

Міністерство освіти і науки України
Державний вищий навчальний заклад
«Національний гірничий університет»

Інститут електроенергетики
Факультет інформаційних технологій
Кафедра безпеки інформації та телекомунікацій

ПОЯСНЮВАЛЬНА ЗАПИСКА
дипломної роботи

магістра
(ступінь підготовки)

галузь знань 17 Електроніка та телекомунікації
(шифр і назва галузі знань)

напрямок підготовки
(спеціальність) 172 Телекомунікації та радіотехніка
(код і назва напрямку підготовки)

спеціалізація
(освітня програма) Телекомунікаційні системи та мережі
(код і назва спеціалізації)

ступінь підготовки магістр
(назва ступеня підготовки)

кваліфікація 2144.2 Інженер в галузі електроніки та телекомунікацій
(код і назва кваліфікації)

на тему: Модернізація телекомунікаційної системи онлайн-трансляцій,
яка працює в режимі реального часу

Виконавець: студент 2 курсу, групи 172м-16-1

Бублик Юлія Олександрівна
(підпис) (прізвище ім'я по-батькові)

Керівники	Прізвище, ініціали	Оцінка	Підпис
роботи	д.т.н., проф. Корнієнко В.І.		
розділів:			
спеціальний	к.ф.-м.н., доцент Магро В.І.		
економічний	к.е.н., доцент Романюк Н.М.		

Рецензент	д.т.н., проф. Алексєєв М.О.		
-----------	-----------------------------	--	--

Нормоконтроль			
---------------	--	--	--

Дніпро
2018

Міністерство освіти і науки України
Державний вищий навчальний заклад
«Національний гірничий університет»

Інститут електроенергетики
Факультет інформаційних технологій
Кафедра безпеки інформації та телекомунікацій

ЗАТВЕРДЖЕНО:

завідувач кафедри
безпеки інформації та телекомунікацій
_____ Корнієнко В.І.

«_____» _____ 2017 року

ЗАВДАННЯ

на виконання кваліфікаційної роботи магістра
спеціальності _____ *172 Телекомунікації та радіотехніка*
(код і назва спеціальності)

студенту _____ *172м-16-1* _____ *Бублик Юлії Олександрівні*
(група) (прізвище ім'я по-батькові)

Тема дипломної роботи _____ *Модернізація телекомунікаційної системи
онлайн-трансляцій, яка працює в режимі реального часу*

1 ПІДСТАВИ ДЛЯ ПРОВЕДЕННЯ РОБОТИ

Наказ ректора Державного ВНЗ «НГУ» від 26 грудня 2017 р. № 2127-Л

2 МЕТА ТА ВИХІДНІ ДАНІ ДЛЯ ПРОВЕДЕННЯ РОБІТ

Об'єкт досліджень _____ *телекомунікаційна система*

Предмет досліджень _____ *телекомунікаційна система онлайн-трансляцій*

Мета НДР _____ *модернізація телекомунікаційної системи онлайн-
трансляцій з метою зменшення затримки передачі відеопотоку при
збереженні HD-якості зображення; створення на базі нової платформи для
медіасерверу можливості автоматизації деяких процесів і функцій*

переспрямування потоків на популярні онлайн-платформи, зокрема YouTube і Facebook.

Вихідні дані для проведення роботи: застаріла система онлайн-трансляцій, яка працює на базі ПЗ NGINX RTMP module, має тривалу затримку онлайн-трансляцій і не здатна переспрямовувати відеопотік на популярні онлайн-платформи, зокрема YouTube і Facebook.

3 ОЧІКУВАНІ НАУКОВІ РЕЗУЛЬТАТИ

Наукова новизна створення на базі API власного доповнення до ПЗ медіасерверу для виконання додаткових вимог роботи

Практична цінність: модернізація за наведеним проектним рішенням систем онлайн-трансляцій у великих інформаційних і спортивних мас-медіа-компаніях Дніпропетровщини.

4 ВИМОГИ ДО РЕЗУЛЬТАТІВ ВИКОНАННЯ РОБОТИ

Затримка передачі відеозображення під час онлайн-трансляцій при збереженні HD-якості не повинна перевищувати 1,5 с.

5 ЕТАПИ ВИКОНАННЯ РОБІТ

Найменування етапів робіт	Строки виконання робіт (початок-кінець)
1. Огляд існуючих систем онлайн-трансляцій	15.09.17-15.10.17
2. Розробка системи онлайн-трансляцій	16.10.17- 25.12.17
3. Вступ, висновки, реферат	26.12.17-04.01.18
4. Розробка мультимедійної презентації	05.01.18-15.01.18

6 РЕАЛІЗАЦІЯ РЕЗУЛЬТАТІВ ТА ЕФЕКТИВНІСТЬ

Економічний ефект Зменшення експлуатаційних витрат на утримання і обслуговування системи онлайн-трансляцій майже втричі

Соціальний ефект можливість користувачів з будь-якої точки світу спостерігати онлайн-трансляції спортивних змагань чи інших заходів в режимі реального часу автоматизованого керування деякими процесами.

7 ДОДАТКОВІ ВИМОГИ

На базі обраної для модернізації програмної платформи мають бути реалізовані усі технічні можливості, що відповідають сучасним потребам телекомунікаційної компанії, а також організована опція рау-реґ-віеу і можливість

Завдання видав _____
(підпис)

Маґро В.І.
(прізвище, ініціали)

Завдання прийняла
до виконання _____
(підпис)

Бублик Ю.О.
(прізвище, ініціали)

Дата видачі завдання: 14 вересня 2017 року

Термін подання дипломної роботи до ДЕК: 15 січня 2017 року

РЕФЕРАТ

Пояснювальна записка: 88 с., 28 рис., 5 табл., 5 додатків, 30 джерел.

Об'єкт дослідження: телекомунікаційна система.

Мета дипломної роботи: мінімізація затримки передачі відеопотоку при збереженні якості зображення; створення на базі нової платформи для медіасерверу можливості автоматизації деяких процесів і переспрямування потоків на популярні онлайн-платформи, зокрема YouTube і Facebook.

Методи дослідження: історичний метод, системний підхід, методи порівняння, структурний аналіз, метод індукції, експеримент.

В спеціальній частині обрана, налаштована і протестована програмна платформа для модернізації системи онлайн-трансляцій; затримка передачі відеопотоку мінімізована до 1,5 с при збереженні HD-якості зображення.

У роботі досліджено історію розвитку медіасерверних ПЗ, обрана платформа, що відповідає сформованим технічним вимогам. До неї на базі API написане програмне доповнення для автоматизації процесів керування платформою і додатковий функціонал для інтеграції в систему компоненту пауз-пер-вію; організовані онлайн-трансляції з круговим оглядом у 360 градусів.

В економічному розділі визначені капітальні витрати на модернізацію системи онлайн-трансляцій, розрахований економічний ефект від зменшення експлуатаційних витрат на утримання системи онлайн-трансляцій.

Практичне значення дипломної роботи: за наведеним проектним рішенням модернізовані системи онлайн-трансляцій у великих інформаційних і спортивних мас-медіа-компаніях Дніпропетровщини.

Наукова новизна: власне доповнення ПЗ платформи.

Напрямок подальших досліджень: supplementing the online broadcast system with the latest options that will arise every following year.

ОНЛАЙН-ТРАНСЛЯЦІЯ, МЕДІАСЕРВЕР, ПРОТОКОЛ RTMP, ПРОТОКОЛ HLS, ПРОГРАМНА ПЛАТФОРМА, API, ЗАТРИМКА ОНЛАЙН-ТРАНСЛЯЦІЙ, ВІДЕОПОТІК.

ABSTRACT

Explaining note: 88 pages, 28 fig.s, 5 tabl., 5 appen., 30 sour.

The object of research: telecommunication system.

The purpose of the diploma work: the delay minimization of the video stream transfer while preserving image quality; creation the possibility of automating some processes and redirecting flows to popular online platforms, particularly YouTube and Facebook, on the basis of the new platform for media server.

Methods of the research: historical method, system approach, methods of comparison, structural analysis, method of induction, experiment.

The special part consists of the selected, customized and tested software platform for modernization of the online broadcasting system; video stream delay is minimized to 1.5 s while maintaining the HD-quality of the image.

In the work the history of media server software development is studied, the platform which corresponds to the formed technical requirements is chosen. On the API basis the software add-on for automating the processes of platform management and additional functionality for system integration into pay-per-view component are written; 360 degree round-trip online broadcasting is organized.

The economic section defines the capital expenditures for the online broadcast system modernization, the economic effect of operating costs reducing of maintaining an online broadcast system is estimated.

The practical importance of the degree work: based on the design solution, online broadcasting systems are modernized in the big information and sport mass-media companies of Dnipropetrovsk region.

Scientific novelty: own addition to the platform software.

The direction of future research: the addition of broadcast option from mobile devices to the online broadcast system.

ONLINE BROADCAST, MEDIA SERVER, RTMP PROTOCOL, HLS PROTOCOL, SOFTWARE PLATFORM, API, ONLINE BROADCAST DELAY, VIDEO STREAM.

РЕФЕРАТ

Пояснительная записка 88 с., 28 рис., 5 табл., 5 прилож., 30 источников.

Объект исследования: телекоммуникационная система.

Цель дипломной работы: минимизация задержки передачи видеопотока при сохранении качества изображения; создание на базе новой платформы для медиасервера возможности автоматизации некоторых процессов и перенаправления потоков на онлайн-платформы YouTube и Facebook.

Методы исследования: исторический метод, системный подход, методы сравнения, структурный анализ, метод индукции, эксперимент.

В специальной части выбрано, настроено и протестировано ПО для модернизации системы онлайн-трансляций; задержка передачи видеопотока минимизирована до 1,5 с при сохранении HD-качества изображения.

В работе исследована история развития медиасерверных ПО, выбрана платформа, соответствующая сформированным техническим требованиям. К ней на базе API написано программное дополнение для автоматизации процессов управления платформой и дополнительный функционал для интеграции в систему компонента *pay-per-view*; организованы онлайн-трансляции с круговым обзором в 360 градусов.

В экономическом разделе определены капитальные затраты на модернизацию системы онлайн-трансляций, рассчитан экономический эффект от уменьшения эксплуатационных расходов системы онлайн-трансляций.

Практическое значение дипломной работы: по приведенному проектному решению модернизированные системы онлайн-трансляций в больших информационных и спортивных масс-медиа-компаниях Днепра.

Научная новизна: собственное дополнение ПО платформы.

Направление дальнейших исследований: дополнение системы онлайн-трансляций всё новыми опциями, которые будут возникать из года в год.

ОНЛАЙН-ТРАНСЛЯЦИЯ, МЕДИАСЕРВЕР, ПРОТОКОЛ RTMP, ПРОТОКОЛ HLS, ПРОГРАММНАЯ ПЛАТФОРМА, API, ЗАДЕРЖКА ОНЛАЙН-ТРАНСЛЯЦИЙ, ВИДЕОПОТОК.

СПИСОК УМОВНИХ СКОРОЧЕНЬ

- AAC – (англ. Advanced Audio Coding) формат аудіофайлу зі втратами; багатоканальний алгоритм кодування аудіо, який підтримує потокову передачу;
- API – (англ. Application Programming Interface) інтерфейс прикладного програмування;
- CBR – (англ. constant bitrate) постійний бітрейт;
- CDN – (англ. Content Delivery Network) мережа доставки контенту;
- DRM – (англ. Digital Rights Management) «Керування цифровими правами»; технічні засоби захисту авторських прав;
- DVR – (англ. Digital Video Recorder) – пристрій, призначений для запису, зберігання та відтворення відеоінформації;
- FFMPEG – набір безкоштовних бібліотек с відкритим висхідним кодом, які дозволяють записувати, конвертувати і передавати цифрові аудіо- і відеозаписи в різноманітних форматах;
- GOP – (англ. Group Of Pictures) групи зображень;
- H264 – стандарт стискання відео;
- HD – (англ. High Definition) висока чіткість зображення і/або аудіо;
- HDD – (англ. hard (magnetic) disk drive) накопичувач на жорстких магнітних дисках;
- HDMI – (англ. High Definition Multimedia Interface) інтерфейс для мультимедіа високої чіткості;
- HDS – (англ. HTTP Dynamic streaming) процес ефективної доставки потокового відео користувачам шляхом динамічного перемикання між різними потоками різної якості та розміру під час відтворення;
- HLS – (англ. HTTP Live Streaming) комунікаційний протокол для

- потокової передачі медіа на базі HTTP;
- HTTP – (англ. HyperText Transfer Protocol) протокол передачі гіпертексту;
- ID – ідентифікатор користувача;
- IP – (англ. Internet Protocol) міжмережевий протокол;
- IPTV – (англ. Internet Protocol Television) телебачення за Інтернет-протоколом, інтерактивне телебачення;
- MMS – (англ. Multimedia Messaging Service) сервіс мультимедійних повідомлень;
- MP3 – формат файлу для зберігання аудіо-інформації;
- MPEG-DASH – (англ. Moving Picture Experts Group Dynamic Adaptive Streaming over HTTP) – технологія адаптивної потокової передачі даних;
- MPEG-TS – (англ. Moving Picture Experts Group 2 – Transport Stream) протокол передачі відео і аудіо;
- ON2VP6 – відеокодек, створений компанією «On2 Technologies»;
- OSI – (англ. Open Systems Interconnection basic reference model) базова еталонна модель взаємодії відкритих систем;
- OTT-TV – (англ. Over the Top Television) — метод надання відеопослуг через мережу Інтернет, частина технології IPTV;
- PAL – (англ. Phase Alternating Line) система аналогового кольорового телебачення;
- RPC – (англ. Remote Procedure Call) виклик віддалених процедур;
- RTMP – (англ. Real Time Messaging Protocol) протокол потокової передачі даних;
- RTMPE – (англ. Encrypted Real Time Messaging Protocol) протокол RTMP з додатковим рівнем безпеки;
- RTMPS – RTMP-протокол із шифруванням, який використовують для

- передачі TLS/SSL- каналами;
- RTMPT – (англ. Real Time Messaging Protocol Tunneled version) «тонельна» версія RTMP, розроблена для обходу суворі корпоративні брандмауери, відправляючи потокові пакети через стандартний веб-порт (порт 80), «загорнуті» в http-пакети;
- RTP – (англ. Real-time Transport Protocol) протокол, що використовується для передачі аудіо і відео даних через IP-мережі в режимі реального часу;
- RTSP – (англ. Real Time Streaming Protocol) поточковий протокол реального часу;
- SDI – (англ. Serial Digital Interface) послідовний цифровий інтерфейс;
- SRTP – (англ. Secure Real-time Transport Protocol) безпечний протокол передачі даних в режимі реального часу;
- SSL – (англ. Secure Sockets Layer) криптографічний протокол «транспорт захищених сокетів», попередник TLS;
- TCP – (англ. Transmission Control Protocol) протокол керування передачею;
- TLS – (англ. Transport Layer Security) криптографічний протокол захисту транспортного рівня;
- UDP – (англ. User Datagram Protocol) протокол датаграм користувача;
- VBR – (англ. variable bitrate) змінний бітрейт;
- VP9 – відкритий і безкоштовний для користувачів стандарт стискання відео;
- VR – (англ. Virtual Reality) віртуальна реальність;
- WebRTC – (англ. real-time communications) «комунікації в реальному часі»; проект з відкритим висхідним кодом,

призначений для організації передачі потокових даних між браузерами або іншими додатками за технологією «точка-точка»;

- WMS – веб-інтерфейс адміністратора медіасерверу;
- AЗ – апаратне забезпечення;
- БД – база даних;
- ОП – оперативна пам'ять;
- ОС – операційна система;
- ПЗ – програмне забезпечення;
- ПК – персональний комп'ютер;
- ЦПП – центральний процесорний пристрій.

ЗМІСТ

	с.
ВСТУП.....	14
РОЗДІЛ 1. СТАН ПИТАННЯ.....	17
1.1 Структура телекомунікаційної системи онлайн-трансляцій, що працює в режимі реального часу	17
1.1.1 Дослідження затримки, що виникає на всіх структурних елементах телекомунікаційної системи онлайн-трансляцій	19
1.2 Історія створення, розвитку та становлення програмних платформ для медіасервісів	22
1.3 Огляд можливостей старої медіасерверної платформи NGINX RTMP module, що функціонує на базі україно-польської компанії «Visionmark».....	26
1.4 Технічні вимоги до нової програмної платформи	28
1.5 Висновки до першого розділу.....	29
1.6 Постановка задачі.....	31
РОЗДІЛ 2. СПЕЦІАЛЬНА ЧАСТИНА.....	33
2.1 Аналіз запропонованих на ринку рішень..	33
2.1.1 Огляд платформи Wowza Streaming Engine	34
2.1.2 Огляд платформи Nimble Streamer	36
2.1.3 Обрання платформи	37
2.2 Налаштування медіасерверу на базі платформи Wowza Streaming Engine... ..	38
2.2.1 Налаштування прийому і ретрансляції відеопотоку	38
2.2.2 Мінімізація затримки і «тонке» налаштування платформи.....	41
2.2.3 Налаштування веб-сторінки з плеєром для відображення відеопотоку на різних платформах	43
2.2.4 Моніторинг системи за параметрами завантаженості ЦПП та оперативної пам'яті на предмет відмовостійкості та доцільного використання ресурсів	44

2.3 Створення програмного доповнення на базі API для автоматизації процесів керування платформою.....	48
2.4 Організація на базі API додаткового функціоналу (модуля), що дозволяє інтегрувати в систему онлайн-трансляцій компонент рау-per-view, розроблений сторонніми програмістами	51
2.5 Організація онлайн-трансляцій з круговим оглядом у 360 градусів	60
2.5.1 Обрання відеокамери	61
2.5.2 Обрання програмного забезпечення для відеомовлення з кругозором у 360 градусів.....	62
2.5.3 Рекомендації щодо обрання засобів перегляду сферичного відео.....	63
2.6 Висновки до другого розділу	63
РОЗДІЛ 3. ЕКОНОМІЧНИЙ РОЗДІЛ	65
3.1 Розрахунок капітальних витрат на модернізацію системи онлайн-трансляцій.....	65
3.2 Визначення економічного ефекту, що виникає за рахунок зменшення експлуатаційних витрат	67
3.2 Висновки до третього розділу.....	74
ВИСНОВКИ.....	75
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	77
ДОДАТОК А. Перелік документів на оптичному носії.....	80
ДОДАТОК Б. Перелік публікацій з досліджуваної тематики	81
ДОДАТОК В. Лістинг проекту модуля для інтеграції в систему онлайн-трансляцій компоненту рау-per-view	82
ДОДАТОК Д. Відгук керівника економічного розділу	87
ДОДАТОК Е. Відгук керівника дипломної роботи.....	88

ВСТУП

Актуальним питанням, що виникає під час сучасної організації онлайн-трансляцій, є можливість мінімізації затримки передачі відеозображення при збереженні його високої візуальної якості. Здебільшого ця потреба обумовлена онлайн-тоталізатором у сфері спорту. Кожна секунда затримки трансляцій може коштувати користувачам, що роблять ставки, великих грошей.

