратами, що може бути корисним для передачі відеоданих. Він має довгу історію, і більшість пристроїв, що підтримують IP, також підтримують UDP. UDP може

бути хорошим вибором для передачі відео, коли відеопотоки мають свою систему корекції помилок, маршрутизація даних контрольована або обмежена, і важлива низька затримка для мінімізації часу передачі.

2.3. TCP

TCP – це орієнтований на з'єднання протокол передачі даних, який надає надійні послуги передачі. Він є найпоширенішим протоколом в Інтернеті та інтрамережах. З'єднання в TCP вимагає встановлення зв'язку між відправником та одержувачем перед передачею даних. З'єднання може ініціюватися будь-яким з кінців, але обидва кінці повинні підтвердити з'єднання, перш ніж почнеться передача даних.

Переваги TCP:

- автоматична повторна передача даних: TCP має вбудовані механізми для автоматичної повторної передачі даних, що забезпечують доставку кожного байта даних відправнику до одержувача. Якщо дані були втрачені або пошкоджені під час передачі, TCP повторно надішле ці дані;
- порядкові номери байтів даних: Порядкові номери дозволяють отримувачу визначити, чи отримані дані, чи відсутні пакети, чи ж пакети були отримані в зміненому порядку;
- багато з'єднань до одного порту: TCP використовує пари сокетів для відстеження повідомлень, що дозволяє кільком клієнтам одночасно встановлювати з'єднання з одним портом на одному хості до кількох інших машин. Це дозволяє кільком клієнтам отримувати доступ до сервера через загальний порт;
- універсальна доступність: TCP підтримується будь-яким пристроєм, що використовується в IP-мережі.

Недоліки TCP:

- потреба в налаштуванні з'єднання: Необхідно завершити процес налаштування з'єднання, перш ніж з'єднання зможе пройти через TCP. У мережах з великими затримками між відправником і одержувачем, процес рукоштовування може зайняти значну кількість часу;

– вплив регулювання потоку на відео: Автоматичний механізм керування потоком TCP сповільнює передачу даних у разі виявлення помилок передачі. Якщо швидкість передачі опускається нижче мінімальної швидкості, необхідної для передачі відеосигналу, отримувач відеосигналу може працювати неналежним чином;

– недоцільність повторної передачі: Якщо пакети повторно передаються, вони можуть надходити занадто пізно, щоб бути корисними. Це може впливати на інші дані, які прибули б своєчасно;

– відсутність багатоадресної передачі: TCP не підтримує багатоадресну передачу, тому окреме з'єднання має бути встановлено відправником для кожного отримувача. Це унеможлиблює використання односторонніх мереж, оскільки TCP вимагає процесу рукоштовкування;

– зменшення пропускної здатності через затримки: В мережах з великими затримками TCP може суттєво зменшити пропускну здатність через час, який витрачається на очікування підтвердження пакетів.

Однією з важливих особливостей TCP є його можливість виявляти та виправляти помилки передачі, зокрема втрачені пакети. TCP ретельно відстежує кожен байт даних, який проходить через з'єднання, за допомогою поля заголовка кожного пакета, відомого як послідовний номер. У кожному пакеті послідовний номер вказує на порядковий номер першого байта цього пакета. Якщо пакет був втрачений або пошкоджений під час передачі, інформація в наступному пакеті не буде відповідати підрахункові, який отримувач зробив щодо отриманих раніше байтів.

За допомогою заголовка TCP, відомого як підтвердження, отримувач повідомляє відправника про відсутність деяких даних та необхідність повторної передачі. Таким чином, TCP забезпечує отримання кожного байта даних, який відправник намагався передати. Ця система є ефективною для передачі файлів, які не терплять пошкоджень, таких як електронна пошта та виконуваний програмний код.

Одна з особливостей TCP полягає у здатності контролювати потік даних під час з'єднання. Приймач встановлює буфер для отримання даних і повідомляє відправнику про стан цього буфера після отримання кожного пакета. Відправник відповідає за запобігання переповненню буфера приймача і регулювання швидкості передачі даних відповідно до змінних умов.

Керування цим процесом здійснюється за допомогою лічильників даних і підтверджень, які обмінюються між приймачем і відправником.

На жаль, окремі з цих механізмів, які роблять TCP ефективним для передачі даних, можуть створювати проблеми для передачі відео.

Необхідно пам'ятати, що дані відеосигналу повинні не лише приходити в повній цілості, але й своєчасно. Тому механізм, який пересилає втрачені пакети, може призводити до двох небажаних наслідків:

- якщо пакет повторно передається, але приходить занадто пізно для відтворення, він може займати буфер приймача, поки обробляється, а потім бути відкинутим;

- коли пакети повторно передаються, вони можуть займати пропускну здатність мережі, необхідну для передачі нових даних.

Крім того, вбудований механізм контролю потоку в TCP може впливати на транспорт відеоданих. Якщо пакети втрачаються до їх доставки приймачу, TCP може автоматично зменшити швидкість передачі. Це розумна стратегія, але не випадково, коли відеосигнал повинен передаватися в режимі реального часу. Зменшення швидкості передачі нижче мінімально необхідного для потоку відео може призвести до того, що відеозображення взагалі не буде формуватися. У разі втрати пакетів в режимі реального часу відеопотоку кращою стратегією є ігнорування втрат і продовження передачі даних настільки швидко, наскільки це необхідно, щоб не відставати від відео (і аудіо) вмісту.

TCP використовує порти та сокети у своїй роботі, але це відрізняється від протоколу UDP. У TCP кожне з'єднання встановлюється як пара сокетів, що складається з IP-адреси та номера порту на кожному кінці з'єднання. Якщо два

з'єднання мають один і той самий сокет на одному кінці (наприклад, два з'єднання з HTTP-сервісним сокетом 192.163.84.67:80), TCP вимагає, щоб інший кінець кожного з'єднання мав унікальний сокет. Це дозволяє одному веб-серверу обслуговувати кілька клієнтів одночасно, що відрізняє TCP від UDP, який об'єднує всі дані, що надходять до сокету.

Одна з особливостей TCP полягає у здатності контролювати потік даних під час з'єднання. Приймач встановлює буфер для отримання даних і повідомляє відправнику про стан цього буфера після отримання кожного пакета. Відправник відповідає за запобігання переповненню буфера приймача і регулювання швидкості передачі даних відповідно до змінних умов.

Керування цим процесом здійснюється за допомогою лічильників даних і підтверджень, які обмінюються між приймачем і відправником.

На жаль, окремі з цих механізмів, які роблять TCP ефективним для передачі даних, можуть створювати проблеми для передачі відео.

Необхідно пам'ятати, що дані відеосигналу повинні не лише приходити в повній цілості, але й своєчасно. Тому механізм, який пересилає втрачені пакети, може призводити до двох небажаних наслідків:

- якщо пакет повторно передається, але приходить занадто пізно для відтворення, він може займати буфер приймача, поки обробляється, а потім бути відкинутим;

- коли пакети повторно передаються, вони можуть займати пропускну здатність мережі, необхідну для передачі нових даних.

Крім того, вбудований механізм контролю потоку в TCP може впливати на транспорт відеоданих. Якщо пакети втрачаються до їх доставки приймачу, TCP може автоматично зменшити швидкість передачі. Це розумна стратегія, але не випадково, коли відеосигнал повинен передаватися в режимі реального часу. Зменшення швидкості передачі нижче мінімально необхідного для потоку відео може призвести до того, що відеозображення взагалі не буде формуватися. У разі втрати пакетів в режимі реального часу відеопотоку кращою стратегією є

ігнорування втрат і продовження передачі даних настільки швидко, наскільки це необхідно, щоб не відставати від відео (і аудіо) вмісту.