Більшість телекомунікаційних систем Дніпропетровщини працюють за принципом «клієнт-сервер». Це передбачає наступний ланцюг передачі даних: камера – кодувальний пристрій – медіасервер – декодер – кінцевий користувач (один чи декілька відображувальних пристроїв). На затримку передачі, яку вносить камера чи кінцевий пристрій, майже не можна вплинути, адже вона залежить від технічних можливостей цих пристроїв. Затримка в обох випадках менша за 1мс, отже нею можна знехтувати. Суттєвіша виникає під час кодування відеосигналу та передачі його через медіасервер до кінцевого користувача. Цю затримку можна мінімізувати, однак, у більшості випадків, така мінімізація понижає якість відеозображення, яке транслюється. Головна мета даної роботи – мінімізувати затримку передачі при збереженні якості зображення.

Пріоритетом під час організації системи онлайн-трансляцій є грамотне поєднання існуючих на ринку технічних рішень. Зокрема, це вибір оптимальної ПЗ-платформи для медіасерверу. Крім задоволення основних вимог, що пов'язані зі зростаючими потребами компанії, обрана платформа повинна мати можливість автоматизації деяких процесів та підтримувати переспрямування потоків на популярні онлайн-платформи, зокрема YouTube і Facebook. Додання цих опцій до функціоналу платформи – другорядна мета даної роботи.

Слід додати, що наукові розробки в області оптимізації програмного забезпечення для медіасерверів проводила низка вчених з усього світу, зокрема: В.Ю. Деарт, І.С. Кожухов, О.В. Пилюгін, Y. Liu, B. Du, S. Wang, H. Yang, C.H. Lien, Y.W. Bai, M.B. Lin та ін. Дослідження для даної роботи проведене на базі

українсько-польської компанії «Visionmark», клієнтами якої є багато інформаційних та спортивних агентств. Зокрема, дніпровський регіональний телеканал «Приват ТБ Дніпро – «9 канал», БК «Дніпро» та інтернет-видання «INFORMATOR.DP.UA», де згодом впровадили дану систему онлайн-трансляцій.

Постановка задачі.

Метою кваліфікаційної роботи магістр є мінімізація затримки відеопередачі, осучаснення існуючої системи онлайн трансляцій, запобігання проблемам із масштабуванням при онлайн-трансляції з круговим оглядом у 360 градусів, підвищення якості контенту і забезпечення мультиплатформенності кінцевих пристроїв. Для цього необхідно розв'язати наступні задачі:

- 1) дослідити історію розвитку ринку медіасерверних ПЗ-платформ, виокремити типові недоліки у роботі цих платформ;
- 2) дослідити роботу існуючих платформ для медіа серверів на базі україно-польської компанії «Visionmark»; визначити їх недоліки, що виникли при зростанні технічних вимог до роботи системи в режимі онлайн-трансляції;
- 3) обрати найбільш ефективнішу платформу серед існуючих нині на ринку;
- 4) модернізувати існуючу нерентабельну систему онлайн-трансляцій шляхом заміни і налаштування ПЗ-платформи для медіасерверу;
- 5) створити власне, орієнтоване на потреби телекомунікаційної компанії ПЗ для автоматизації процесів керування платформою і додатковий функціонал, який дозволяє інтегрувати в систему онлайн-трансляцій компонент рау-рег-віу, розроблений сторонніми програмістами;
- 6) на базі модернізованої системи впровадити новітню технологію сферичної онлайн-трансляції з круговим оглядом у 360 градусів;
- 7) розрахувати капітальні витрати на модернізацію системи і визначити економічний ефект, що виникає внаслідок зменшення експлуатаційних витрат.

Об'єкт дослідження – телекомунікаційна система.

Предмет дослідження – процес модернізації телекомунікаційної системи онлайн-трансляцій.

При вирішенні поставлених завдань у дипломному проекті використані: історичний метод, системний підхід, методи порівняння, структурний аналіз, метод індукції, експеримент.

Наукова новизна роботи полягає у власному доповненні програмного забезпечення платформи. Завдяки API впроваджений програмний код для автоматизації процесів керування платформою, а також створений додатковий функціонал, який дозволяє інтегрувати в систему онлайн-трансляцій компонент *pay-per-view*, розроблений сторонніми програмістами.

Практична цінність проекту:

- за рядом показників підвищена ефективність роботи медіасерверів великих інформаційних і спортивних телеканалів та інтернет-видань;
- вперше в регіональних мас-медіа створена можливість онлайн-трансляцій з круговим оглядом у 360 градусів.

Апробація результатів даної роботи полягає на сам перед у практичному застосуванні здобутків. Зокрема, дана модернізація телекомунікаційної системи онлайн-трансляцій була протягом 2017 року втілена у двох комерційних компаніях:

- 1) дніпровському регіональному телевізійному каналі «Приват ТБ Дніпро – «9 канал»;
- 2) Інтернет-виданні «INFORMATOR.DP.UA».

РОЗДІЛ 1

СТАН ПИТАННЯ

1.1 Структура телекомунікаційної системи онлайн-трансляцій, що працює в режимі реального часу

Ланцюг передачі відео під час онлайн-трансляції схематично можна поділити на 6 етапів: зйомку, кодування відеосигналу, передачу від кодуючого пристрою до медіасерверу, передачу від медіасервера до клієнта, декодування та відображення на пристрої користувача (рис. 1.1).

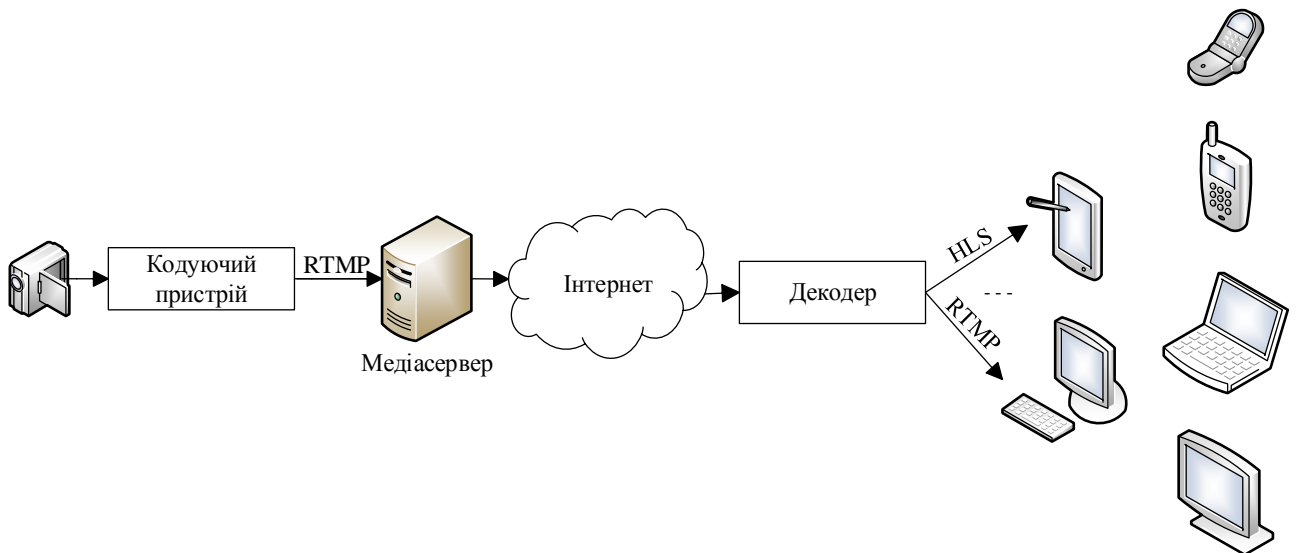


Рисунок 1.1 – Ланцюг передачі відео під час онлайн-трансляцій

Зображення, що перетворюється камерою у відеосигнал SDI, composite або component, в його безпосередньому вигляді не придатне до передачі в каналі зв'язку. Тому перед його передачею з використанням мережевих протоколів є необхідність оцифрувати його у формат, що придатний до подальшого поширення сигналу в мережі. У наш час існує три загальноприйняті формати потокового відео H264, ON2VP9 і MPEG2-TS. З вищезгаданих форматів в системах глобального розповсюдження відео

використовується H264 формат. Тому перед передачею відеосигналу є необхідність перетворити його у відеопотік в H264 форматі без втрати якості або з мінімальною втратою якості, в якості винятку. Для цього використовуються кодуючі пристрої. Це може бути програмне забезпечення або спеціально спроектований пристрій, який керується мікропрограмами. Після кодування відеопотік у зазначеному форматі готовий до передачі вищими рівнями (транспортний – прикладний) моделі OSI. Мережею він передається до медіасервера в одному з можливих мережевих протоколів (UDP, RTSP або RTMP). Медіасервер по суті є основною структурною одиницею, яка розповсюджує прийнятий потік до кінцевих користувачів, і від неї здебільшого залежить часова затримка та якість відеопотоку, що передається. Медіасервери використовуються, зокрема, для таких цілей:

- трансляції мультимедійних потоків у режимі реального часу (live-трансляції) віддаленим користувачам;
- відео за запитом;
- трансляції по плейлистам;
- трансляції з IP-камер.

Завдяки можливостям медіасервера відеопотік, прийнятий ним, до клієнтського пристрою передається у необхідному протоколі (HLS, RTMP, MPEG-DASH, RTSP). Вибір протоколу обумовлений операційною системою клієнта. Затримка кожного із цих протоколів залежить від їхньої характеристики. Існує так звані «повільні протоколи» (MPEG-DASH, HLS), специфіка яких спрямована на кількість кінцевих користувачів, а не на швидкість передачі сигналу. Тривалість затримки у цьому випадку становить 5-30 секунд. «Швидкими» називають протоколи RTMP і RTSP, які використовують у телекомунікаційних системах і в телебаченні зокрема. Їхня затримка обумовлена тільки часом, який витрачається на кодування, мережеву передачу та опрацювання медіасервером. Клієнтський пристрій, приймаючи відеопотік, повинен декодувати його у зображення. Наразі майже всі пристрої

роблять це без вагової затримки – менше, ніж за 1 мс. Тому на цьому етапі затримкою можна знехтувати.

Серед клієнтських пристроїв розрізняють:

- мобільні пристрої з операційними системами Android, iOS, Blackberry, Windows Mobile;
- персональні комп'ютери з різноманітними ОС;
- телевізійні приставки;
- телевізори Smart TV;
- ігрові консолі Sony PlayStation, Xbox.

1.1.1 Дослідження затримки, що виникає на всіх структурних елементах телекомунікаційної системи онлайн-трансляцій

Розглянемо можливу тривалість затримки на кожному етапі передачі відео.

Фіксування відео. Затримка залежить від камери, яку використовують при зйомці. Тривалість цієї затримки – менша за 1 мс, тож нею у розрахунках зазвичай нехтують.

Кодування відео. В сьогоденні стандартом де-факто є відео в H.264 з аудіо в AAC. Робота на цьому етапі в подальшому впливає на весь ланцюг. Перевагу краще віддати апаратним рішенням, адже програмні рішення збільшують час затримки, необхідний для роботи з ресурсами, і тягнуть за собою додаткові навантаження на ресурси операційної системи. Натомість правильно налаштований кодер не додає відчутної затримки, але він задає бітрейт результуючого потоку і його тип. Розрізняють змінний бітрейт (variable bitrate, VBR) і постійний (constant bitrate, CBR). Головна перевага VBR в тому, що він видає потік з найкращим співвідношенням якості зображення і кількості зайнятих даних. Однак для нього потрібна наявність більшої обчислювальної потужності. Крім того, якщо бітрейт в окремий момент часу більший за пропускну здатність каналу, це призведе до буферизації на етапі декодування.

Тому для передачі відео в режимі реального часу з невеликою затримкою рекомендовано використовувати CBR. Втім, з CBR також не все так просто. Насправді постійний бітрейт не є постійним у кожен окремий момент часу, адже потік H.264 містить кадри різної величини. Тому в кодуючому пристрої існує контроль усереднення бітрейту на окремих проміжках часу, щоб об'єм даних був однаковим протягом всієї трансляції. Це усереднення, звичайно, негативно впливає на якість відеозображення. Чим менший період усереднення, тим менший буфер на етапі декодування і тим гірша якість відео, що передається.

Передача від кодуючого пристрою до медіасервера. Затримку на цьому етапі здебільшого визначає робота мережі між кодуючим пристроєм та медіасервером, а також протоколи передачі даних. Для передачі відео в режимі реального часу використовують RTSP, RTMP та UDP протоколи. Деякий час тому найшвидшим та загальноприйнятим вважався UDP протокол через його розташування на транспортному рівні моделі OSI і можливість безадресної передачі. Останнє, в свою чергу, і стало одним з факторів, через які цей протокол не так широко використовується зараз. Тому що при передачі у такий спосіб сторона, яка передає дані, не має підтвердження їхнього отримання адресатом, тобто протокол – без підтвердження і урахування помилок. Згодом був створений RTMP (Real-Time Messaging Protocol), де немає такого недоліку. RTMP – це протокол на базі TCP, який підтримує постійні з'єднання та дозволяє спілкуватися з низькою затримкою. Щоб плавно передавати потоки та передавати якомога більше інформації, він розподіляє потоки на фрагменти, а їх розмір динамічно узгоджується між клієнтом і сервером. Іноді він зберігається незмінним; розмір фрагментів за замовчуванням - 64 байти для аудіоданих, 128 байт – для відеоданих та більшості інших типів даних. Фрагменти з різних потоків можуть потім змішуватись і мультиплексувати через одне з'єднання. За допомогою довгих фрагментів даних протокол виконує лише один байтовий заголовок на кожному фрагменті, тому він має дуже невелику кількість накладних витрат. Проте на практиці окремі фрагменти, як

правило, не змішуються. Замість цього, перемішування та мультиплексування виконується на рівні пакету, причому пакети RTMP через декілька різних активних каналів змішані таким чином, щоб кожен канал відповідав його пропускної здатності, затримці та іншим вимогам якісного обслуговування. Пакети, змішані таким чином, розглядаються як неподільне, і не перетинаються на рівні фрагмента.

RTMP визначає декілька віртуальних каналів, на які можуть надсилатися та прийматися пакети, і які працюють незалежно один від одного. Наприклад, існує канал для обробки запитів і відповідей RPC, канал для даних відеопотоку, канал для даних аудіопотоку, канал для позадіапазонних контрольних повідомлень (узгодження розмірів фрагментів) тощо. Під час типового сеансу RTMP кілька каналів можуть бути активними одночасно в будь-який час. Коли дані RTMP кодуються, створюється заголовок пакета. Пакетний заголовок містить, серед іншого, ідентифікатор каналу, на який він буде відправлений, мітку часу, інформацію про те, коли вона була згенерована (при необхідності), а також розмір корисного навантаження пакету. Цей заголовок передається із корисним вмістом пакета, який фрагментований відповідно до попередньо узгодженого розміру фрагмента. Сам пакетний заголовок ніколи не фрагментується, і його розмір не враховується до даних у першому фрагменті пакета. Іншими словами, лише корисне навантаження пакету (медіа дані) піддається фрагментації.

На більш високому рівні, RTMP інкапсулює MP3 або AAC аудіо та FLV1 відеомультимедійні потоки, і може робити виклики віддаленої процедури (RPC), використовуючи формат Action Message. Усі необхідні служби RPC виконуються асинхронно, використовуючи єдину модель запиту/відповіді клієнта/сервера – таку, при якій спілкування в режимі реального часу не вимагається.

Передача від медіасерверу на клієнтський пристрій. На цьому етапі виникає найбільша затримка. Перший фактор у цьому ланцюзі – буферизація всередині медіасерверу при трансмуksингу (перепакуванні) потоку з одного

протоколу в інший. Другий фактор пов'язаний зі специфікою кожного протоколу.