TCP використовує порти та сокети у своїй роботі, але це відрізняється від протоколу UDP. У TCP кожне з'єднання встановлюється як пара сокетів, що складається з IP-адреси та номера порту на кожному кінці з'єднання. Якщо два з'єднання мають один і той самий сокет на одному кінці (наприклад, два з'єднання з HTTP-сервісним сокетом 192.163.84.67:80), TCP вимагає, щоб інший кінець кожного з'єднання мав унікальний сокет. Це дозволяє одному веб-серверу обслуговувати кілька клієнтів одночасно, що відрізняє TCP від UDP, який об'єднує всі дані, що надходять до сокету.

2.4. RTP

Транспортний протокол реального часу (або протокол реального часу, якщо вам більше подобається) призначений для мультимедійних програм в режимі реального часу, таких як голос та відео через Інтернет. RTP спеціально розроблений для передачі сигналів, де час є важливим фактором. Наприклад, для багатьох сигналів реального часу, якщо швидкість доставки пакетів опускається нижче критичного рівня, то неможливо сформувати корисний вихідний сигнал на приймачі. У таких випадках втрати пакетів можуть бути краще терпимі, ніж запізнення доставки.

RTP був створений для забезпечення набору функцій, які корисні для передачі відео та аудіо в режимі реального часу через Інтернет.

Один з прикладів використання RTP для сигналів в режимі реального часу – це голосові розмови. Більшість користувачів мобільних телефонів підтвердять, що випадкові шумові артефакти (такі як звукові спотворення) несуттєво не заважають поточній розмові. У порівнянні з цим, якщо мобільний телефон призначався для зупинки передачі та повторної передачі пакетів голосових даних при кожній помилці, система постійно переривалась би і була малофункціональною. Аналогічна ситуація відноситься до відео: короткі тимчасові збої бажаніші, ніж «ідеальний» сигнал, який постійно переривається та перезапускається, щоб

відновити втрачені біти. RTP будується на цій філософії: випадкові помилки даних або втрачені пакети не передаються автоматично повторно. Крім того, RTP не намагається контролювати бітрейт, який використовується відправником додатка; він не має вбудованих функцій автоматичного зниження швидкості для обробки перевантажених мереж, як TCP.

RTP застосовується в різних додатках, включаючи популярний стандарт відеоконференцій H.323. Однією з приємних особливостей RTP є його підтримка багатопартійних конференцій, що дозволяє різним учасникам конференції відстежувати приєднання та відключення інших учасників. RTP також може підтримувати передачу до багатьох адресатів одночасно.

Крім того, RTP надає функцію відмітки часу, яка дозволяє створювати множинні потоки з одного джерела для забезпечення синхронізації. Кожен тип корисного навантаження (відео, аудіо, голос) має відповідний спосіб кодування в RTP.

Наприклад, приймач може отримувати лише аудіо частину відеоконференції (що може бути корисно для учасника з низькопропускнуою мережею). Кожне джерело вставляє відмітки часу в заголовки вихідних пакетів, які можуть бути оброблені приймачем для відновлення тактового сигналу потоку, необхідного для правильного відтворення аудіо та відео.

Справді, RTP використовує UDP як механізм передачі пакетів. Зокрема, RTP додає заголовок до кожного пакету, який потім передається UDP для подальшої обробки, разом з іншим заголовком. RTCP, або протокол управління RTP (скорочення, яке може бути плутаним), використовується разом з RTP. При встановленні з'єднання RTP, RTCP синхронізує файли між різними типами носіїв, наприклад відео і аудіо. RTCP містить інформацію про відмітку часу, яка використовується приймачем для синхронізації годинників у кожному потоці RTP, забезпечуючи синхронізацію аудіо- і відеосигналів.

Кожен приймач також повинен підраховувати кількість інших приймачів, які присутні на сесії. Ця кількість використовується для визначення частоти

надсилання звітів кожним приймачем. Зі збільшенням кількості приймачів, які приєднуються до сесії, частота надсилання звітів зменшується, щоб уникнути перевантаження мережі або інших приймачів. Для розрахунку цього використовується рекомендація, згідно з якою трафік RTCP не повинен перевищувати 5% від загального обсягу трафіку RTP, пов'язаного з ним.

Переваги протоколу RTP:

- різноманітні формати: RTP має вбудовану підтримку багатьох типів аудіо- та відеопотоків, що дозволяє легко обмінюватися MPEG-відео та іншими аудіоформатами, використовуючи чітко визначені стандарти;

- порядкові номери пакетів: Ці номери дозволяють визначити, чи отримала програма всі пакети і чи прийшли вони у правильному порядку. Це сповіщає отримувача про необхідність повторної передачі, виправлення помилок або маскування даних;

- підтримка мультикастингу: RTP може розподіляти потоки за допомогою мультикастової передачі, що дозволяє одному джерелу відео передавати його одночасно до кількох (або сотень) приймачів;

- вбудована синхронізація даних: Синхронізація між різними типами потоків дозволяє передавати мультимедійні сигнали (відео, аудіо тощо) окремо та зберігати їх у відповідності до загального часового шкалі на приймачі;

- різні швидкості передачі: Приймаючі пристрої можуть вибирати декодування лише певної частини загального обсягу програми, тому пристрої з низькими швидкостями мережі можуть вибирати декодування тільки аудіопотоку, наприклад.

Недоліки протоколу RTP:

- проблеми з брандмауером: Деякі брандмауери можуть блокувати RTP-трафік, оскільки UDP-пакети часто заблоковані у багатьох налаштуваннях;

- відсутність пріоритетів: RTP не має вбудованого механізму для забезпечення своєчасної передачі пакетів, наприклад, пріоритетів черги;

– обмежена поширеність: Не всі мережеві пристрої підтримують RTP, тому необхідно переконатися, що мережа належним чином налаштована для використання цього протоколу;

– загалом, RTP надає багато функціональних можливостей в порівнянні з UDP, не додаючи багато непотрібних функцій, як у TCP. Наприклад, RTP не автоматично знижує пропускну здатність передачі при втраті пакетів, а надає інформацію для подальшої обробки.

3. Потоковий сервер

Стрімінговий сервер відповідає за розподіл медіапотоків до глядачів. Він приймає мультимедійний вміст, який зберігався внутрішньо, та створює потік для кожного запиту глядача. Ці потоки можуть бути одноадресними або багатоадресними і керуватися за допомогою різних механізмів. У цьому розділі ми розглянемо одноадресну передачу, оскільки вона є найпоширенішою формою потокового передавання на сьогоднішній день.

Зберігання та пошук вмісту – одна з основних функцій стрімінгового сервера. Під час підготовки вмісту він зазвичай стискається у різних рівнях компресії, щоб користувачі з різною швидкістю інтернет-підключення могли вибрати відповідну швидкість відтворення відео.

Наприклад, на багатьох веб-сайтах користувачам надається вибір між швидкістю відтворення, що підходить для мобільного з'єднання (56 кбіт/с, іноді називається низькою швидкістю), середньою швидкістю з'єднання (100–300 кбіт/с) та високою швидкістю з'єднання (іноді 500–1500 кбіт/с).

Кожна з цих різних швидкостей відтворення вимагає створення різних версій файлу вмісту під час процесу стиснення. Таким чином, одна частина вмісту може бути представлена у трьох різних файлах на сервері, по одному для кожної варіації швидкості відтворення. Ця ситуація може стати ще складнішою, оскільки сервер може підтримувати кілька різних типів програвачів, таких як QuickTime або Windows Media Player. Оскільки більшість постачальників контенту прагнуть охопити якомога більше аудиторії, вони зазвичай кодують відео, що сумісне з

двома або трьома найпопулярнішими програвачами. Отже, потоковий сервер може містити безліч різних копій кожного вмісту. Наприклад, якщо сервер має обробляти три різні варіанти швидкості з'єднання (низьку, середню і високу) та три різні формати медіаплеєра (Flash, Windows Media Player і QuickTime), загалом на сервері буде зберігатися дев'ять різних форматів відеофайлів.

Великою роботою потокового сервера є створення пакетів для кожного потоку у режимі реального часу. Як ми вже згадували, кожен IP-пакет повинен мати джерело та призначення IP-адресу. Для того, щоб відеопакети досягали правильного місця призначення, сервер повинен створювати заголовки цих пакетів з правильною призначенням IP-адресою. Якщо сервер використовує стандартні пакети RTP, відео- та аудіопотоки повинні бути надіслані як окремі потоки пакетів на різні порти приймача.

Ще одним завданням сервера є шифрування пакетів у вихідному потоці, якщо це вимагає власник вмісту. Оскільки більшість сучасних методів шифрування використовують криптографію з відкритим ключем, кожен потік повинен бути зашифрований унікальним ключем для кожного користувача.

Таким чином, шифрування повинно бути здійснене на кожному потоці, поки він відправляється з потокового сервера. Оскільки трансляція відбувається в режимі реального часу, сервер також повинен створювати стабільні потоки, забезпечуючи регулярний темп пакетів без швидкого прискорення або сповільнення.

Також важливо, щоб швидкість передачі пакетів відповідала вимогам програмного забезпечення програвача, щоб уникнути переповнення або втрати даних у буферах програвача. Основна мета потокового сервера полягає в тому, щоб доставляти потік тривалістю, наприклад, 5 хвилин і 10 секунд, за 5 хвилин і 10 секунд.

Іноді потоковий сервер відповідає за зміну швидкості потоку, що передається глядачеві, в залежності від зміни мережевих умов. Наприклад, швидкість потоку може починатися зі значення 300 кбіт/с, а потім знижуватися до 100 кбіт/с, якщо

мережа перевантажена, на основі зворотного зв'язку, що надсилається з програвача на сервер потокової передачі.

У такому випадку сервер переключиться на поточкову версію вмісту, яка була записана з меншою швидкістю. Головна мета полягає в тому, щоб цей процес відбувався плавно, щоб користувач не помічав перемикання, за винятком незначного падіння якості відео або звуку. Цей процес також називається масштабуванням потоку і є важливою функцією багатьох передових систем потокової передачі.

Однією з великих переваг потокової технології є можливість користувача мати доступ до різноманітних медіапотоків. Це означає, що користувач може перемикатися вперед або назад у межах вмісту. Для реалізації цієї функції потрібна тісна співпраця між поточковим сервером та програвачем.

Програмне забезпечення програвача повинно приймати команди користувача і передавати їх на сервер, який потім може змінювати вихідний потік. Поточкові сервери, які підтримують одночасних переглядачів, також вимагають високошвидкісного зберігання та мережових з'єднань.

Для великих серверів, які обробляють широкосмугові потоки, використання гігабітних мережових інтерфейсів Ethernet стає стандартом. Крім того, для підтримки сотень або тисяч користувачів вміст може бути розміщений на кількох серверах у різних фізичних місцях.

4. Попередня обробка

Шляхи покращення ефективності стиснення відео CODEC включають попередню обробку відеокадрів перед кодуванням. Проблеми, пов'язані з вихідним матеріалом та/або системою захоплення відео, можуть погіршити ефективність кодування відеокодера. На рисунку 2.3 зображено шум камери, який виникає внаслідок роботи самої камери або процесу оцифрування.



Рисунок 2.3 – Зображення, на якому показано шум камери (нижня половина)

Верхня половина даного зображення є майже безшумною, що є характерною особливістю для високоякісних цифрових камер. Зображення, зроблені низькоякісними джерелами, часто містять шум (як видно в нижній половині цього зображення). Шум камери може виникати на високих просторових частотах і змінюватися від кадру до кадру. Кодер розпізнає цей шум як високочастотний компонент, який присутній у компенсованому руховому залишку, і кодує його разом з необхідними залишковими даними, що призводить до збільшення бітрейту кодування. Таким чином, шум камери може значно знизити ефективність стиснення кодера. Застосування фільтрації до вхідної відеопослідовності перед кодуванням може зменшити шум камери і, отже, покращити ефективність стиснення. Однак слід обережно вибирати параметри фільтра, щоб уникнути фільтрування корисних функцій відеопослідовності.

Іншим чинником, що може знизити ефективність стиснення, є тремтіння камери – невеликі рухи камери між послідовними кадрами, що характерні для ручних або погано стабілізованих камер. Кодер сприймає їх як глобальний рух між кадрами. Компенсація руху може частково скоригувати цей рух, але блочні

алгоритми оцінки руху зазвичай не можуть повністю усунути тремтіння камери, що призводить до збільшення залишкової енергії та зниження продуктивності стиснення. Багато споживачів та професіоналів відеокамер використовують системи стабілізації зображення, які автоматично намагаються компенсувати невеликі рухи камери за допомогою механічних або обробних методів зображення. Таке покращення зовнішнього вигляду знятої відеопослідовності сприяє підвищенню ефективності стиснення, оскільки матеріал кодується з використанням компенсації руху. [12]

5. Пост-обробка

Алгоритми стиснення відео, такі як квантування (наприклад, основні алгоритми MPEG-4 Visual і H.264), по суті мають втрати, тобто декодовані відеокадри не є ідентичними оригіналу. Мета будь-якого практичного CODEC – мінімізувати спотворення та максимізувати ефективність стиснення. Часто можна зменшити фактичні або помітні спотворення декодованої відеопослідовності шляхом обробки (фільтрації) декодованих кадрів [12]. Якщо відфільтровані декодовані кадри використовуються для компенсації, процес фільтрації може мати додаткову перевагу в поліпшенні передбачення з компенсацією руху і, отже, ефективності стиснення.



Рисунок 2.4 – Спотворення, спричинене візуальним кодуванням MPEG-4

Блокові кодеки, засновані на перетворенні, спричиняють характерні спотворення в декодованих відеоданих. На нижній половині рисунку 2.4 показані типові спотворення в кадрі, закодованому і декодованому з використанням простого профілю MPEG-4 (верхня половина – оригінальний, некомпресований кадр). У цьому прикладі показані спотворення, що виникають внаслідок блокування (спричиненого невідповідністю меж реконструйованих блоків розміром 8×8) та «дзвінкі» спотворення (слабкі візерунки по краях об'єктів, що виникають внаслідок «прориву» базових шаблонів DCT).

Блокування, ймовірно, є найбільш візуально помітним (і, отже, найважливішим) типом спотворень, який відображається в стисненому відео. На рисунку 2.5 (нижня половина) показано результат кодування та декодування без застосування циклічного фільтру в H.264. Зменшений розмір перетворення в H.264 (4×4 зразки, замість 8×8) означає, що блокуючі артефакти є менш помітними, але все ще спостерігаються.

Висновки до розділу

У цьому розділі було розглянуто спосіб створення IP-пакетів із стиснених відеопотоків. Було розпочато з обговорення компромісів, пов'язаних з використанням довгих або коротких пакетів. Далі розглянуто популярний формат сигналу, який використовується відеоіндустрією для передачі MPEG транспортних потоків.