Протоколи на базі HTTP значно збільшують затримку. Для передачі в реальному масштабі потрібно використовувати протоколи RTMP або RTSP. На другий зазначений фактор вплинути майже неможливо: він пов'язаний зі швидкістю прокладки на рівні локальних провайдерів і проблемами каналів зв'язку мережі Інтернет в міжконтинентальному масштабі, а саме з міжнародними зонами обміну трафіком. Швидкість передачі варто просто додати до загальної величини затримки. Декодер виправить можливі проблеми з буферизацією.

Декодування. Цей етап також сильно впливає на швидкість передачі. Щоб виправити можливу нестачу даних при передачі, буфер програвання повинен містити дані одного повного усередненого періоду з урахуванням затримок мережі. Тому буфер може містити від кількох GOP (group of pictures) до кількох кадрів, залежно від параметрів кодуючого пристрою і стану мережі. Багато плеєрів прирівнюють мінімальне значення буфера програвання до однієї секунди і змінюють його в ході роботи. Мінімумально можливий буфер досягається при використанні апаратних декодерів (плеєрів).

Відображення. На цьому етапі тривалість затримки настільки незначна, що нею можна знехтувати. Все компенсується можливостями пристрою користувача.

Підсумуємо: в ланцюзі передачі відеопотоку вплинути на затримку ми можемо лише на етапі його проходження через медіасервер. Тому для досягнення головної мети дипломної роботи, а саме для мінімізації затримки передачі відео при онлайн-трансляції, необхідно обрати ПЗ для медіасерверу, яке б найкраще відповідало поставленим цілям.

1.2 Історія створення, розвитку та становлення програмних платформ для медіасервісів

Розвиток систем онлайн-мовлення розпочався у 2005 році. Тоді 15 листопада вийшов реліз другої версії медіасерверу Adobe Flash Media Server. Він підтримував можливість стримінгу відео в режимі реального часу на базі відеокодеку ON2VP6. Також до складу релізу входила безкоштовна версія для розробників. Широкомасштабного розвитку технологія онлайн-трансляцій почала набувати приблизно у 2007-2009 роках. Тоді до широкого загалу були представлені наступні продукти:

- безкоштовний медіасервер Red 5, реалізований на Java, перший реліз якого вийшов 4 червня 2009 року;
- Wowza Media Server, перший реліз якого вийшов 19 лютого 2007 року.

Поштовх до розвитку цих медіасерверів надав світ спорту, що потребував миттєвих онлайн-трансляцій матчів у режимі реального часу. Але розвиток технологій був уповільнений низькошвидкісними на той час каналами Інтернету, наприклад, переважна більшість користувачів мали канали менше 1-2 Мбіт/с, а потік прийнятної якості зображення в форматі PAL 720i x 576 у кодеку ON2VP6 стискався до 1,5 – 2,5 Мбіт/с, що було граничним потоком для користувача. Також переважна більшість серверів мала 100-мегабітний канал, що навіть при максимальному навантаженні приводило до того, що один сервер, звичайно, без агрегації каналів підтримував менше сотні користувачів, які дивилися трансляцію у гарній якості. Паралельно із розвитком ПЗ розвивалися апаратні пристрої для того, щоб якомога сильніше стискати відеосигнал без втрати якості. Але вони мали дуже високу ціну. Так наприклад, апаратні кодери компанії «Digital Rapids» з набором кодеків для онлайн-мовлення коштував близько 10 тис. доларів, а його аналоги не менш відомої американської фірми «Haivision» коштували майже 25 тисяч доларів. Це робило використання технологій онлайн-мовлення доступним лише для телебачення. Паралельно з апаратним кодуванням розвивали і програмні кодери. Довгий час майже єдиним і найбільш вагомим було ПЗ, зроблене Adobe Systems (Macromedia Systems) - Adobe Flash Media encoder, перший реліз якого

прийшовся на 23 лютого 2007 року, а майже за рік – в жовтні того самого року вийшов другий реліз, який вже можна було використовувати для роботи. Саме це програмне забезпечення, останній реліз якого (версія 3.2) вийшов 12 грудня 2009 року і підтримувався до 2015. Тож майже шість років це було найбільш розповсюджене ПЗ в сфері онлайн-трансляцій. Також однією з важливих особливостей цього ПЗ було те, що до версії 2.5 кожен мав змогу отримати вихідний код цього додатку. До 2010 року процес розвитку таких технологій сповільнювався тим, що все доступне ПЗ, яке було на той час, потребувало досить великої потужності процесора. Тому коли всередині 2010 року компанія Intel випустила свої перші процесори Core I3/I5/I7, а вже всередині 2011 світ побачили друге покоління цих процесорів з технологією Sandy Bridge, проблема з програмним кодуванням відео була вирішена – широке коло користувачів здобуло можливість кодувати відео в форматі H264 з якістю не менше 1080p. Ще слід додати, що до 2012 майже не існувало серверного ПЗ, яке б мало змогу стабільно працювати під великим навантаженням «з коробки» (тобто з мінімальним налаштуванням). Після 2012 року більш доступними стали широкі канали Інтернету. Один сервер міг мати канал до 16 Гбіт, а створені розробниками серверного ПЗ системи масштабування Origin-Edge дали змогу спокійно будувати системи для багатьох тисяч користувачів по всьому світу з можливістю геооптимізації. В період після 2011 року це привернуло увагу найбільш платіжоспроможної індустрії світу – спорту. Це підштовхнуло виробників ПЗ вдосконалювати свої продукти і значно підвищило конкуренцію на ринку. Деякі проекти зазнавали занепаду, деякі – навпаки, тільки розпочинали свій розвиток. Двома найбільш конкурентноспроможними та функціональними медіасерверами на той час виявилися Wowza Media Server і Adobe Flash Media Server. Вони стали стандартом у виборі ПЗ для побудови систем масових онлайн-трансляцій (більше 10 тис. одночасних підключень). Компанія Adobe намагалась розробити мультизадачний сервер (в тому числі і для онлайн-ігор), а Wowza мала вужчу спеціалізацію, яка була направлена на підтримку майже всіх протоколів для

онлайн-трансляцій (табл. 1.1) (це зробило її популярною для побудови IPTV і OTT-TV) та підтримку відеотрансляцій майже всіма пристроями (навіть ігровими приставками).

З 2014 року кожен з медіасерверів зайняв свою нішу: Adobe Flash Media Server – у сфері онлайн-додатків та ігор, Red5 – у сфері зв'язку в режимі реального часу (онлайн-конференції, відео-чати), Wowza – у сфері онлайн-відеомовлення.

Таблиця 1.1 – Підтримка протоколів різним ПЗ для медіасерверів

Медіасервер Протокол	Flash Media Server	Nimble Streamer	Red5 (open source)	Wowza Streaming Engine
HTTP	Підтримує (HTTP Live Streaming, HTTP Dynamic Streaming)	Підтримує (HTTP Live Streaming, Smooth Streaming)	Підтримує	Підтримує (HTTP Live Streaming, Smooth Streaming, HTTP Dynamic Streaming)
MPEG DASH	?	Підтримує	Не підтримує	Підтримує
WebRTC	Не підтримує	Не підтримує	В розробці	Підтримує
RTSP	Не підтримує	Підтримує	Не підтримує	Підтримує
MMS	Не підтримує	Не підтримує	Не підтримує	Не підтримує
RTP	Не підтримує	Не підтримує	Не підтримує	Підтримує
RTCP	Не підтримує	Не підтримує	Не підтримує	Підтримує
UDP	?	Підтримує	Не підтримує	Підтримує
TCP	?	Підтримує	Підтримує	Підтримує
RTMP	Підтримує	Підтримує	Підтримує (RTMP, RTMPE, RTMPTE, RTMPT, RTMPS)	Підтримує (RTMP, RTMPE, RTMPTE, RTMPT, RTMPS)

Продовження табл. 1.1

Медіасервер Протокол	Flash Media Server	Nimble Streamer	Red5 (open source)	Wowza Streaming Engine
MPEG TS	Не підтримує	Підтримує	Не підтримує	Підтримує
Real Data Transport	?	Не підтримує	Підтримує	Не підтримує
Web sockets	В розробці	В розробці	Підтримує	Не підтримує
HLS	В розробці	Підтримує	?	Підтримує
SRTP	В розробці	В розробці	Не підтримує	Підтримує

1.3 Огляд можливостей медіасерверної платформи NGINX RTMP module, що функціонує на базі україно-польської компанії «Visionmark»

На початку 2012 року перед компанією «Visionmark» стояла задача налаштувати платформу з пропусковою здатністю до 20 тис. одночасних переглядів у HD-якості за якомога менші кошти. В результаті огляду існуючих на той час платформ було обране безкоштовне доповнення до розповсюдженого HTTP-сервера NGINX – RTMP-модуль. Приваблювало у цьому рішенні те, що цей проект був написаний російськомовним автором і мав дуже непогані результати з працездатності під навантаженням. Але виходячи з безкоштовності цього ПЗ, давалась взнаки відсутність постійної технічної підтримки, пояснювальної документації, а також дуже важкий інструмент конфігурації. Всю конфігурацію необхідно було писати у вигляді текстового файлу, після чого перезапускати сервер і перевіряти все знову. На досягнення поставленої мети було витрачено близько трьох місяців постійних тестів та підбирання конфігурацій, що було обумовлене відсутністю детальної документації до ПЗ.

Тоді платформа була встановлена одразу на чотирьох апаратних серверах, з каналами, що мали пропускну здатність 2 Гбіт/с. Структурна схема системи онлайн-трансляції виглядала так, як зображено на рис. 1.2.

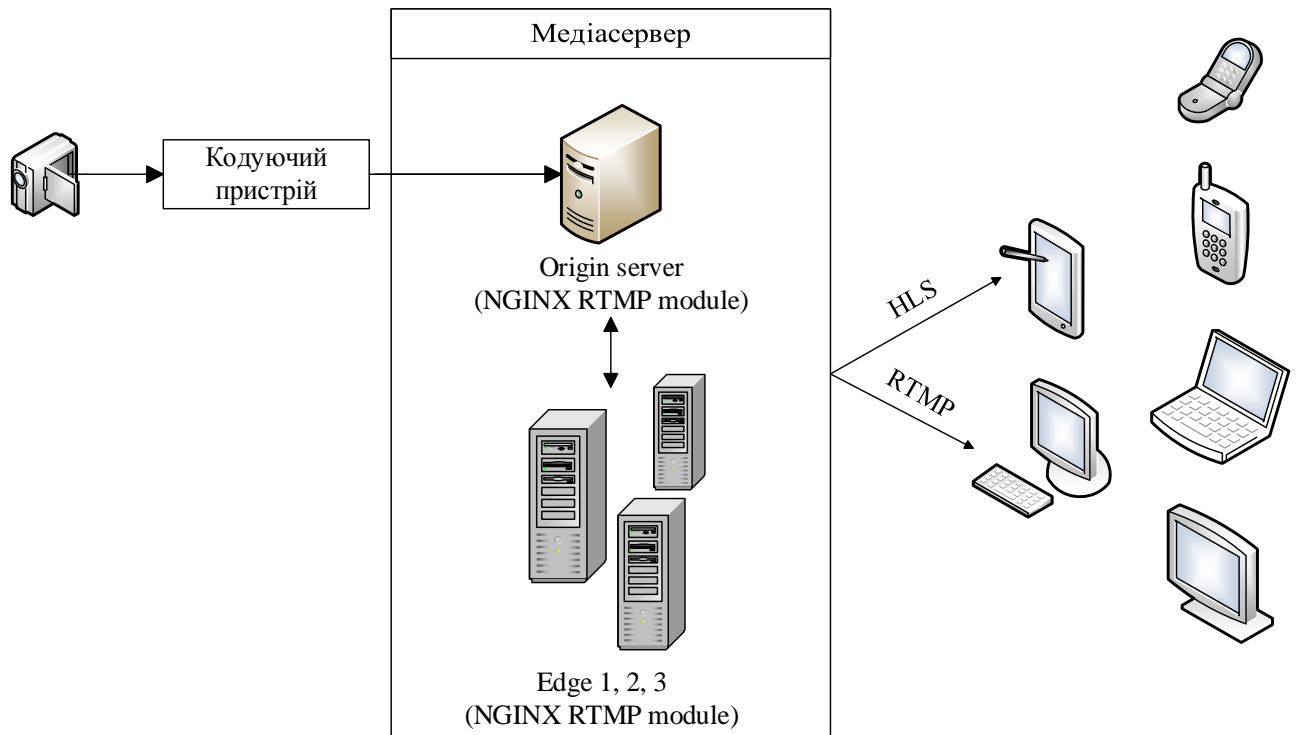


Рисунок 1.2 – Структурна схема онлайн-трансляції на базі платформи NGINX RTMP module

Також однією зі слабких сторін цього рішення було те, що ПЗ було погано адаптоване до роботи з HLS-протоколом, який завдяки компанії Apple став одним з найпоширеніших протоколів в сфері онлайн-мовлення і зараз майже витіснив протокол RTMP на ділянці «сервер - клієнт». Також для реалізації здавалося б простих завдань треба було постійно писати додаткові скрипти та вигадувати власні рішення на основі сторонніх продуктів, наприклад, використання утиліти FFmpeg для трансрейту (створення декількох потоків нижчої якості від основного, спрямоване на користувачів зі слабкими каналами зв'язку). Але варто зазначити, що така ситуація була не лише негативною. Багато чого можна було зробити більш функціональним, ніж

в існуючих на той час версіях медіасерверів інших компаній. Коли стало зрозуміло, що NGINX RTMP module не відповідає сучасним вимогам, а саме: не якісно підтримує HLS-протокол, не дозволяє робити правильне переспрямування потоків на популярний сервіс YouTube, не підтримує можливість трансляції з мобільних пристроїв і, за рахунок не дуже якісної системи транскодування та трансрейту, дуже неефективно використовує процесорну потужність, було прийняте рішення переходу на більш гнучку медіасерверну платформу. Підсумовуючи вищесказане, функціонал NGINX RTMP module підтримує лише два протоколи – RTMP і HLS, при чому останній – не дуже якісно. Такий стан речей був допустимий досить довго, але вже в 2016 році зі зростанням вимог до системи онлайн-мовлення це стало незручним, а вже наприкінці 2016 – недопустимим. Зростання об'ємів трансляцій і неможливість оперативного керування ними (зокрема, зміни напрямків переспрямування потоків чи керування системою через веб-інтерфейс) примусили компанію «Visionmark» шукати вихід з цієї ситуації.

1.4 Технічні вимоги до нової програмної платформи

Враховуючи інформацію, вказану в попередньому підрозділі, а також беручи до уваги зростаючі потреби компанії «Visionmark», були чітко сформульовані технічні вимоги до нової медіасерверної платформи.

Правильно обрана і налаштована платформа повинна не лише мінімізувати затримку при збереженні якості зображення. Вона повинна:

- 1) забезпечувати стабільність передачі (навіть при великій кількості кінцевих користувачів, що перевищує 20 тис.);
- 2) мати високу відмовостійкість при навантаженні – 99,8%;
- 3) мати можливість автоматизації деяких процесів (наприклад, переспрямування потоків на YouTube і Facebook);
- 4) на рівних умовах підтримувати різні платформи кінцевих пристроїв, такі як: мобільні пристрої під керуванням ОС: iOS, Android, Windows тощо, ПК та всі браузері, що використовують на ПК;

- 5) задовольняти наступні критерії безпеки:
 - мати можливість керувати доступом як для користувачів, які відправляють відеопотік у систему, так і керувати доступом до потоку кінцевих користувачів;
 - мати можливість підписання потоку цифровим підписом;
 - віддавати потік лише після введення паролю.
- 6) мати можливість керування системою через web-інтерфейс, де обов'язково повинна відображатися щодо витрачених системою ресурсів, кількості клієнтів (вцілому і по кожному потоку зокрема)
- 7) мати інструменти аналізу, які дозволяють бачити, по яким протоколам яка кількість користувачів дивиться потік в окремий момент часу
- 8) мати можливість транскодувати та робити трансрейт вхідного потоку в режимі реального часу;
- 9) мати можливість приймати вхідний потік не тільки за технологією PUSH, але і за PULL-технологією, тобто коли сервер ініціює з'єднання з пристроєм, який є джерелом відеопотоку і самостійно забирає потік з пристрою, який не має можливості відправляти потоки на медіасервери;
- 10) можливість розгортання системи та її масштабування за рахунок Cloud-технологій;
- 11) мати найменшу часову затримку, що виникає під час трансмуксингу (перепакування пакетів у тимчасову форму для передачі кінцевому користувачеві).

1.5 Висновки до першого розділу

Беручи до уваги все вищесказане, можна зробити висновок:

1 Ланцюг передачі відео під час онлайн-трансляції схематично можна поділити на 6 етапів: зйомку, кодування відеосигналу, передачу від кодуючого пристрою до медіасервера, передачу від медіасервера до клієнта, декодування та відображення на пристрої користувача. Найбільша затримка трансляції відеосигналу виникає на ділянці «кодуючий пристрій – медіасервер

– декодер». Однак вплинути на неї можна лише на етапі проходження через медіасервер, де відеопотік проходить процедуру трансмуksингу – перепакування пакетів даних з одного протоколу в інший. Також мінімізувати затримку на цій ділянці можна вибором швидкого, але надійного протоколу передачі. Таким є протокол RTMP. Саме його варто використовувати на етапі передачі відеосигналу від кодуєчого пристрою до медіасервера.