Потім детально розглянуто транспортні протоколи, які можна використовувати для транспортування відеопотоків з різним ступенем успіху. TCP забезпечує доставку кожного байта даних від відправника до одержувача і має вбудовані механізми для автоматичного забезпечення цього. У випадку втрати або пошкодження даних на шляху, TCP повторно передає дані.

Також розглянуто протокол UDP, який має просту схему адресації портів. Ця схема дозволяє безпосередньо переадресувати будь-які дані, що надходять до певного сокета, до програми, що прикріплена до цього сокета. Це також усуває потребу в транспортному протоколі (в даному випадку UDP) для відстеження декількох різних підключень до одного порту.

Зроблено висновок про RTP, який є цінним методом обробки аудіо- та відеосигналів у реальному часі.

Стратегія планування змінюється при будь-якій зміні конфігурації мережі. Тому в одноранговому зв'язку вибирається інший блок послідовності відповідно до поточної конфігурації мережі або поточного попиту в одноранговій мережі. Активна адаптація є динамічною і має вирішальне значення для схеми планування.

3 ПРАКТИЧНЕ ПОРІВНЯННЯ СУЧАСНИХ ІНСТРУМЕНТІВ ДЛЯ ПРОВЕДЕННЯ ПРЯМИХ ВІДЕОТРАНСЛЯЦІЇ

1. Практична реалізація відеотрансляцій за допомогою HLS (HTTP Live Streaming)

1.1. Як працює HLS

HLS (HTTP Live Streaming) використовує сегментований підхід до передавання відео, що дозволяє гнучко і ефективно передавати відеоконтент через мережу Інтернет. Відеофайли розбиваються на невеликі сегменти, які можуть бути завантажені і відтворені поетапно на пристроях переглядачів.

Процес розбиття відео на сегменти виконується на сервері, який відповідає за трансляцію. Спочатку відеофайл перетворюється на кілька різних бітрейтів, що представляють різні якості відео. Кожен бітрейт відповідає за один варіант сегменту відео. Наприклад, можуть бути створені сегменти з низьким, середнім і високим бітрейтами.

Після розбиття відео на сегменти, сервер створює медіаплейлист, який містить посилання на всі доступні сегменти відео. Цей медіаплейлист може бути представлений у форматі M3U8 - текстовому файлі, який містить URL-адреси сегментів відео і їх властивості, такі як бітрейт, розмір, тривалість тощо.

Коли користувач відкриває відеоплеєр на своєму пристрої, плеєр завантажує медіаплейлист з сервера. За допомогою медіаплейлиста, плеєр отримує список доступних сегментів відео і їх властивості. Плеєр починає завантажувати перший сегмент відео з відповідного бітрейту.

Після завантаження сегмента, плеєр починає відтворення цього сегмента. Під час відтворення плеєр постійно перевіряє наявність нових сегментів і автоматично завантажує їх під час відтворення. Цей процес дозволяє плеєру адаптуватися до змінюючихся умов мережі і пристосовувати якість відео в реальному часі. Наприклад, якщо з'єднання мережі стає повільним, плеєр може переключитися на нижчий бітрейт, щоб забезпечити безперебійне відтворення.

Таким чином, HLS забезпечує адаптивну передачу відео залежно від швидкості мережі та можливостей пристрою переглядача. Користувачі можуть насолоджуватися відеоконтентом в різних якостях і на різних пристроях без необхідності завантажувати весь відеофайл перед відтворенням. HLS надає гнучкість і забезпечує оптимальне відеопередавання з високою якістю зображення та ефективним використанням пропускну здатності мережі.

1.2. Переваги та недоліки HLS

Переваги HLS:

– адаптивна передача відео: Однією з найбільших переваг HLS є його здатність адаптуватися до різних умов мережі та пристроїв переглядачів. HLS може динамічно змінювати якість відео, вибираючи оптимальний бітрейт в залежності від швидкості мережі. Це дозволяє користувачам насолоджуватися безперебійним відтворенням навіть при обмежених мережевих ресурсах;

– кросс-платформеність: HLS підтримується більшістю сучасних платформ і пристроїв, таких як комп'ютери, смартфони, планшети та Smart TV. Це робить його універсальним рішенням для доставки відеоконтенту на різних пристроях та операційних системах;

– зменшення навантаження на сервер: HLS дозволяє розподіляти навантаження на сервер за рахунок розбиття відео на сегменти. Користувачі завантажують лише необхідні сегменти, що допомагає зменшити пропускну здатність та навантаження сервера, особливо при великій кількості одночасних переглядів;

– підтримка живого відео: HLS добре підходить для трансляції живого відео, так як сегментований підхід дозволяє реалізувати миттєве відтворення без буферизації великих відеофайлів. Це особливо важливо для подій у реальному часі, спортивних змагань та онлайн-трансляцій.

Недоліки HLS:

– затримка відтворення: так як відео розбивається на сегменти, існує невелика затримка між виробництвом та відтворенням сегментів. Це може

впливати на реалізацію реального часу, особливо для важливих подій, де миттєва відповідь є критичною;

- потреба у буферизації: HLS використовує буферизацію, щоб забезпечити безперебійне відтворення відео при зміні якості або умов мережі. Це може приводити до невеликого затримки та буферизаційних проблем у деяких ситуаціях.

- використання додаткових ресурсів: HLS вимагає додаткових ресурсів для сегментування відео та управління плейлистами. Це може призводити до збільшення обсягу зберігання та обчислювального навантаження на сервері;

- підтримка старих версій протоколу: Деякі старіші пристрої та браузері можуть не підтримувати новіші версії HLS, що може призвести до несумісності та обмежень у доступі до відеоконтенту.

Враховуючи ці переваги та недоліки, HLS залишається одним з популярних стандартів передачі відео в режимі реального часу. Його гнучкість, кросс-платформеність та підтримка адаптивного відтворення роблять його привабливим варіантом для постачання відеоконтенту на різних пристроях та мережах.

2. Практична реалізація відеотрансляцій за допомогою RTMP (Real-Time Messaging Protocol)

Одна з основних переваг RTMP полягає у його здатності передавати відео в реальному часі з мінімальною затримкою. Це особливо важливо для стрімінгу спортивних подій, концертів, вебінарів та інших подій, де потрібно миттєве відтворення вмісту. RTMP також забезпечує гнучкість інтерактивності між стрімером та глядачами. Наприклад, користувачі можуть відправляти коментарі, поставити лайк або ділитися емоціями під час прямого ефіру. Це дозволяє взаємодіяти з аудиторією в реальному часі, що створює відчуття спільноти і підвищує залученість глядачів.

Крім того, RTMP підтримує різні якості відео, що дозволяє адаптувати потік до швидкості Інтернет-з'єднання глядача. Наприклад, якщо у користувача повільне з'єднання, RTMP може автоматично знизити якість відео, щоб запобігти зависанню

або буферизації. Це забезпечує плавний та безперебійний перегляд навіть при обмежених мережевих умовах.

Для використання RTMP необхідно мати налаштований енкодер, який кодує відео та аудіо в потік даних, сумісний з протоколом RTMP. Часто використовуються популярні енкодери, такі як OBS (Open Broadcaster Software), Wirecast і XSplit. Вони надають широкі можливості для налаштування відео, аудіо та інших параметрів трансляції.

RTMP також підтримує можливість підключення зовнішніх камер і пристроїв за допомогою кодерів. Це дозволяє використовувати професійну камеру або систему зйомки, щоб отримати високоякісний відеоконтент для трансляцій. Крім того, RTMP дозволяє налаштувати резервні потоки, які автоматично активуються, якщо основний потік виявляється недоступним або має проблеми зі з'єднанням.