2 Розвиток систем онлайн-мовлення розпочався у 2005 році з виходу релізу другої версії медіасерверу Adobe Flash Media Server. Двома роками пізніше – у 2007 році – вийшов перший реліз не менш якісного ПЗ від компанії «Wowza» – Wowza Media Server. Ці дві платформи для медіасерверів протягом багатьох років мали першість на ринку і були основними конкурентами в цій галузі. Компанія Adobe намагалась розробити мультизадачний сервер (в т.ч. і для онлайн-ігор), а Wowza мала вужчу спеціалізацію, направлену на підтримку майже всіх протоколів та підтримку відеотрансляцій майже всіма пристроями. В результаті з 2014 року нішу онлайн-мовлення на ринку медіасерверів зайняла друга компанія, тоді як Adobe спрямувала увагу на створення медіасерверів для онлайн-додатків та ігор. Оскільки дана дипломна робота стосується модернізації системи онлайн-мовлення, подальший розгляд ПЗ від компанії Adobe, що спеціалізується на сторонній тематиці, можна опустити.

3 Функціонал колишньої медіасерверної платформи NGINX RTMP module, на якій базувалась система онлайн-трансляцій досліджуваної компанії «Visionmark», підтримувала лише два протоколи – RTMP і HLS, при чому останній – не дуже якісно. Такий стан речей був допустимий досить довго, але наприкінці 2016 року зростання об'ємів трансляцій і неможливість оперативного керування ними примусили телекомунікаційну компанію шукати більш гнучке програмне рішення для медіасерверу. Для цього у даній дипломній роботі було сформовано 11 технічних вимог до нової медіасерверної платформи. Найбільш вагомими серед них є:

- найменша часова затримка, що виникає під час трансмуksингу;
- висока відмовостійкість при навантаженні – 99, 8%;

- можливість автоматизації деяких процесів (зокрема, переспрямування потоків на YouTube і Facebook);
- відповідність критеріям безпеки;
- можливість розгортання системи та її масштабування за рахунок Cloud-технологій;
- підтримка більшості протоколів передачі.

Для вдалої модернізації системи онлайн-трансляцій компанії «Visionmark» необхідно відповісти на наступні питання:

- 1 Які медіасерверні платформи серед запропонованих на ринку задовольняють технічним вимогам компанії?
- 2 Який вагомий фактор вплине на вибір конкретного ПЗ у разі, коли декілька платформ повністю відповідають встановленим вимогам?
- 3 Як правильно налаштувати і протестувати систему?
- 4 Які додаткові функції можна втілити на базі нової платформи для осучаснення системи?

1.6 Постановка задачі

Головною метою даної роботи є мінімізація затримки передачі відеопотоку при збереженні якості зображення. Другорядна мета полягає у створенні на базі нової платформи для медіасерверу можливості автоматизації деяких процесів, підтримки переспрямування потоків на популярні онлайн-платформи, зокрема YouTube і Facebook.

Для цього необхідно виконати наступні задачі:

- 1 Серед поширених медіасерверних платформ обрати такі, що задовольняють технічним вимогам компанії «Visionmark».
- 2 Визначити затримку трансмуксингу для кожної обраної платформи.
- 3 Для модернізації системи онлайн-трансляцій обрати ПЗ з найменшою затримкою.
- 4 Налаштувати і протестувати обрану платформу.
- 5 Звести затримку до мінімуму.

6 На базі API створити програмне доповнення для автоматизації процесів керування платформою та переспрямування потоків на популярні онлайн-платформи YouTube і Facebook.

7 На базі API розробити додатковий функціонал, який дозволяє інтегрувати в систему онлайн-трансляцій компонент рау-рег-view, розроблений сторонніми програмістами.

8 Дослідити організацію онлайн-трансляцій з круговим оглядом у 360 градусів.

РОЗДІЛ 2. СПЕЦІАЛЬНА ЧАСТИНА

2.1 Аналіз запропонованих на ринку рішень

Нині існує декілька основних поширених медіасерверів:

- Adobe Flash Media Server (платний);
- Wowza Media Server (с 2014 года Wowza Streaming Engine) (платний);
- Helix Universal Server;
- Nimble Streamer (безкоштовний);
- Moment Video Server.

Серед запропонованих на ринку ПЗ рішень технічним вимогам, вказаним у підрозділі 1.4, відповідають одразу дві медіасерверні платформи: Nimble Streamer і Wowza Streaming Engine. Оскільки на вибір медіа-платформи вагомо впливає затримка проходження відеопотоку через медіасервер, було прийняте рішення визначити затримку на цих двох платформах емпіричним шляхом.

Враховуючи інформацію, наведену в пункті 1.1.1, були виділені етапи, на які при побудові телекомунікаційної системи передачі відео в реальному часі вплинути майже неможливо. Їхній вибір обумовлений можливостями, необхідними в сучасному телебаченні. Так само це стосується мережі передачі відеопотоку. Швидкість та затримка на цьому етапі визначаються обладнанням та кількістю Інтернет-провайдерів, які задіяні в ланцюзі передачі відеопотоку від кодуючого пристрою до кінцевого користувача через всю телекомунікаційну систему. Виходячи з цього, тестування затримки були проведені в межах мережі одного провайдера, де розташовані кодуючий пристрій, медіасервер і кінцеві клієнти. В якості кодуючого пристрою був використаний програмно-апаратний комплекс VMix, який мало чим поступається апаратним кодерам за тривалістю затримки і якістю відеопотоку. Отже, враховуючи, що ми досягаємо сталих характеристик на етапі кодування, декодування і передачі даних, нам вдасться визначити реальну затримку (на етапі перепакування пакетів) того чи іншого медіасерверу.

Емпіричне фіксування затримки онлайн-трансляцій на базі кожної з платформ виконане методом одночасної фіксації початкового відео на етапі зйомки та кінцевого відео, трансльованого на веб-сайт. Для цього на один з серверів, на яких працювало колишнє ПЗ компанії – NGINX RTMP module, встановлені пробні безкоштовні версії платформ Nimble Streamer і Wowza Streaming Engine. Часова різниця в програванні конкретних зображень вимірювалась секундоміром (рис. 2.1 і рис. 2.2).

2.1.1 Огляд платформи Wowza Streaming Engine

Даний сервер має всього дві версії продукту: одна – безкоштовна, що дозволяє одночасне підключення 10 клієнтів, друга – комерційна версія без будь-яких обмежень. Остання версія продукту – це Wowza Streaming Engine. На відмінну від Adobe Media Server, Wowza має підтримку протоколів мовлення RTP/RTSP, що є значною перевагою в багатьох проектах.

Також до основних переваг і функцій медіасервера Wowza можна віднести:

- трансляцію live-потоків в режимі реального часу в форматі H.264 з можливістю програвання на всіх популярних пристроях і плеєрах, що підтримують один з наступних протоколів: RTMP, HDS, HLS, Smooth Streaming;
- адаптивний бітрейт для live-потоків, реалізований за допомогою модуля Transcoder AddOn або за допомогою сторонніх додатків;
- можливість накладання логотипу (картинки) на відеопотік, що транслюється, за допомогою модуля Transcoder AddOn;
- перемотку живих потоків (DVR);
- запис live-потоків на сервер у форматі *.mp4 чи *.flv; записані файли, можна програвати як відео за запитом;
- роботу з плейлистами - трансляція live-потоків і відеофайлів по плейлисту;

- трансляцію відеопотоків з IP-камер з можливістю запису потоку на медіасервер Wowza у форматі *.mp4 або *.flv;
- підтримку субтитрів для live-трансляцій и для відео за запитом;
- підтримку вибору однієї з декількох звукових доріжок (працює лише для відео за запитом);
- підтримку технології адаптивного бітрейту для відео за запитом;
- можливість шифрування відеопотоку, а також використанні різноманітних сторонніх DRM-систем для захисту відеоконтента;
- обслуговування різноманітних додатків, наприклад, відеочату, й інших мультимедійних інтерактивних додатків, які розроблені спеціально під медіасервер Wowza;
- запис з веб-камер, IP-камер відео на сервер у форматі *.mp4 чи *.flv; записані файли можна програвати як відео за запитом.

Wowza Streaming Engine гідно демонструє себе, як комерційне рішення для трансляцій наживо и для відео за запитом. Також існує можливість самостійно створювати додатковий функціонал серверу за допомогою наданого розробниками API.

Тим часом вирішальний критерій – затримка трансмуksингу для даної платформи – становить 1,475 с (рис. 2.1).



Рисунок 2.1 - Затримка трансмуksингу платформи Wowza Streaming Engine

2.1.2 Огляд платформи Nimble Streamer

Даний сервер є безкоштовним, але такі сервіси, як Web-інтерфейс керування (WMS-панель) і транскодинг, які є одними з найважливіших, доступні лише в оренду. При чому безкоштовну версію WMS-panel можна застосовувати у двадцятиденний термін. По суті, даний медіасервер підтримує такий самий функціонал, як і Wowza Streaming Engine, але його конфігурація без платної WMS-панелі – майже неможлива, що негативно впливає на вибір цього програмного рішення при побудові систем онлайн-трансляцій. Також варто зазначити, що компанія-виробник Nimble Streamer ООО «Софтвелум» розпочала свою діяльність і набула ваги тому, що писала альтернативну панель конфігурацій до Wowza Media Server, а продукт Nimble Streamer як самостійну платформу почала просувати близько трьох років тому.

Затримка трансмуksингу для Nimble Streamer складає 1,756 с (рис. 2.2).



Рисунок 2.2 – Затримка трансмуksингу платформи Nimble Streamer

2.1.3 Обрання платформи

Вимірjana затримка на перепакування пакетів коротша при використанні платформи Wowza Streaming Engine (різниця з Nimble Streamer становить 281 мс). Виходячи з того, що затримка до 500 мс є несуттєвою, на вибір платформи здебільшого вплинули умови ліцензування обох систем. Оскільки Nimble Streamer з WMS-panel не можуть бути передані на баланс компанії, а доступні тільки за підпискою (в оренду), було прийняте рішення використання Wowza Streaming Engine та покупки однієї ліцензії рівня Perpetual для модернізації телекомунікаційної системи онлайн-трансляцій, що працює в режимі реального часу. А для вирішення проблем з піковим навантаженням обрані до використання серверні потужності Wowza Streaming Cloud. Це дозволило зменшити кількість апаратних серверів з чотирьох до одного, тому структурна схема системи тепер виглядає так, як зображено на рис. 2.3:

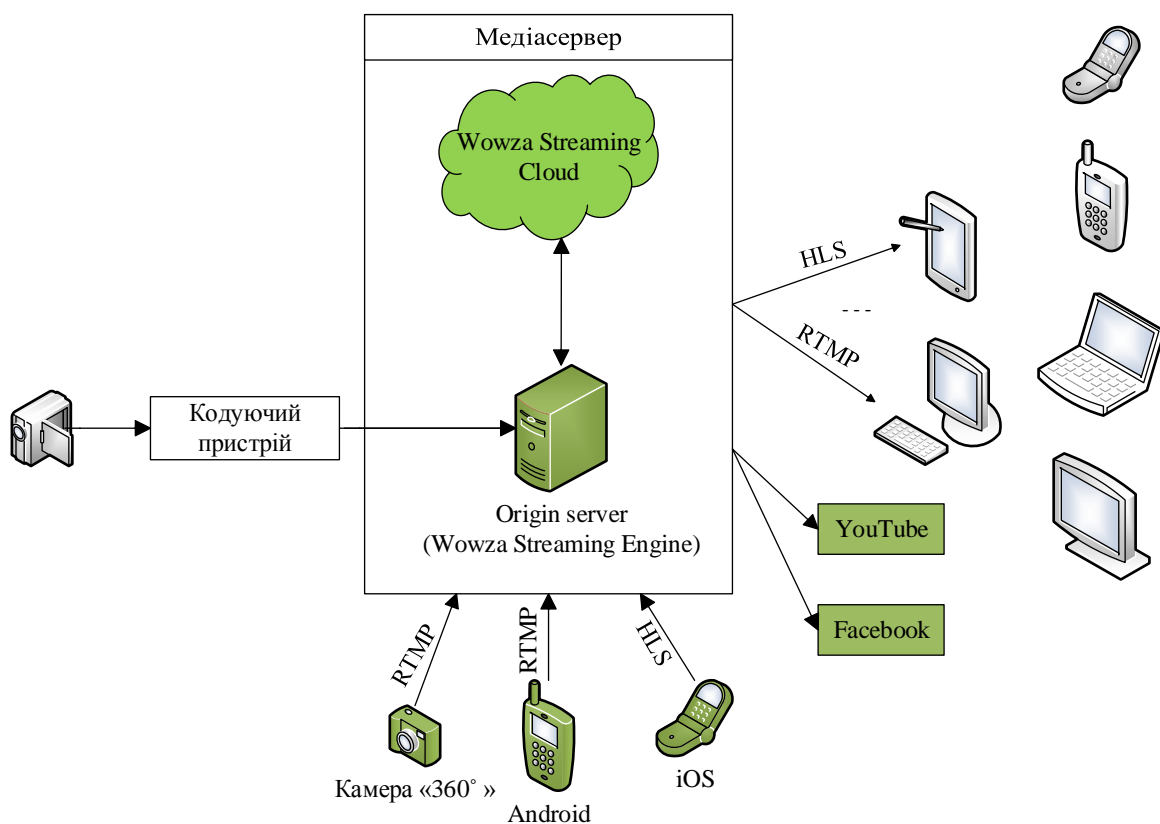


Рисунок 2.3 - Структурна схема системи онлайн-трансляцій після її модернізації

2.2 Налаштування медіасерверу на базі платформи Wowza Streaming Engine

Налаштування медіасерверу Wowza Streaming Engine складається з чотирьох етапів:

- встановлення і базове налаштування ПЗ;
- налаштування прийому відеопотоку за протоколом RTMP і його ретрансляції за протоколами RTMP и HLS;
- налаштування веб-сторінки з плеєром для відображення відеопотоку на платформах Windows (протокол RTMP), iOS (протокол HLS), Android (протокол HLS);
- моніторингу системи за параметрами завантаженості ЦПП та операційної пам'яті на предмет відмовостійкості та доцільного використання ресурсів.

Перший крок – елементарний, тож перейдемо одразу до другого.

2.2.1 Налаштування прийому і ретрансляції відеопотоку

На базі працюючого серверу, використовуючи Web-інтерфейс, який включено до програмного пакету, налаштуємо екземпляр додатку для прийому потоку від клієнта компанії ТОВ «Приват ТБ Дніпро – «9 канал». Спочатку обираємо тип додатку Live – Single Server or origin (рис. 2.4). Саме цей тип використовується для передачі прямих потоків до користувачів (одного сервера) або для передачі прямих потоків на інші сервери, що використовують програмне забезпечення Wowza Media Server або Wowza Streaming Engine, щоб масштабувати доставку контенту великій кількості користувачів. Натомість тип додатку Live Edge використовується для доставки прямих потоків до користувачів (через єдиний сервер), а тип Live HTTP Origin – для передачі прямих потоків до інфраструктури кешування HTTP за допомогою протоколів HTTP потокового передавання (MPEG-DASH, Apple HLS, Adobe HDS, and Microsoft Smooth Streaming).

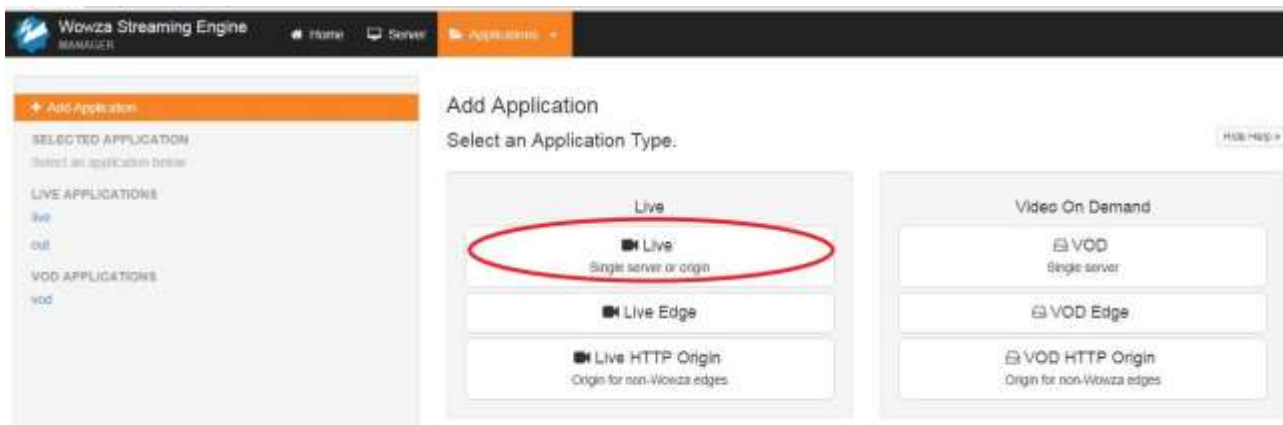


Рисунок 2.4 – Обрання типу додатку

В наступному діалоговому вікні (рис. 2.5) вводимо ім'я екземпляру – в нашому випадку 9tv.

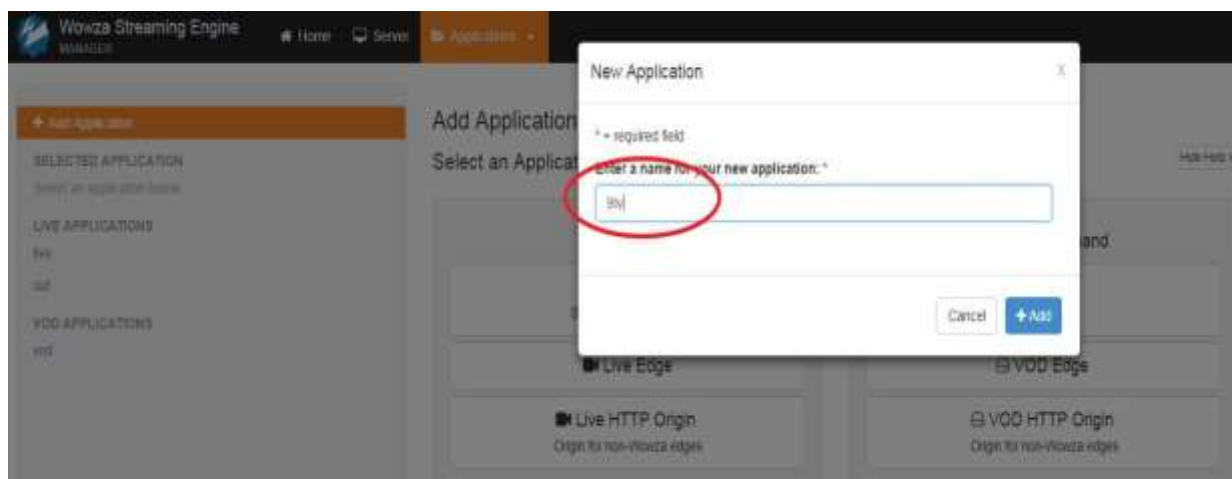


Рисунок 2.5 – Назва додатку

Далі обираємо протоколи вихідних потоків, з якими буде працювати наш екземпляр, та інші параметри екземпляру (рис. 2.6). Зокрема, обираємо всі можливі протоколи, не вмикаємо запис та лишаємо інші налаштування за замовчуванням.

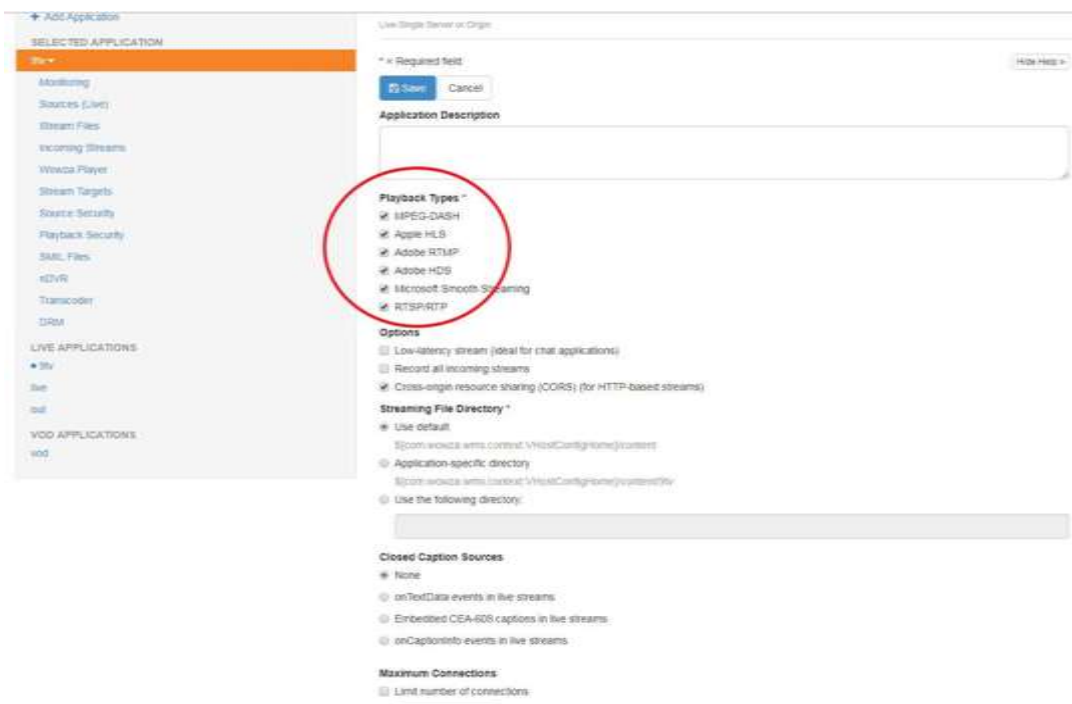


Рисунок 2.6 – Обрання протоколів вихідних потоків

Наступний крок (рис. 2.7) – обрання параметрів безпеки для екземпляру, а саме паролю на аутентифікацію, щоб убезпечити себе від прийому несанкціонованих потоків та втручання в потік клієнта.

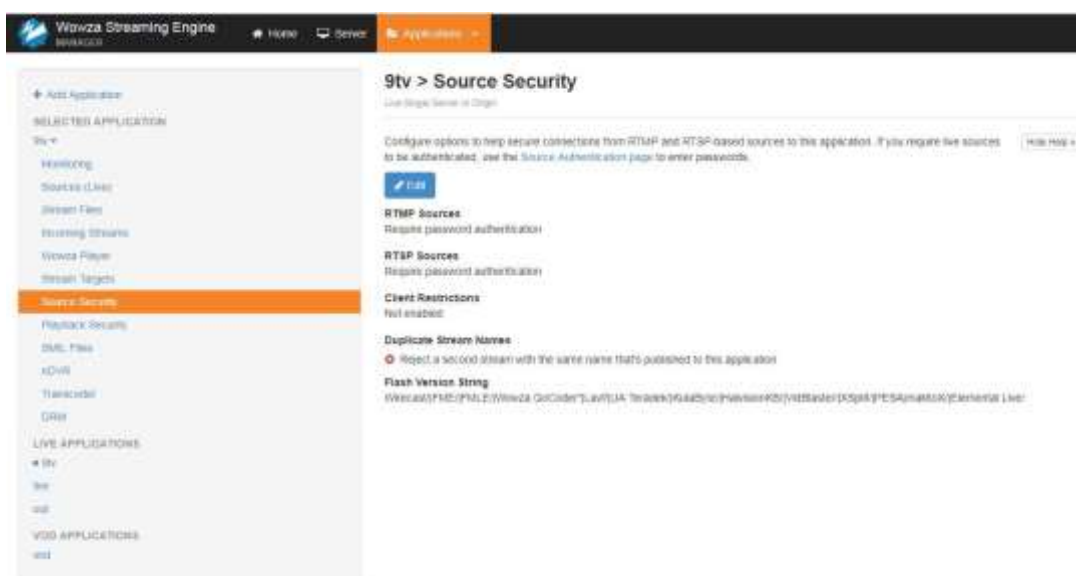


Рисунок 2.7 – Обрання параметрів безпеки

2.2.2 Мінімізація затримки і «тонке» налаштування платформи

Наступним кроком буде «тонке» налаштування платформи у розділі Performance Tuning. Одним з найголовніших на цьому етапі є налаштування параметрів запуску Java, на яких працює платформа (рис. 2.8).

Java Heap – це область пам'яті, в якій об'єкти, створені в класі Java, знаходяться під час виконання програми. Коли дана програма закінчує використання об'єкта, колектор сміття Java повертає пам'ять у Java heap, Програмне забезпечення Wowza Streaming Engine містить декілька варіантів збирання сміття (GC).

Варто зауважити, що за замовчуванням сервер використовує параметр розміру Java Heap для рівня розробки, який підходить для середовищ розробки додатків, але повинен бути змінений, якщо необхідно розгорнути сервер у виробничих середовищах, де він може широко використовувати пам'ять при значному завантаженні. Недостатній параметр розміру Java Heap може призвести до помилки сервера при винятті Java за винятком «OutOfMemory».

Розміри Java Heap:

1 Рівень розвитку. Встановлює розмір Java Heap до заданого значення, яке підходить для середовищ розробки додатків. Це налаштування розміру Java Heap нижче, ніж рекомендовано для медіасерверів у великих обсягах виробничого середовища, які зазвичай сильно завантажуються.

2 Рівень виробництва. Встановлює розмір Java Heap до заданого значення, яке підходить для використання у виробництві.

3 Користувацький рівень. Дозволяє ввести індивідуальне значення розміру Java Heap між 512 і 10240 мегабайтами.

Нажаль, строку з усіма налаштуваннями Java не можна представляти до широкого загалу, адже вона складає корпоративну таємницю компанії «Visionmark».

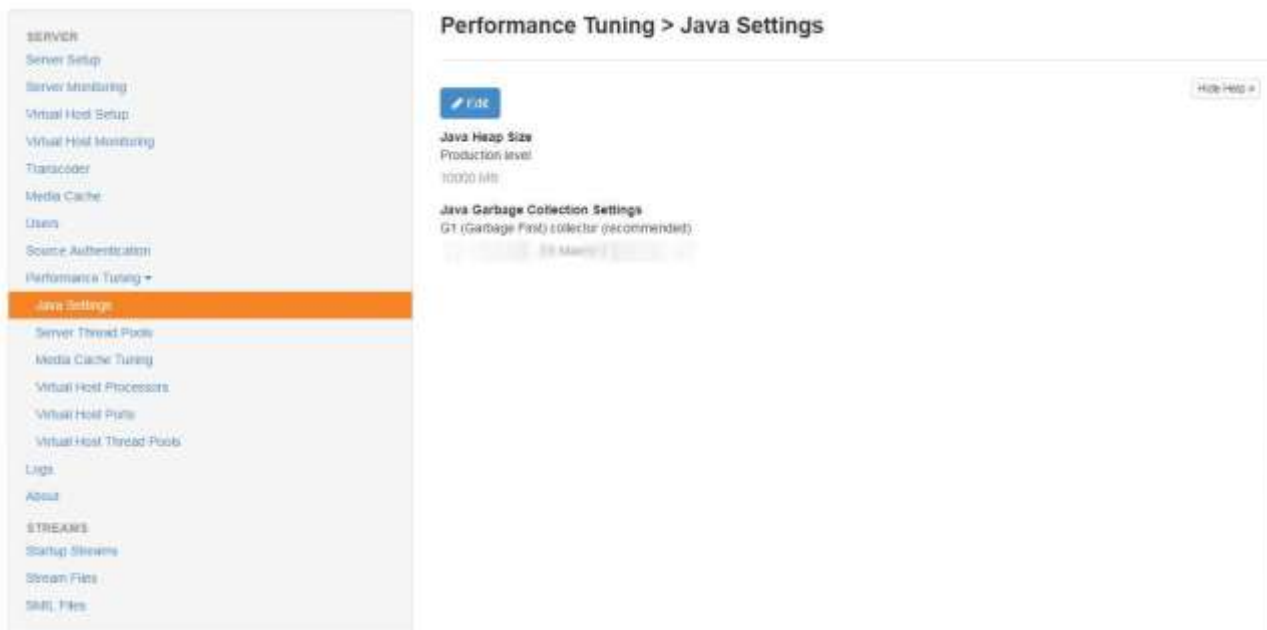


Рисунок 2.8 – Налаштування параметрів запуску Java

Також на параметри затримки впливає розмір кеш-пам'яті та її налаштування. Як видно з рис. 2.9, ми маємо 2 параметри: один впливає на кількість вхідних даних, які тримаються в кеші, а другий - на кількість кешу для вихідних потоків.

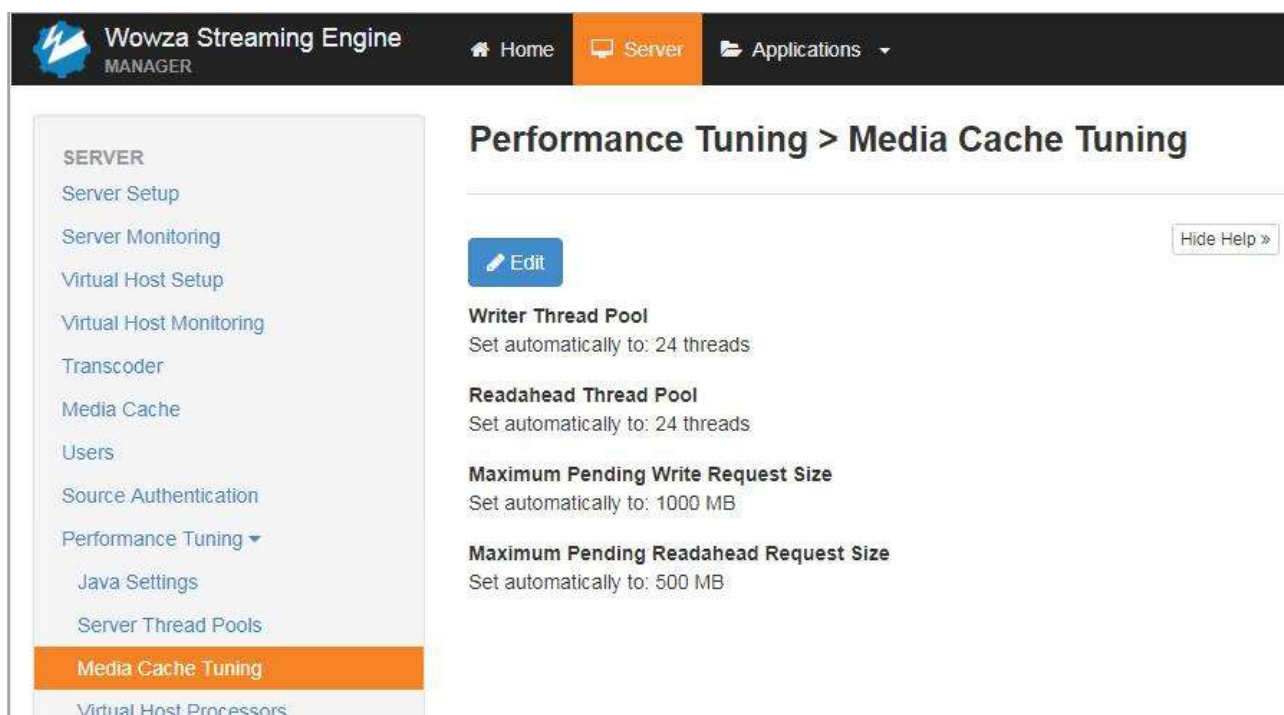


Рисунок 2.9 – Налаштування кешу сервера

Оскільки нас цікавить на даному сервері не кількість підключень, а швидкість передачі даних, ці кеші зменшені у відповідності до цієї вимоги - 1000 та 500 МБіт. Це означає, що вхідні потоки почнуть кешуватися, якщо їхній загальний трафік перевищить 1 ГБіт. При цьому частина користувачів (або всі користувачі при нормальному навантаженні) прийматиме відеопотік в режимі реального часу з мінімальною затримкою (1,5с) без втрати якості, а решта - із запізненням, але зі збереженням HD-якості.

2.2.3 Налаштування веб-сторінки з плеєром для відображення відеопотоку на різних платформах

Створення плеєра для перегляду онлайн-трансляцій є не дуже складним. У даній дипломній роботі воно виконане на базі Wowza Player Builder, де були введені дані потоку і виконана перевірка (рис. 2.10).

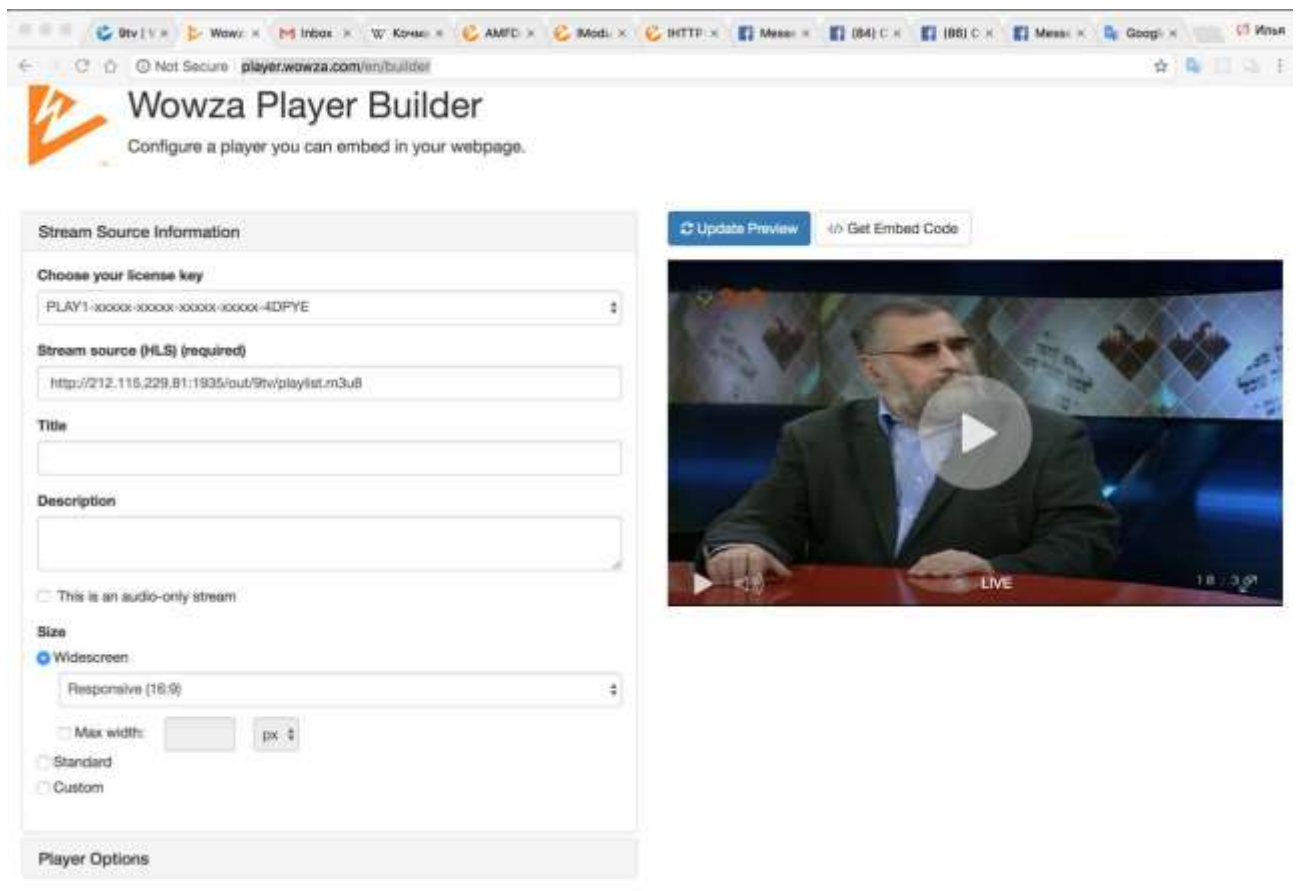


Рисунок 2.10 – Базове налаштування плеєра для перегляду онлайн-трансляцій

Натиснувши Get отримуємо код, який клієнт може інтегрувати на свій сайт (рис. 2.11).