2.2. Переваги та недоліки RTMP

RTMP (Real-Time Messaging Protocol) - це протокол передачі в реальному часі, який забезпечує ефективну та надійну доставку відео контенту через Інтернет. RTMP використовується для стрімінгу відео на живі платформи, такі як YouTube Live, Twitch і Facebook Live. Протокол RTMP пропонує кілька переваг, які зробили його популярним серед стримерів і відеоблогерів.

Основні переваги RTMP:

- низька затримка: RTMP дозволяє передавати відео практично в реальному часі, що особливо важливо для прямих трансляцій, таких як спортивні події або концерти. Завдяки мінімальній затримці між відправкою та отриманням даних, глядачі можуть переглядати вміст майже миттєво;

- висока якість відео: RTMP підтримує передачу відео з високою якістю, включаючи HD та 4K. Це дозволяє стримерам представляти свій контент у високій роздільній здатності і забезпечує більш якісний досвід перегляду для глядачів;

- широка сумісність: RTMP підтримується багатьма платформами та пристроями, включаючи комп'ютери, мобільні пристрої і телевізори з підтримкою

інтернет-підключення. Це дозволяє стрімерам досягти більшої аудиторії та забезпечити доступність контенту на різних пристроях;

– інтерактивність: RTMP дозволяє взаємодіяти з глядачами у реальному часі. Це означає, що стрімери можуть спілкуватися з аудиторією через чат, приймати команди або відповідати на запитання під час трансляції. Це створює більш особистий та залучаючий досвід для глядачів.

Незважаючи на ці переваги, RTMP також має деякі обмеження і недоліки:

– потреба в медіа-сервері: для використання RTMP потрібен медіа-сервер, який може бути складним у налаштуванні та вимагати додаткових витрат. Це може бути проблемою для початківців або користувачів з обмеженими ресурсами;

– проблеми зі сумісністю: деякі платформи або пристрої можуть не підтримувати протокол RTMP, що обмежує можливості розповсюдження контенту. В таких випадках може знадобитися конвертація формату або використання альтернативних протоколів;

– вразливість до затримок і втрати даних: RTMP може бути вразливим до затримок та втрати даних при нестабільному Інтернет-з'єднанні. Це може призвести до переривання трансляції або зниження якості відео;

– обмежена масштабованість**: RTMP може стикатися з обмеженнями масштабованості при великій кількості одночасних переглядів. Це може призвести до зниження продуктивності або нестабільності трансляції.

Однак, з урахуванням правильного налаштування та використання, RTMP залишається потужним інструментом для проведення прямих відеотрансляцій. Залежно від потреб і особливостей використання, користувачі можуть обирати між різними платформами та протоколами, включаючи RTMP, для досягнення найкращих результатів.

Висновки до розділу

Ця стаття порівнює два популярних протоколи, які можуть використовуватись для стрімінгу відео на власному веб-сайті і забезпечують

більшу гнучкість та контроль над відеовмістом. Це RTMP (Real-Time Messaging Protocol) і HLS (HTTP Live Streaming).

RTMP є протоколом передачі в реальному часі, який використовується для стрімінгу відео на платформи, такі як YouTube Live та Twitch. Він забезпечує швидку та надійну доставку відео з мінімальною затримкою. Однак, RTMP має свої обмеження, зокрема щодо сумісності з деякими пристроями та браузерами. Деякі пристрої, особливо мобільні, можуть мати обмежену підтримку RTMP.

HLS, з іншого боку, є протоколом передачі в реальному часі через HTTP. Він розбиває відео на невеликі частини та передає їх через HTTP запити. Це дозволяє використовувати HLS на широкому спектрі пристроїв і браузерів, включаючи мобільні пристрої. HLS також має вбудовану підтримку адаптивного потоку, що дозволяє автоматично пристосовувати якість відео до швидкості з'єднання глядача.

Одним із переваг HLS є можливість індексування відеовмісту пошуковими системами, що дозволяє легше знаходити та просувати вміст на веб-сайті. Крім того, HLS забезпечує більшу безпеку, оскільки відео потоки можуть бути захищені шифруванням та DRM-технологіями.

Вибір між RTMP та HLS залежить від вашої конкретної ситуації та потреб. Якщо вам потрібно стрімити на популярні платформи, як YouTube Live, то RTMP може бути зручним варіантом. Однак, якщо ви бажаєте мати повний контроль над відеовмістом та забезпечити сумісність з широким спектром пристроїв, HLS може бути кращим вибором для вас.

4 ЕКСПЕРИМЕНТАЛЬНА ПЕРЕВІРКА РЕЗУЛЬТАТІВ АНАЛІТИЧНИХ ДОСЛІДЖЕНЬ

Для проведення дослідження бітрейту в потоковому відео було розроблено та реалізовано код мовою програмування JavaScript з використанням бібліотеки Node.js та пов'язаних інструментів, таких як ffprobe, Express, Node Media Server та ffmpeg. Ці інструменти дозволили створити середовище для запису відео з різними дозволами та аналізу їхнього бітрейту. Експеримент являє собою поточкову передачу відео за протоколом RTMP з подальшим транскодуванням в різні роздільні здатності та вимірюванням бітрейту за допомогою інструмента ffprobe.

1. Реалізація коду

Ця програма була налаштована на запис відео з роздільними здатностями 1080p, 720p, 480p та 240p. Для цього було використано потужний інструмент ffmpeg, який надав гнучкість у кодуванні відео та аудіо, а також у виборі формату вихідного файлу. Були визначені параметри кодування, такі як відеокодек (libx264), щоб забезпечити оптимальну якість відео із заданим бітрейтом.

Код написаний за допомогою фреймворку Node Media Server (NMS) та бібліотеки Fluent-FFmpeg для обробки медіа. Нижче наведено огляд блоків коду та їх функціональності.

На рисунку 4.1 наведений блок коду, який здійснює налаштування Node Media Server на передачу потокового відео за допомогою RTMP-протоколу.

```

const NodeMediaServer = require('node-media-server');

const config = {
  rtmp: {
    port: 1935,
    chunk_size: 60000,
    gop_cache: true,
    ping: 30,
    ping_timeout: 60,
  }
};

const nms = new NodeMediaServer(config);

```

Рисунок 4.1 – Ініціалізація NMS та налаштування параметрів RTMP-сервера

Далі було реалізовано транскодування відео в обробнику події prePublish, який викликається перед початком публікації потоку. Це показано на рисунку 4.2.

```

nms.on('prePublish', (id, StreamPath, args) => {
  const inputUrl = `rtmp://localhost${StreamPath}`;

  const outputFilename = `output_${Date.now()}.mp4`;

  const outputPath = `./media/${outputFilename}`;

  const command = ffmpeg(inputUrl)
    .output(outputPath)
    .videoCodec(transcodeOptions.videoCodec)
    .audioCodec(transcodeOptions.audioCodec)
    .format(transcodeOptions.outputFormat)
    .outputOptions('-g 10')
    .outputOptions('-t 10')
    .on('end', () => {
      console.log('Transcoding finished.');
      getVideoInfo(outputPath)
    })
    .on('error', (err) => {
      console.error('An error occurred:', err.message);
    });
  // Start transcoding
  command.run();

```

Рисунок 4.2 – Транскодування відео

Після транскодування відео, була використана функція getVideoInfo для отримання інформації про бітрейт за допомогою інструмента ffprobe. Далі

запускалася команда `ffprobe` із зазначенням шляху до вихідного файлу, який був транскодований за допомогою `ffmpeg`. Отриманий результат бітрейту виводився консоль для подальшого аналізу, як показано на рисунку 4.3.

```
function getVideoInfo(videoSource) {
  exec(`ffprobe -v error -select_streams v:0 -show_entries stream=bit_rate -of default=noprint_wrappers=1:nokey=1 ${videoSource}`,
    (err, stdout, stderr) => {
      if (stderr) {
        console.error('Ошибка при выполнении команды ffprobe:', stderr);
        return;
      }

      console.log('Битрейт видео 240p:', stdout / 1000);
    }
  );
};
```

Рисунок 4.3 – Вимір бітрейту

Далі запускається Node Media Server для обробки RTMP-потоків командою `nms.run()`.