Рисунок 2.11 – Код, який клієнт може інтегрувати на свій сайт

2.2.4 Моніторинг системи за параметрами завантаженості ЦПП та оперативної пам'яті на предмет відмовостійкості та доцільного використання ресурсів

На рис. 2.12, який відображає моніторинг основних параметрів сервера, видно їхнє навантаження у поточний момент часу.

Server Monitoring

All dates/times shown are in your local browser time

Current Connections Incoming and outgoing

Total: 1

Wowza Streaming Engine Uptime

Wowza Streaming Engine up since 24 Dec 2017 06:15:27 PM

Wowza Streaming Engine up for about 23 days

Current Usage

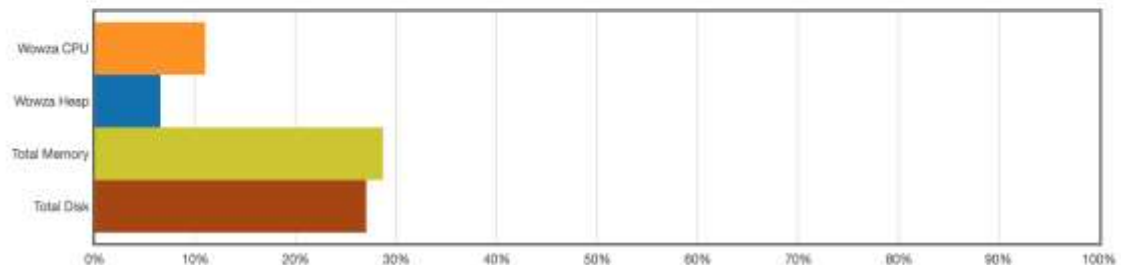


Рисунок 2.12 – Моніторинг серверу за параметрами завантаженості ЦПП та операційної пам'яті

Також у розділі моніторингу можна зробити вибірку окремих параметрів у потрібний інтервал часу. Цю функцію було використано для визначення відмовостійкості сервера. Так на рис. 2.13 видно безперервність вхідного потоку для одного з клієнтів компанії «Приват ТБ Дніпро» за 30 днів. Варто зазначити, що від клієнта приймається один потік, що ретранслюється на Youtube і Facebook. Це видно на рис. 2.14, який ілюструє використання трафіку клієнтом. При цьому величина вихідного трафіку в 2 рази перевищує вхідний.



Рисунок 2.13 – Вхідний потік для одного з клієнтів за тридцятиденний період

Навантаження на канал цього клієнта ілюструє графік на рис. 2.14

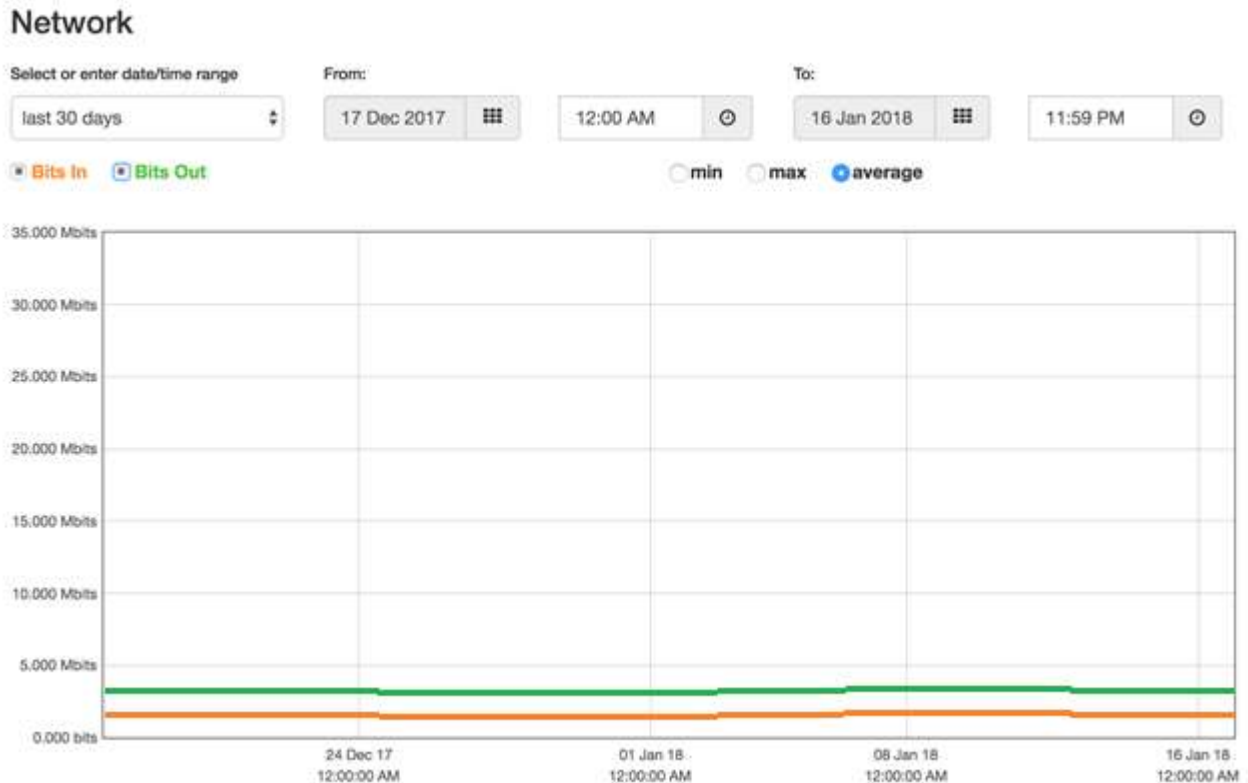


Рисунок 2.14 – Використання трафіку клієнтом

Виходячи з двох вищенаведених графіків (рис. 2.13 і рис. 2.14), можна зробити висновок, що потік надходив до медіасервера стабільно і безперервно протягом останніх 30 днів. Тим часом графіки використання ЦПП та ОП, приведені нижче (рис. 2.15 і рис. 2.16), доводять, що ПЗ працювало стабільно. Це видно з коливань завантаженості ЦПП, які не перевищували 2% протягом обраного періоду. Виділення ОП також були майже незмінними.

Memory

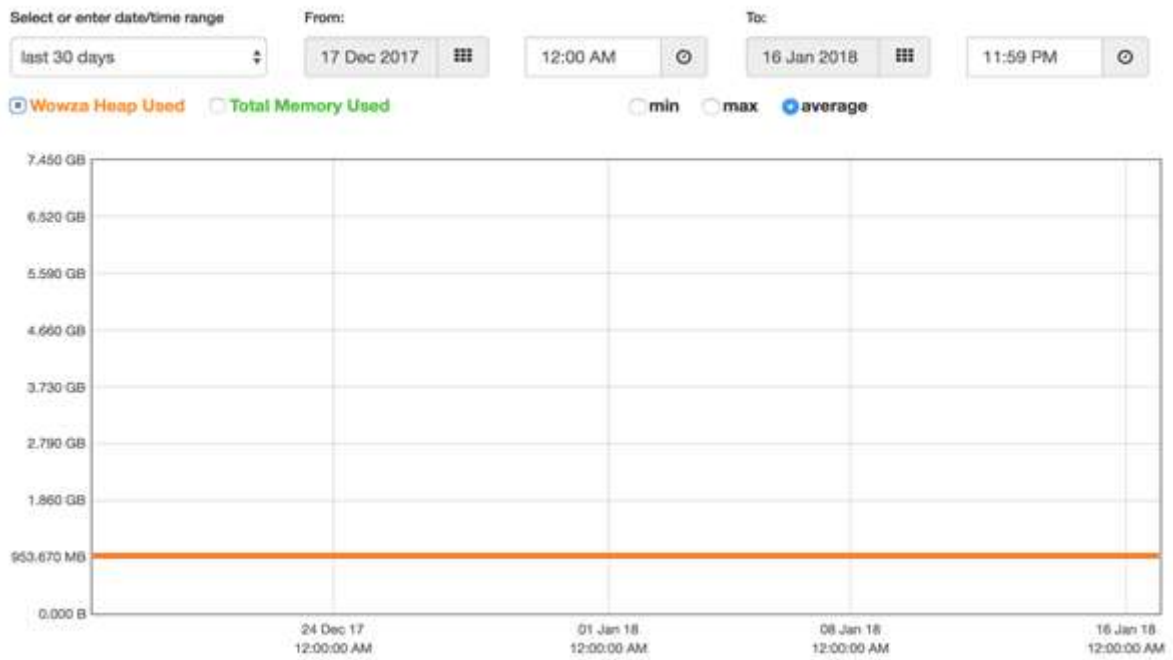


Рисунок 2.15 – Графік використання ОП

Wowza CPU

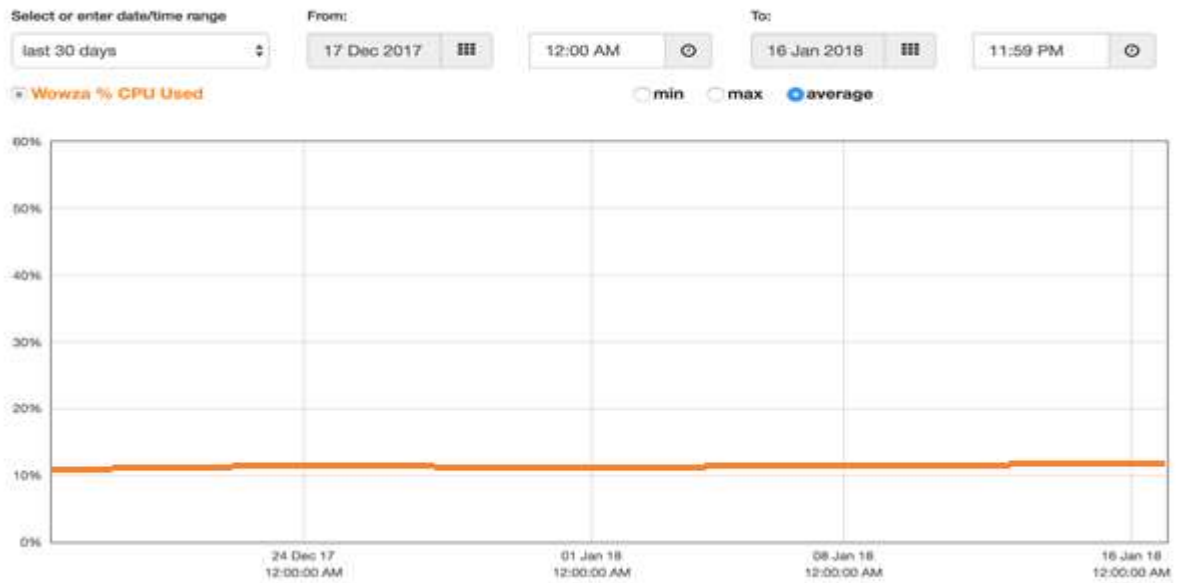


Рисунок 2.16 – Графік використання ЦП

2.3 Створення програмного доповнення на базі API для автоматизації процесів керування платформою

Другорядною метою дипломної роботи є додання до функціоналу платформи опцій автоматизації деяких процесів та переспрямування потоків на популярні онлайн-платформи, зокрема YouTube і Facebook. Втілення цієї мети наведено у даному підрозділі.

Одним з важливих факторів вибору Wowza Media Server був той факт, що до складу цього ПЗ входить API (англ. application programming interface – інтерфейс прикладного програмування), який дозволив написати сторонній програмний код для управління платформою. У пригоді стала обширна документація, що описує даний API, безліч прикладів створених на його базі програм і постійна онлайн-підтримка на сайті виробника.

Маючи на меті автоматизувати процес запуску прямого ефіру 9 телеканалу на платформу YouTube і Facebook, було обрано можливість підключення до медіаплатформи за допомогою вбудованих в Linux утиліт CRON і CURL. Таким чином у вбудованому в ПЗ Wowza веб-інтерфейсі у розділі Stream-targets було створено 2 шаблони, необхідні для переспрямування потоків (рис.2.17).

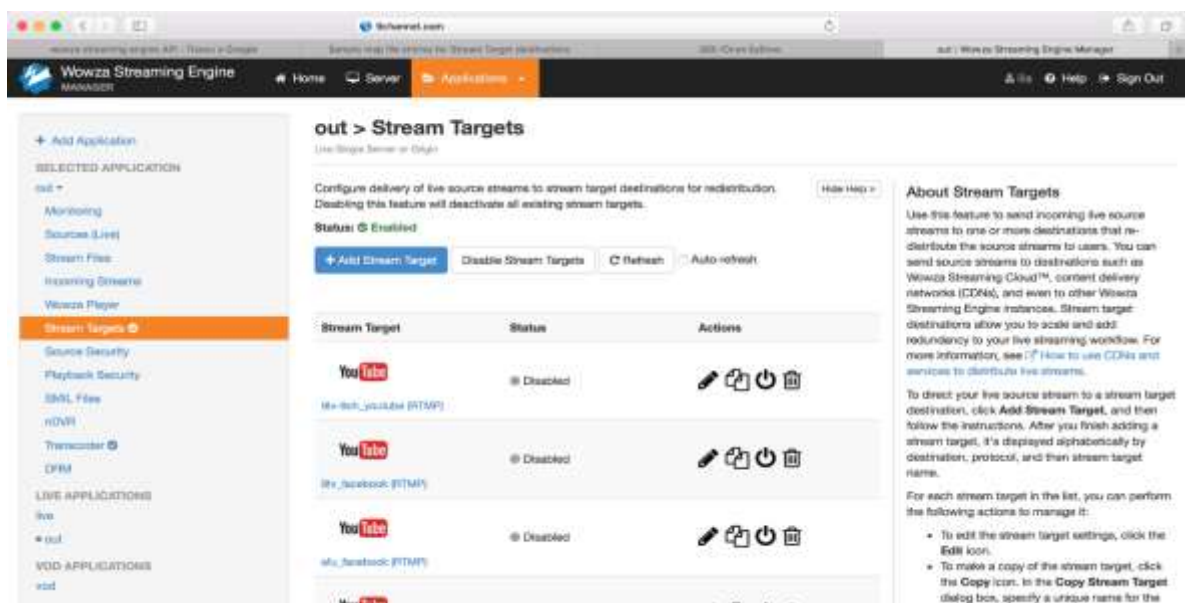


Рисунок 2.17 – Розділ Stream-targets

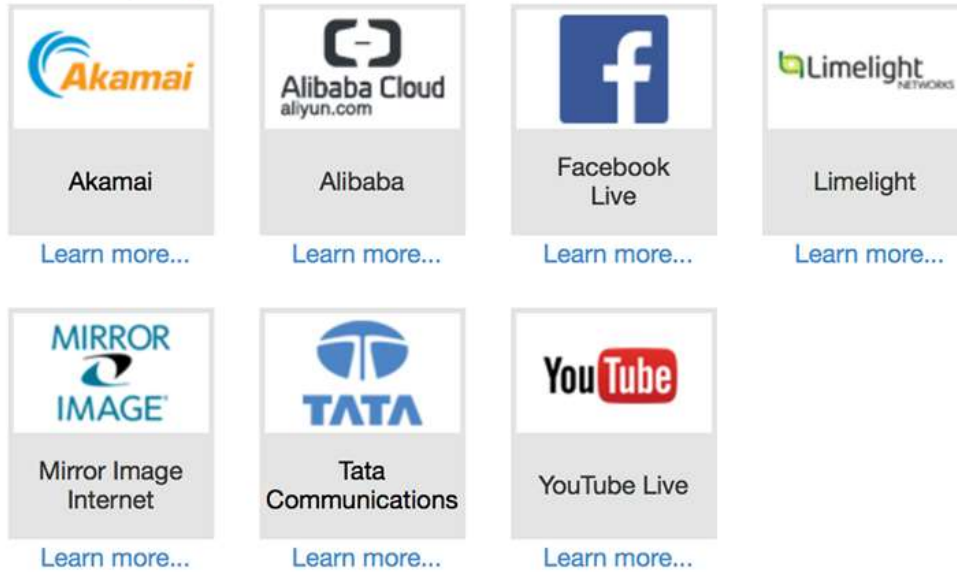
Налаштування цих шаблонів приведені на рис. 2.18 (а і б).