Потім, для того, щоб створити запис відео з пристрою захвату відео, було реалізовано код, який продемонстровано на рисунку 4.4.

```
const cameraSource = 'video=@device_pnp_\\\\\\\\?\\usb#vid_04f28

const ffmpegCommand = ffmpeg()
  .input(cameraSource)
  .inputFormat('dshow')
  .videoCodec('libx264')
  .outputOptions('-tune', 'zerolatency')
  .outputOptions('-rtbufsize', '300M')
  .outputOptions('-f', 'flv')
  .outputOptions('-s', '320x240')
  .outputOptions('-t', '15')
  .output('rtmp://localhost/live/stream')
  .on('error', (err, stdout, stderr) => {
    console.error('Ошибка ffmpeg:', err.message);
    console.error('stdout:', stdout);
    console.error('stderr:', stderr);
  });

ffmpegCommand.run();
```

Рисунок 4.4 – Запуск процесу для запису відео з камери

І нарешті треба налаштувати та запустити сервер для того, щоб усе працювало. Код наведений нижче на рисунку 4.5.

```
const express = require('express');
const app = express();

app.listen(3000, () => {
  console.log('Сервер запущен на порту 3000');
});
```

Рисунок 4.5 – налаштування та запуск сервера

2. Вимір бітрейту

Головне завдання експерименту було виміряти, як бітрейт відео залежить від роздільної здатності. Було проведено кілька запусків програми, щоразу записуючи відео з певною роздільною здатністю, а потім проаналізовано отримані значення бітрейту (Рисунок 4.6).

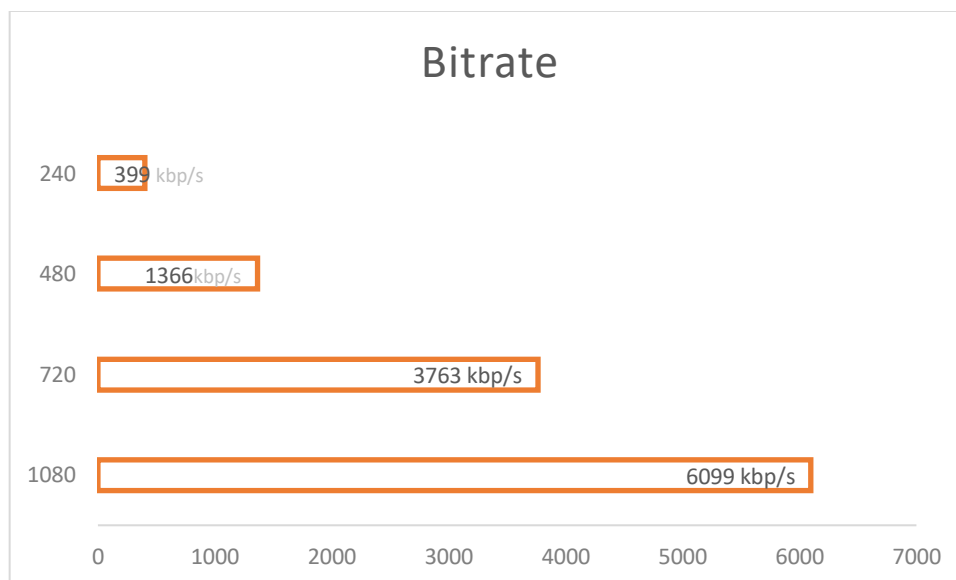


Рисунок 4.6 – Діаграма звіту

Це допомогло зрозуміти, як різні дозволи впливають на якість відео та його розмір. За результатами цих експериментів, було виявлено, що зі збільшенням роздільної здатності відео, бітрейт також збільшується. Нижче наведено отримані значення бітрейту для кожної роздільної здатності:

- роздільна здатність 1080р: 6099 кбіт/с. Рисунок 4.7;
- роздільна здатність 720р: 3763 кбіт/с. Рисунок 4.8;
- роздільна здатність 480р: 1366 кбіт/с. Рисунок 4.9;
- роздільна здатність 240р: 399 кбіт/с. Рисунок 4.10.