<p>Configure your Stream Target below.</p> <p>Edit</p> <p>Source Stream Name 9tv</p> <p>Destination Application Name live2</p> <p>Destination Application Instance -Not Set-</p> <p>Destination Host a.rtmp.youtube.com</p> <p>Destination Port 1935</p> <p>Destination Stream Name 99w2-</p> <p>User Name -Not Set-</p> <p>Password -Not Set-</p> <p style="text-align: right;">a)</p>	<p>Configure your Stream Target below.</p> <p>Edit</p> <p>Source Stream Name 9tv</p> <p>Destination Application Name rtmp</p> <p>Destination Application Instance -Not Set-</p> <p>Destination Host live-api.facebook.com</p> <p>Destination Port 80</p> <p>Destination Stream Name 1785759614777048?ds=1&a=AThDZTUMN-sfxuRb</p> <p>User Name -Not Set-</p> <p>Password -Not Set-</p> <p style="text-align: right;">б)</p>
---	--

Рисунок 2.18 – Налаштування шаблону переспрямування відеопотоку на:
а – YouTube, б – Facebook

За допомогою цих шаблонів можна вмикати і вимикати переспрямування вхідного потоку, який задається в налаштуваннях на один з доступних сервісів трансляції (рис. 2.19).

Third-Party Target Destinations



Generic Target Destinations

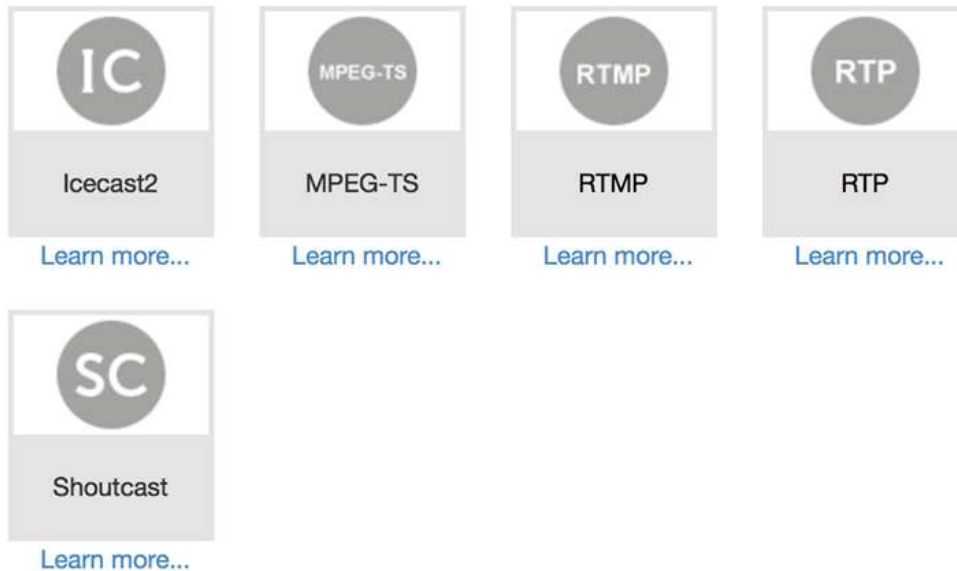


Рисунок 2.19 – Перелік шаблонів за замовчуванням

Деякі шаблони (рис. 2.19) вже мають попереднє налаштування безпосередньо в ПО платформи, інші можна налаштувати вручну. Отже, лишилося написати на базі API скрипт, який би автоматично запускав та зупиняв у визначений період часу заздалегідь підготовлені шаблони. Лістинг цього скрипта:

```
#!/bin/sh
echo date +%Y%m%d-%H:%M:%S' >> /var/log/ilia_script/online.log
```

```

# запис до лог файлу дати та часу виконання скрипта
action=$1
# передача параметру запуску скрипта (допустимі параметри - enable/disable)
/usr/bin/curl --digest -u 'login:password' -X PUT \
--header 'Accept:application/json; charset=utf-8' \
--header 'Content-Type:application/json; charset=utf-8' \
http://localhost:8087/v2/servers/_defaultServer_/vhosts/_defaultVH
ost_/applications/out/publish/mapentries/9tv-
9ch_youtube/actions/${action} \
>> /var/log/ilia_script/online.log 2>/dev/null
# безпосередній запуск API з задалегідь визначеними параметрами
echo '' >> /var/log/scripts/online.log
# запис статусу до лог-файлу

```

Написаний скрипт був доданий до таблиці автоматичного запуску CRONTAB з наступними параметрами:

```

5 18 * * * /home/ilia/9tv_start-stop.sh enable
# запуск скрипта о 18:05
55 18 * * * /home/ilia/9tv_start-stop.sh disable
# зупинка скрипта о 18:05

```

2.4 Організація на базі API додаткового функціоналу (модуля), що дозволяє інтегрувати в систему онлайн-трансляцій компонент pay-per-view, розроблений сторонніми програмістами

Сервіс pay-per-view – це різновид надання послуги перегляду відеоконтенту за передплатою.

Існує декілька типів надання таких послуг. Перший тип передбачає наступне: користувачеві надається декілька секунд (зазвичай до 20 с) перегляду онлайн-трансляції безкоштовно, потім він отримує пропозицію сплатити за подальший перегляд. Також існує підписка на довготривалий термін, коли користувач сплачує за місяць або рік перегляду відеоконтенту постачальника, і

передплата на захід – тип надання послуг, з яким доводиться найчастіше працювати компанії «Visionmark». Незалежно від обраного типу надання послуг, схема організації сервісу pay-per-view на стороні медіаплатформи – майже незмінна (рис. 2.20).



Рисунок 2.20 – Схема організації сервісу pay-per-view

На рис. 2.20 зображені компоненти, задіяні в організації сервісу pay-per-view:

- «Користувач» – користувач із web-браузером;
- «Веб-сервер» – сервер, що обраляє запити від користувачів, опрацьовує білінгову інформацію, має різноманітні форми вводу (пароль користувача, ключ тощо);
- «Wowza» або Wowza Streaming Engine – ПЗ для медіасерверу;
- «БД користувачів» – база даних, що містить інформацію про користувачів та інформацію про наявність дозволу перегляду того чи іншого контенту;

– «Додаток PPV» – екземпляр додатку Wowza Streaming Engine.

Детальніше розглянемо взаємодію цих компонентів.

Користувач відправляє запит на перегляд того чи іншого відео до веб-сервера клієнта компанії, на якому написана система аутентифікації або білінгу. Якщо користувач пройшов авторизацію, тобто його дані співпали з даними, які зберігаються в базі даних, веб-сервер у відповідь формує користувачеві унікальне посилання на відеопотік або код плеєра з цим посиланням. Залежно від того, як налаштована система, також можливе проставляння статусу, що посилання видано, для убезпечення від несанкціонованого перегляду одного і того самого унікального посилання декількома користувачами. Приклад цього посилання:

```
rtmp://origin.visionmark.pl/9tv?userId=123456&starttime=1516286310&key=FsdFew/9tv
```

З цього посилання видно, що до екземпляру додатку у відкритому вигляді передається наступна інформація:

- `userID` – унікальний ID користувача, який не може повторюватися і не може бути пустим;
- `starttime` – дата видачі посилання у форматі Unix Timestamp (використовується здебільшого для змінної при створенні HASH);
- `key` – випадково згенерований набір символів (може використовуватися як додаткова змінна в незакодованому вигляді, так і як змінна для створення HASH).

Наступна частина стосується безпосередньо медіаплатформи. Виходячи з того, що усі змінні, відправлені користувачеві у посиланні, були записані до БД та враховуючи найбільш поширені підходи до зберігання ключів аутентифікації, ці дані можуть бути збережені у БД як у відкритому вигляді, так і у вигляді створеного HASH по заздалегідь визначеній HASH-функції. Коли користувач запитує посилання у медіасервера, останній перенаправляє його до екземпляра додатку, зазначеного в посиланні. І вже залежно від налаштувань та

використання додатково написаного ПЗ до цього екземпляру, виконуються наступні дії.

Отримані дані (у відкритому вигляді або у вигляді HASH) порівнюються з даними, записаними для цього користувача у БД, і в разі співпадіння – користувачеві надається можливість перегляду відеопотоку. Якщо процес порівняння закінчився невдачею – користувачеві надсилається відповідь про помилкові дані або несанкціонований доступ або він переспрямовується на перегляд іншого, зокрема рекламного контенту.

Нижче приведена реалізація частини цієї системи на стороні медіасервера в найпростішому вигляді. Спочатку для цього на сервер було встановлене необхідне ПЗ: MariaDB Server і MariaDB. Нижче приведений лістинг командної строки, за допомогою якої була створена БД:

```
MariaDB [(none)]> create database wowza;
Query OK, 1 row affected (0.00 sec)
MariaDB [(none)]> use wowza;
Database changed
MariaDB [wowza]> create table ppv
( id BIGINT(20) UNSIGNED NOT NULL AUTO_INCREMENT,
  secret_key VARCHAR(20),
  allow BOOL NOT NULL DEFAULT '0',
  PRIMARY KEY (id) );
Query OK, 0 rows affected (0.02 sec)
```

Для прикладу була створена таблиця з трьома стовпчиками, де ID буде головним унікальним ідентифікатором, що заповнюється автоматично, key – довільне символічно-числове значення, довжиною до 20 символів; allow – параметр типу BOOL, який буде означати дозволений клієнту перегляд потоку чи ні.

Лістинг стовпчиків наведений нижче:

```
MariaDB [wowza]> show tables;DESCRIBE ppv;
+-----+
| Tables_in_wowza |
```

```

+-----+
| ppv          |
+-----+
1 row in set (0.00 sec)

+-----+-----+-----+-----+-----+-----+
| Field        | Type                | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id           | bigint(20) unsigned | NO   | PRI | NULL    | auto_increment |
| secret_key  | varchar(20)         | YES  |     | NULL    |                |
| allow       | tinyint(1)         | NO   |     | 0       |                |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)

```

Простота таблиці та параметрів обумовлена тим, що лістинг, приведений вище, та інформація, яка буде надана далі є лише наглядним прикладом, створеним на основі працюючого ПЗ в компанії «Visiomark» та наданим для відображення проведеної роботи, оскільки основний код захищений авторським правом. Отже, дані до таблиці будуть додані в ручному режимі у кількості п'яти рядків:

```

MariaDB [wowza]> insert into ppv ( secret_key, allow ) VALUES (
'123456', 1 );

```

```

Query OK, 1 row affected (0.00 sec)

```

```

MariaDB [wowza]> insert into ppv ( secret_key, allow ) VALUES (
'123457', 1 );

```

```

Query OK, 1 row affected (0.00 sec)

```

```

MariaDB [wowza]> insert into ppv ( secret_key, allow ) VALUES (
'123458', 1 );

```

```

Query OK, 1 row affected (0.01 sec)

```

```

MariaDB [wowza]> insert into ppv ( secret_key, allow ) VALUES (
'123459', 1 );

```

```

Query OK, 1 row affected (0.00 sec)

```

```
MariaDB [wowza]> insert into ppv ( secret_key, allow ) VALUES (
'123410', 1 );
Query OK, 1 row affected (0.00 sec)
```

Кінцевий вигляд таблиці:

```
MariaDB [wowza]> select * from ppv;
+----+-----+-----+
| id | secret_key | allow |
+----+-----+-----+
| 1  | 123456     | 1     |
| 2  | 123457     | 1     |
| 3  | 123458     | 1     |
| 4  | 123459     | 1     |
| 5  | 123410     | 1     |
+----+-----+-----+
5 rows in set (0.00 sec)
```

Далі за документацією Wowza Streaming Engine Java API [1] був створений модуль авторизації користувача, відповідно до параметрів, переданих до нашого екземпляру додатку. Оскільки модуль написаний на Java до основного ПЗ Wowza Streaming Engine був підключений драйвер MySQL Connector/J. Це офіційний драйвер для взаємодії Java та MySQL [2].

Далі на комп'ютері розробки було встановлене ПЗ Eclipse IDE і за рекомендаціями Wowza [3] налаштоване середовище розробки та створений типовий проект модуля (рис. 2.21).

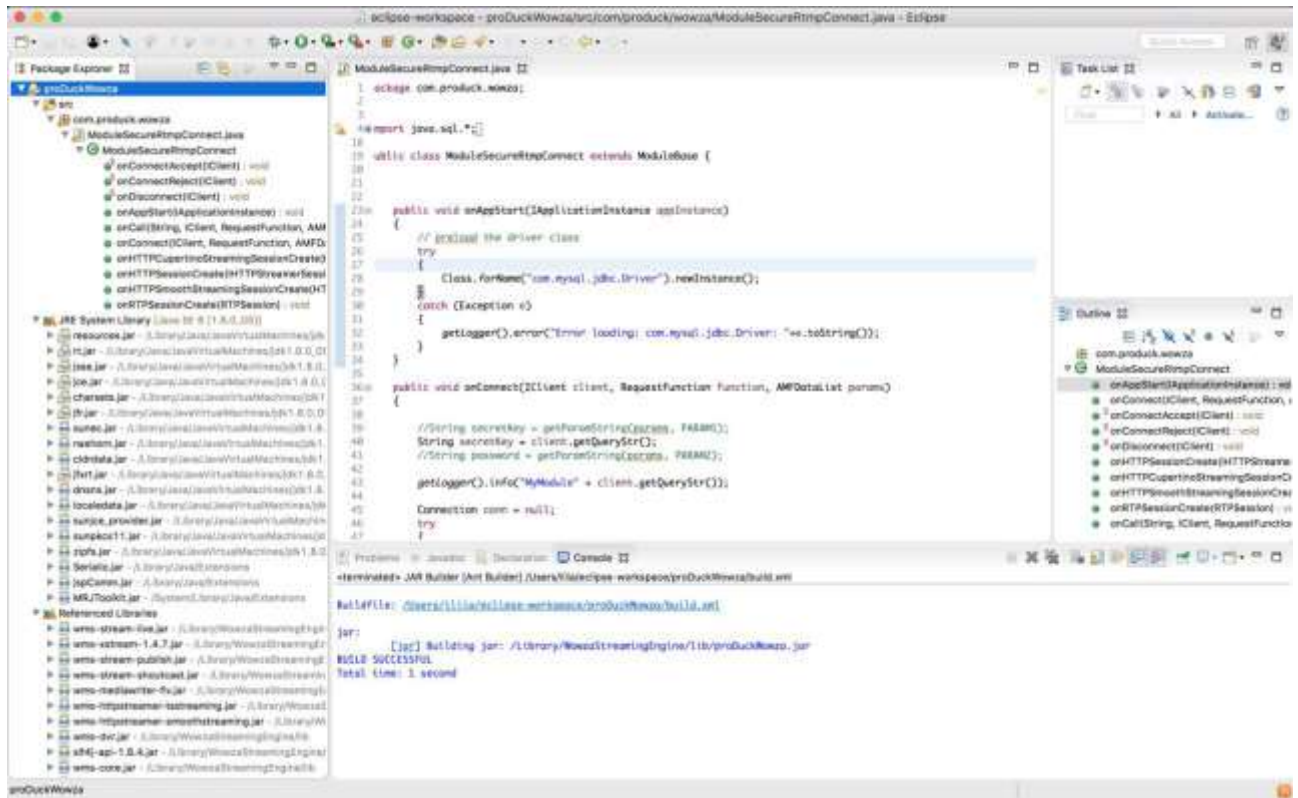


Рисунок 2.21– Середовище розробки для створення модуля

Детальний лістинг модуля наведений у Додатку Б.

Працездатність модуля перевірена за допомогою медіаплеєра VLC та інструменту перегляду лог-файлу, влаштованого до ПЗ платформи.

Спочатку вводимо до діалогового вікна VLC-медіаплеєру адресу з вірним даними, а саме з вірним посиланням на потік: `rtmp://origin.visionmark.com/9tv?123456/9tv` (рис. 2.22).

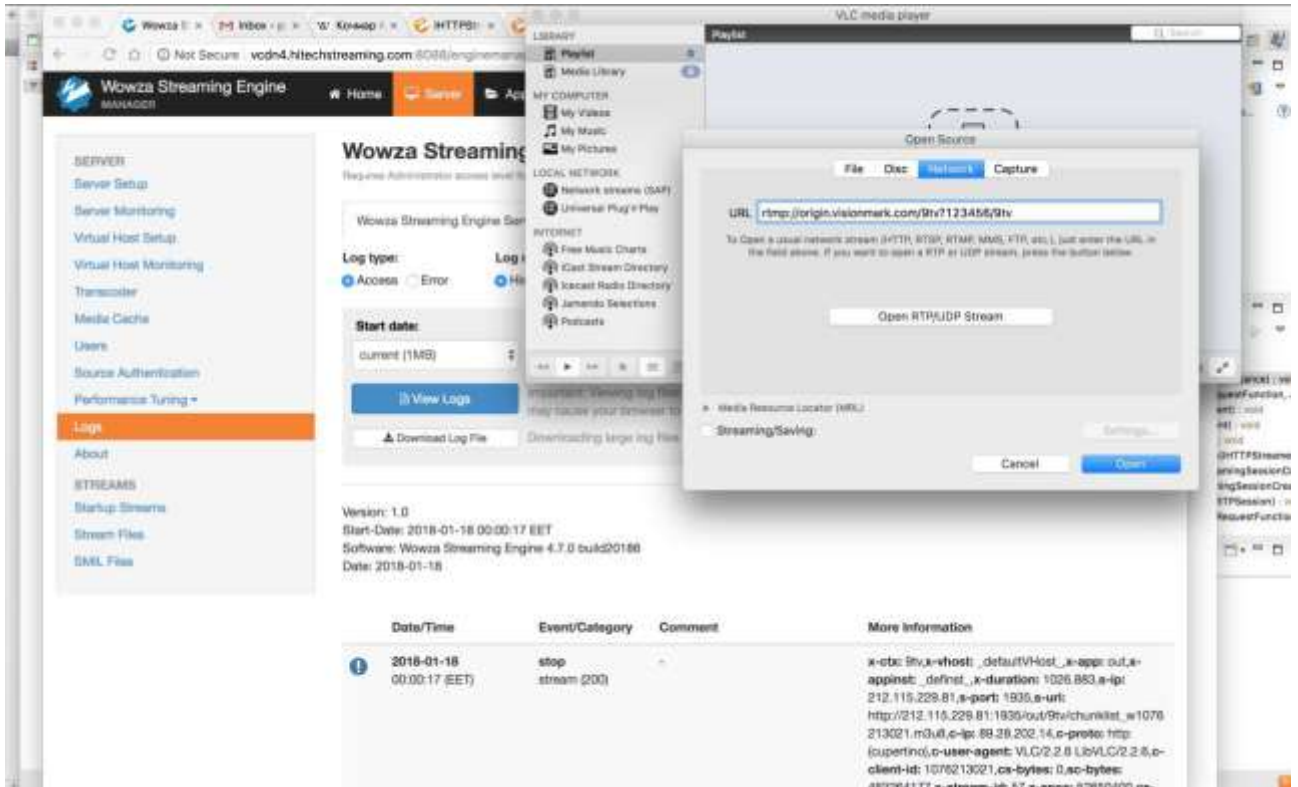


Рисунок 2.22 – Тестування працездатності модуля заздалегідь вірним посиланням на потік

Натиснувши «play», було оглянуто записи у лог-файлі (рис. 2.23).