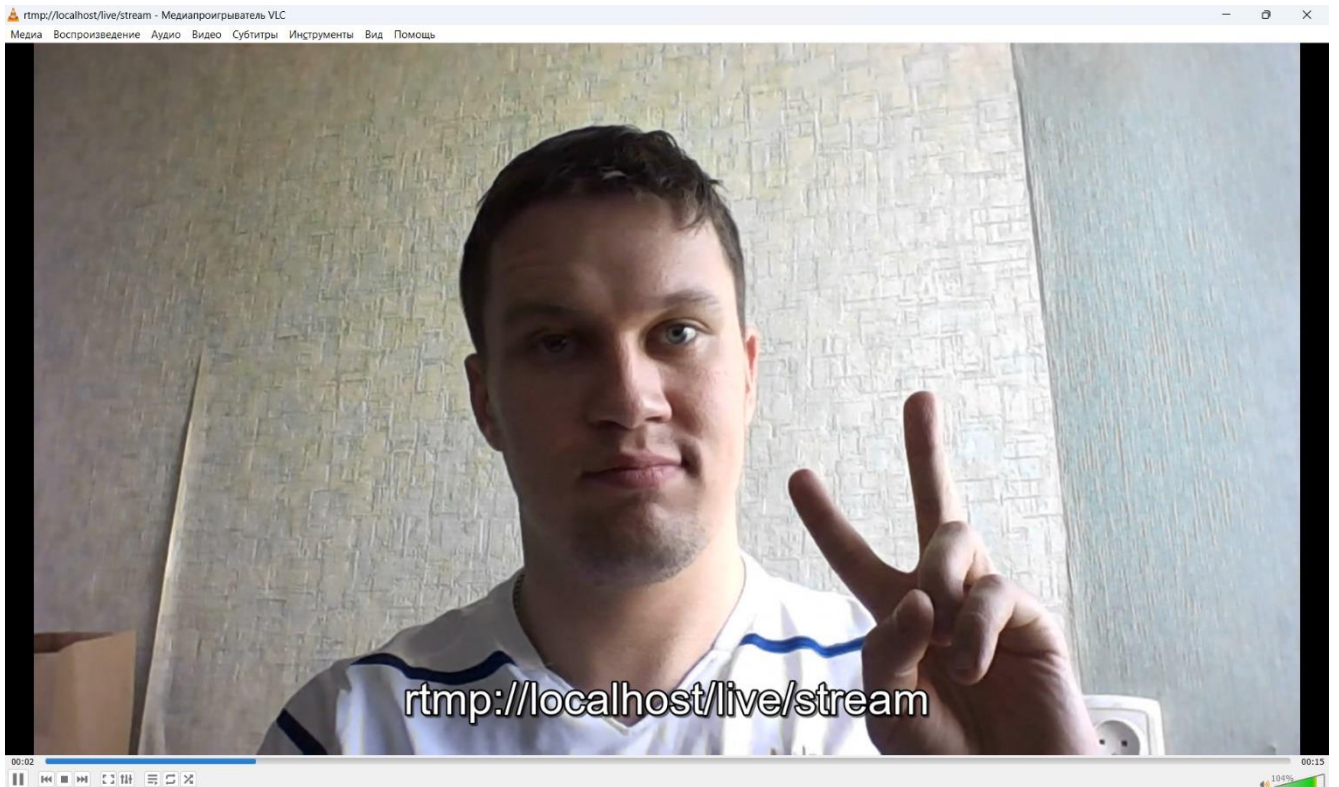


Рисунок 4.7 – Зображення з роздільною здатністю 1080p

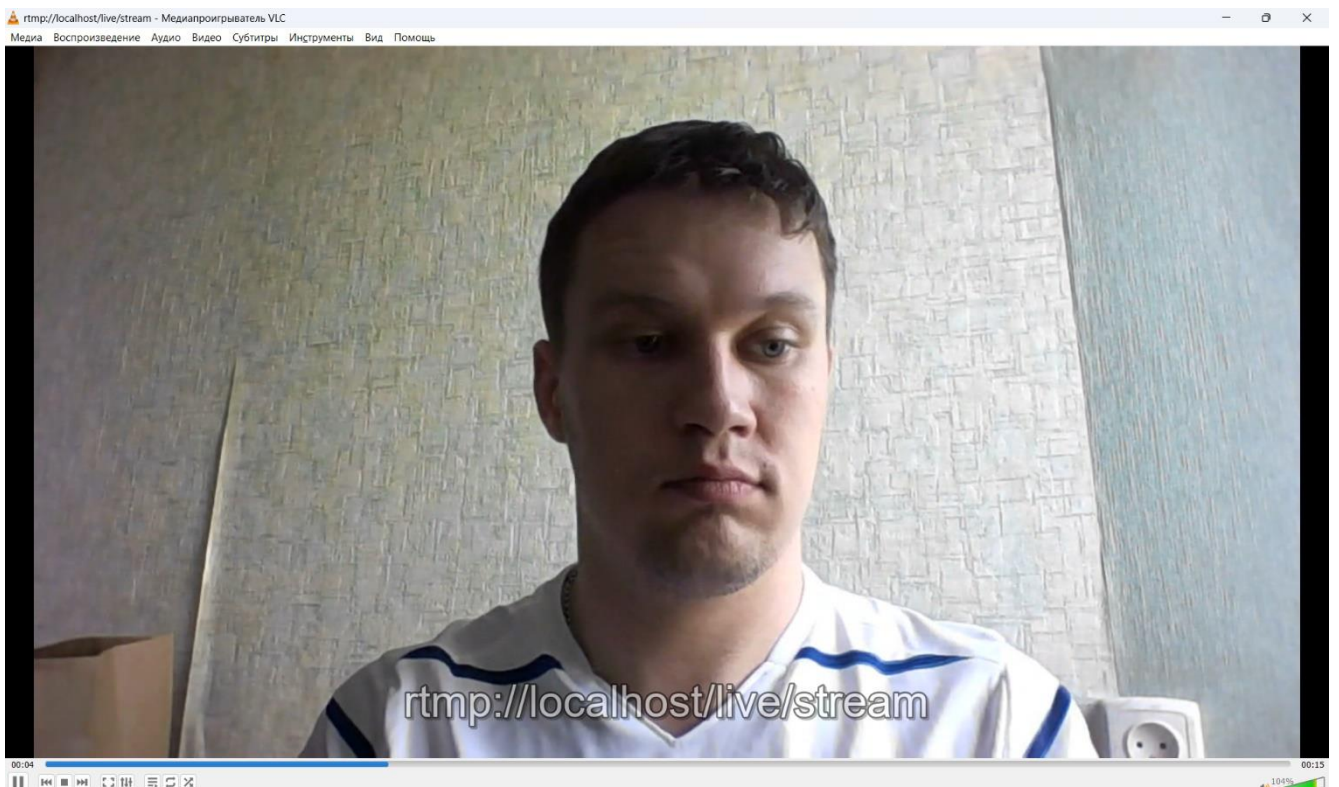


Рисунок 4.8 – Зображення з роздільною здатністю 720p

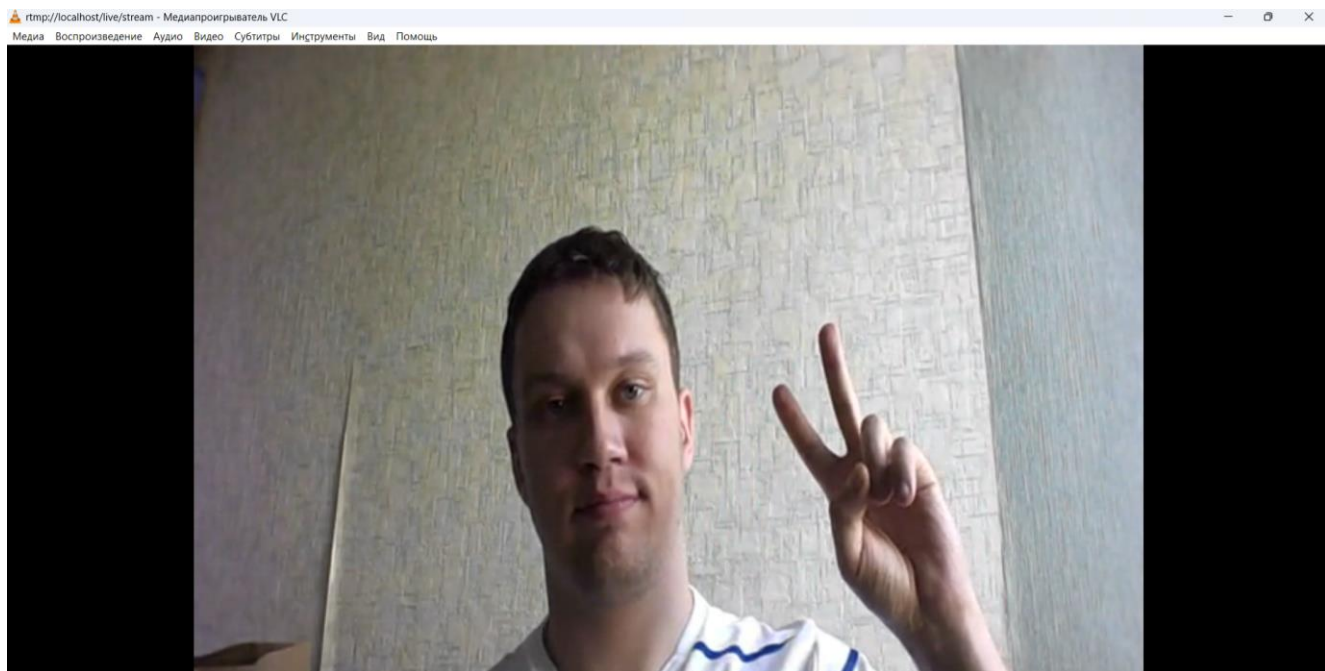


Рисунок 4.9 – Зображення з роздільною здатністю 480р

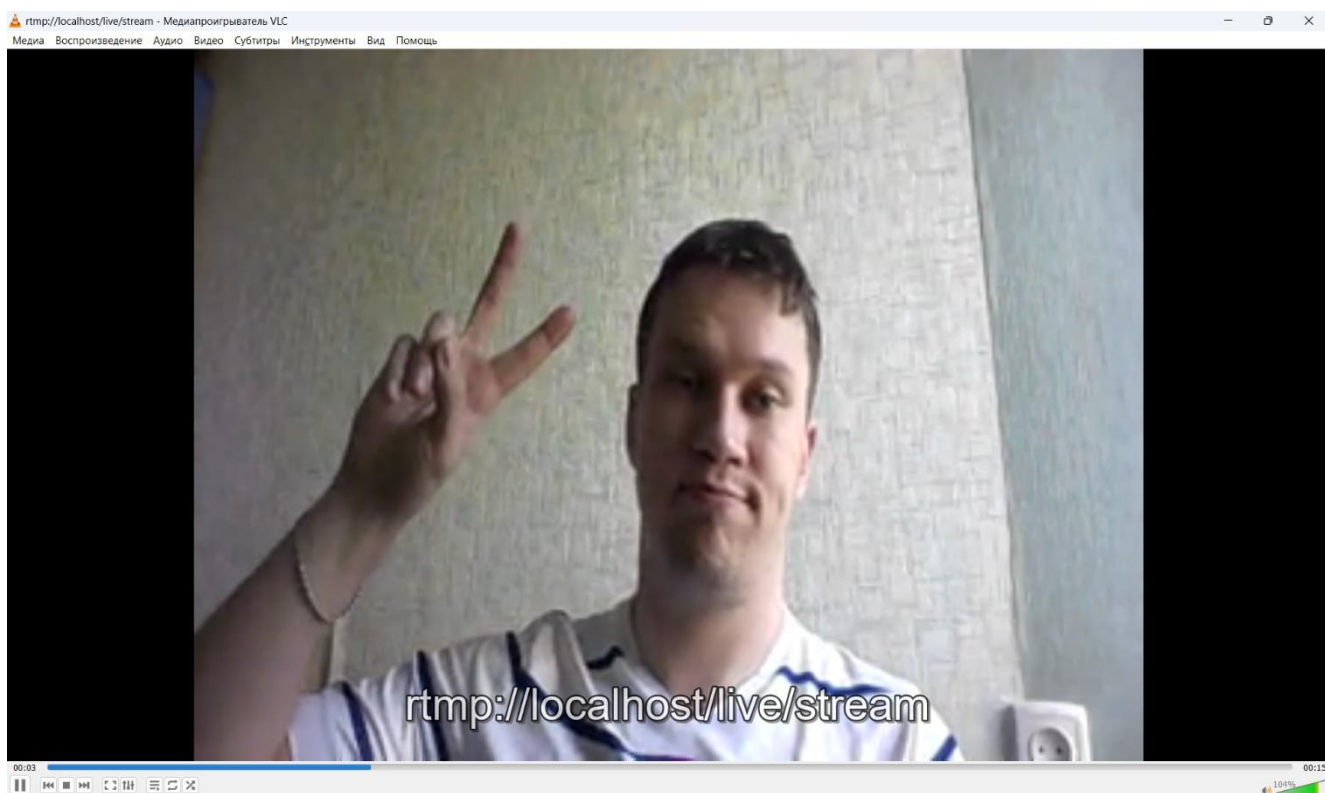


Рисунок 4.10 – Зображення з роздільною здатністю 240р

3. Висновки до розділу

Для проведення вимірювань бітрейту необхідно використовувати інструменти `ffmpeg` і `ffprobe`. `Ffmpeg` дозволяє записувати відео з певними параметрами, а `ffprobe` – аналізувати отримані відеофайли для отримання інформації про бітрейт. Ці інструменти надали потужні можливості вивчення залежності бітрейту від роздільної здатності.

Аналіз залежності бітрейту відео від різної роздільної здатності зображення дозволяє отримати важливі практичні дослідження. У цьому дослідженні були проаналізовані роздільні здатності 1080p, 720p, 480p і 240p, і були отримані такі результати: бітрейт відео зростає зі збільшенням роздільної здатності.

Було виявлено, що відео з роздільною здатністю 1080p демонструє найвищий бітрейт у розмірі 6099 кбіт/с, тоді як відео з роздільною здатністю 720p має бітрейт у розмірі 3763 кбіт/с. Для відео з роздільною здатністю 480p бітрейт становить 1366 кбіт/с, а для відео з роздільною здатністю 240p - 399 кбіт/с.

Таким чином, результати дослідження показують, що більша роздільна здатність зображення призводить до більш високого бітрейту відео. Це означає, що для досягнення більш високої якості відео та збереження більшої кількості деталей візуальної інформації, необхідно використовувати вищий бітрейт при кодуванні відео з більш високою роздільною здатністю.

Дане дослідження має практичне значення для розробників відео-платформ, стрімінгових сервісів та медіа-вмісту в цілому. Воно дозволяє приймати поінформовані рішення про вибір оптимального дозволу відео та відповідного бітрейту, ґрунтуючись на балансі між якістю відео та його розміром. Це може призвести до покращення продуктивності, оптимізації споживання ресурсів та підвищення задоволеності користувачів під час перегляду відео контенту.

Однак слід зазначити, що залежність бітрейту від роздільної здатності відео є лише одним із факторів, що впливають на якість відео. Додаткові фактори, такі як кодеки, стиснення, формат файлу та інші параметри, також можуть значно

впливати на якість відео та його бітрейт. Тому важливо враховувати всі ці фактори при розробці та оптимізації відео-систем та сервісів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Adaptive Multimedia Streaming. URL: <https://www.intechopen.com/books/recenttrends-in-communication-networks/a-survey-on-adaptive-multimedia-streaming> (дата звернення 26.04.2023)
2. David Austerberry. The technology of video and audio streaming. Tatlor&Francis Group. 2005. 356 с.
3. Wes Simpson. Video Over IP: IPTV, Internet and Video, H.264, P2P, Web TV, and Streaming: A Complete Guide to Understanding the Technology. Routledge. 2008. 518 с.
4. Ієрархія сервера. URL: <https://www.akamai.com/glossary/what-is-a-cdn> (дата звернення 26.04.2023).
5. Li BLB, Yin HYN. Peer-to-peer live video streaming on the internet: Issues, existing approaches, and challenges [Peer-to-Peer Multimedia Streaming]. IEEE Communications Magazine. 2007. №;45(6). с. 94-99
6. Zheng Y, Lin F, Yang Y, Gan T. Adaptive resource scheduling mechanism in P2P file sharing system. Springer. Peer-to-Peer Networking and Applications. 2020. V. 13. Issue 6.
7. Karayer E, Sayit M. A path selection approach with genetic algorithm for P2P video streaming systems. Multimedia Tools and Applications. 2015. № 75(23):16039-16057
9. RTSP.URL: <https://wave.video/ua/blog/rtmp-vs-rtsp/> (дата звернення 26.04.2023).
10. OBS Studio. URL: <https://obsproject.com/> (дата звернення 26.04.2023).
11. YouTube Live Streaming & Premieres. URL: <https://www.youtube.com/howyoutubeworks/product-features/live/> (дата звернення 26.04.2023).
12. Iain E. G. Richardson. Video Coding for Next-generation Multimedia. URL:

https://www.researchgate.net/publication/44512776_H264_and_MPEG-4_Video_Compression_video_coding_for_next-generation_multimedia_Iain_E_G_Richardson (дата звернення 26.04.2023).

13. Твердохліб В.В., Юрченко В.Р., Ольховський В.І., Дяченко Д.А., «Аналіз алгоритмів балансування бітової швидкості відео трафіку реального часу»
<https://openarchive.nure.ua/server/api/core/bitstreams/cc3fd885-b4af-48ee-b768-2a18cc23b292/content> (дата звернення 26.04.2023).

14. Node.js Documentation: <https://nodejs.org/en/docs/> (дата звернення 26.04.2023).

15. FFmpeg Documentation: <https://ffmpeg.org/documentation.html> (дата звернення 26.04.2023).