Date/Time	Event/Category	Comment	More Information
2018-01-18 18:13:55 (EET)	connect session (200)	-	x-ctx: 89.28.202.14,x-vhost: _defaultVHost_,x-app: 9tv,x-appinst: _definst_,x-duration: 0.03,s-ip: [any],s-port: 1935,s-uri: rtmp://vcdn4.hitechstreaming.com:1935/9tv?123456,c-ip: 89.28.202.14,c-proto: rtmp,c-user-agent: LNX 9,0,124,2,c-client-id: 934210152,cs-bytes: 3310,sc-bytes: 3073,cs-uri-stem: rtmp://vcdn4.hitechstreaming.com:1935/9tv,cs-uri-query: 123456
2018-01-18 18:13:55 (EET)	comment server (200)	onConnect MyModule: 123456-934210152	x-vhost: _defaultVHost_,x-app: 9tv,x-appinst: _definst_,x-duration: 53368.321
2018-01-18 18:13:55 (EET)	comment server (200)	MyModule: 123456 connect client: 934210152	x-vhost: _defaultVHost_,x-app: 9tv,x-appinst: _definst_,x-duration: 53368.321
2018-01-18 18:13:55 (EET)	comment server (200)	MyModule123456	x-vhost: _defaultVHost_,x-app: 9tv,x-appinst: _definst_,x-duration: 53368.313

Рисунок 2.23 – Записи в лог-файлі при тестуванні працездатності модуля заздалегідь вірним посиланням на потік

На рис. 2.23 видно, що модуль відпрацював зі статусом (200), що свідчить про успішне підключення.

Наступним кроком вводимо завідомо хибне посилання (рис. 2.24), а саме: `rtmp://origin.visionmark.com/9tv?145456/9tv`, де ключ має хибне значення 145456, яке не закріпле за жодним із користувачів.

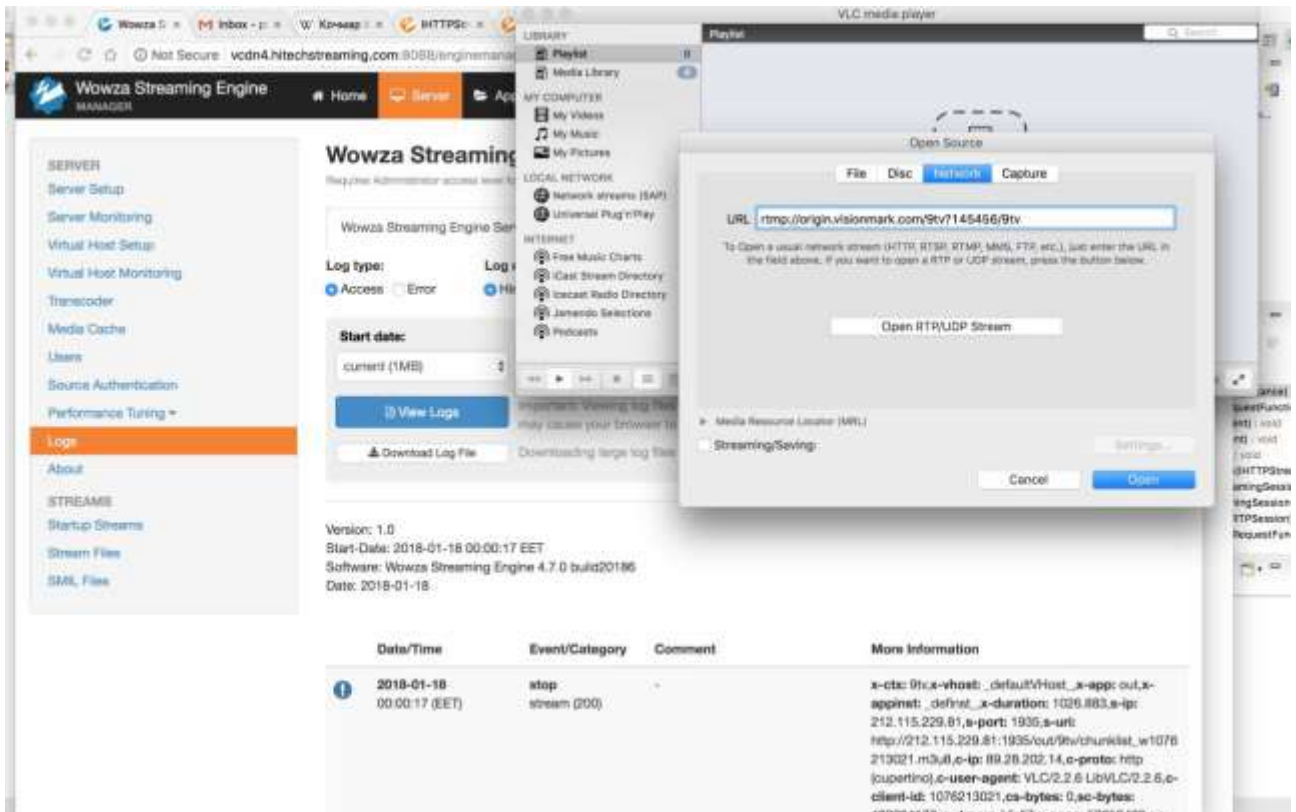


Рисунок 2.24 – Тестування працездатності модуля заздалегідь хибним посиланням на потік

Як видно з лог-файлу (рис. 2.25), після спроби підключення клієнту повернувся статус (401), що означає «не авторизоване з'єднання», після чого відбулось відхилення з'єднання – подія «onConnectReject».

2018-01-18 18:20:50 (EET)	comment server (200)	onConnectReject my Module: 1179682316	x-vhost: _defaultVHost_x-app: 9tv,x-appinst: _definst_x-duration: 53783.201
2018-01-18 18:20:50 (EET)	connect session (401)	-	x-ctx: 89.28.202.14,x-vhost: _defaultVHost_x-app: 9tv,x-appinst: _definst_x-duration: 0.025,s-ip: [any],s-port: 1935,s-uri: rtmp://vcdn4.hitechstreaming.com:1935/9tv? 145456,c-ip: 89.28.202.14,c-proto: rtmp,c-user- agent: LNX 9,0,124,2,c-client-id: 1179682316,cs- bytes: 3310,sc-bytes: 3073,cs-uri-stem: rtmp://vcdn4.hitechstreaming.com:1935/9tv,cs-uri- query: 145456
2018-01-18 18:20:50 (EET)	comment server (200)	onConnect MyModule: 145456- 1179682316	x-vhost: _defaultVHost_x-app: 9tv,x-appinst: _definst_x-duration: 53783.2
2018-01-18 18:20:50 (EET)	comment server (200)	MyModule145456	x-vhost: _defaultVHost_x-app: 9tv,x-appinst: _definst_x-duration: 53783.191

Рисунок 2.25 – Записи в лог-файлі при тестуванні працездатності модуля задалегідь хибним посиланням на потік

З приведенного лістингу та скриншотів робимо висновок, що модуль відпрацював у заданому режимі. Звичайно ж, робочий модуль, який використовується на Enterprise-платформі компанії «Visionmark», має набагато складніший алгоритм, але основні вузли були відображені у дипломній роботі.

2.5 Організація онлайн-трансляцій з круговим оглядом у 360 градусів

Відеомовлення з камер в режимі сферичного відео набирає у світі все більшої популярності. Основними складовими (рис. 2.26) такої системи є:

- камера 360 градусів чи набір камер, підключених одна до одної, які разом дають огляд у 360 градусів;
- програмне чи апаратне забезпечення для захоплення і передачі потоку на медіасервер;
- медіасервер Wowza Streaming Engine або однойменний Cloud-сервіс;
- програмне забезпечення або пристрій для перегляду.

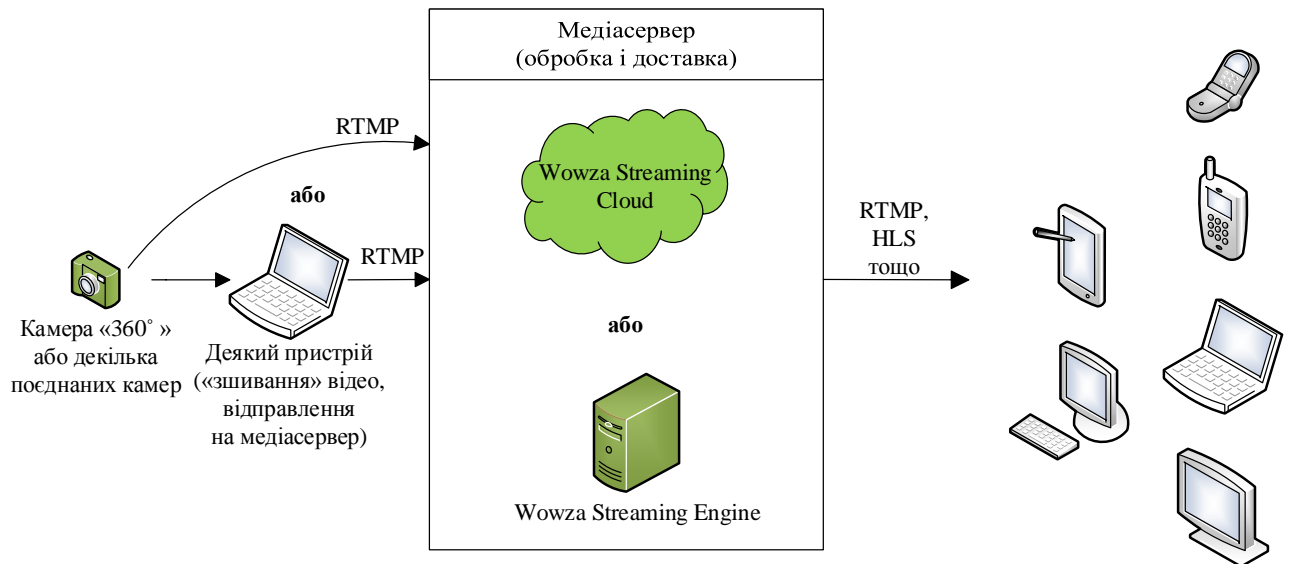


Рисунок 2.26 – Структурна схема передачі відеопотоку при організації онлайн-трансляцій з круговим оглядом у 360 градусів

У нашому випадку відеомовлення з круговим оглядом у 360 градусів підтримується налаштуванням додатку до Wowza Streaming Engine, що інтегрований у платформу за замовчуванням, і від основної схеми онлайн-трансляцій відрізняється лише пристроєм, що надсилає потік (додається відеокамера з круговим оглядом у 360 градусів та підтримкою передачі RTMP-потіку), і пристроєм, що його відтворює (пристрій користувача). А зі сторони медіаплатформи необхідна підтримка передачі потоку великої роздільної здатності (4k і більше). Wowza Streaming Engine відповідає цій вимозі.

2.5.1 Обрання відеокамери

Відеокамера може складатися з кількох окремих відеокамер, наприклад, декількох GoPro, поєднаних між собою за допомогою спеціального утримувача, або бути самостійним пристроєм (Giroptic iO camera, 360fly, Orah 4i). Деякі відеокамери «вміють» самостійно передавати контент на медіасервер або на сервіс відеомовлення, наприклад на YouTube за протоколом RTMP в форматі H.264. Одну з таких було обрано компанією «Visionmark» для організації трансляцій з круговим оглядом у 360 градусів. Відеокамера Yi 360 VR

відповідає технічним вимогам компанії при достатньо низькій вартості, що забезпечує гарне співвідношення ціна/якість.

Деякі камери можна підключати до мобільного пристрою і вже за рахунок встановленого на мобільному пристрої додатку потік буде відправлений на медіасервер.

Ще один варіант – підключення декількох камер через HDMI до спеціального апаратного пристрою (наприклад, Teradek Sphere HDMI 360 LS), який вже направляє потік на медіасервер. Teradek Sphere можна одночасно підключити до чотирьох камер. Можливе поєднання кількох Teradek Sphere між собою. Це достатньо якісне рішення, однак його ціна (близько трьох тисяч доларів) перевищує можливості багатьох українських телекомунікаційних компаній.

2.5.2 Обрання програмного забезпечення для відеомовлення з круговим оглядом у 360 градусів.

У нашому випадку додаток, що забезпечує відеомовлення з круговим оглядом у 360 градусів, інтегрований за замовчуванням до ПЗ Wowza Streaming Engine, а обрана відеокамера має підтримку передачі RTMP-потіку, а отже, має змогу напряму підключатися до медіасерверу. Однак для інших випадків існує ПЗ, яке можна встановити на комп'ютер і використовувати для захоплення відео з кількох камер, формування і відправки потоку на медіасервер. Одне з найпопулярніших ПЗ – Vahana VR.

Воно підтримує як HDMI, так і SDI камери, що можуть підключатися до комп'ютера через завдяки різним картам захоплення, наприклад, Blackmagic Decklink, Magewell, Aja. Більше того на вхід даної програми можуть надходити потоки за протоколом RTP/RTSP.

Вихідні потоки Vahana VR можуть бути такими:

- протокол RTMP;
- стримінг на YouTube;
- SDI (за допомогою спеціальних карт);

- запис на HDD;
- шлеми віртуальної реальності: Oculus Rift, HTC Vive.

2.5.3 Рекомендації щодо обрання засобів перегляду сферичного відео

Перший і найбільш розповсюджений спосіб – перегляд на відеоплеєрі, наприклад, на Bitdash. У цьому випадку глядач має можливість мишкою розвертати відео і дивитись усе, що є навколо. Якщо перегляд здійснюється на смартфоні, зміна ракурсу можлива у два способи: завдяки тачскріну або нахилу пристрою. Ще один засіб перегляду — окуляри або шлем віртуальної реальності (Samsung Gear VR, Sony PlayStation VR, Oculus Rift, HTC Vive), коли зміна ракурсу залежить від повороту голови глядача. Вартість таких пристроїв – від 60 доларів. Пересічний мешканець Дніпропетровської області таких ресурсів не має. Тож перегляд зазвичай здійснюється завдяки портативним пристроям – планшету, ПК або смартфону. Відтак замість окулярів або шлему віртуальної реальності можна використовувати дешеві паперові окуляри віртуальної реальності Google Cardboard сумісно зі смартфоном.

2.6 Висновки до другого розділу

Підсумовуючи роботу, проведену в другому розділі, можна зробити такі висновки:

1 Серед запропонованих на ринку ПЗ рішень сформульованим у першому розділі технічним вимогам відповідають одразу дві медіасерверні платформи: Nimble Streamer і Wowza Streaming Engine. Оскільки на вибір медіа-платформи вагомо впливає затримка проходження відеопотоку через медіасервер, було прийняте рішення визначити затримку на цих двох платформах емпіричним шляхом. Затримка трансмуksингу Nimble Streamer склала 1,756с, а затримка трансмуksингу Wowza Streaming Engine – 1,475с. Останню платформу було обрано для модернізації онлайн-системи. У якості додаткових серверних потужностей був обраний сервіс Wowza Streaming Cloud, який дозволив зменшити кількість апаратних серверів з чотирьох до одного.

2 ПЗ Wowza Streaming Engine було успішно налаштоване і протестоване. Зокрема, мінімізувати затримку до 1,5с дозволило «тонке» налаштування. При цьому частина користувачів (або всі користувачі при нормальному навантаженні) прийматиме відеопотік в режимі реального часу з мінімальною затримкою (1,5с) без втрати якості, а решта - із запізненням, але зі збереженням HD-якості.

3 На базі API було написане власне програмне доповнення, що автоматизує процес переспрямування відеопотоку в режимі реального часу на популярні платформи YouTube і Facebook. Крім того, на базі API був створений додатковий функціонал (модуль) для інтеграції в систему онлайн-трансляцій компоненту pay-per-view. Модуль був успішно протестований введенням правильного та хибного посилань.

4 Були наведені рекомендації для організації онлайн-трансляцій з круговим оглядом у 360 градусів, при чому було обране бюджетне рішення придбання відеокамери з достатньо високим співвідношенням ціна/якість.

РОЗДІЛ 3. ЕКОНОМІЧНИЙ РОЗДІЛ

Приведений у другому розділі проект модернізації телекомунікаційної системи онлайн-трансляцій передбачає повну заміну програмного і часткову заміну апаратного забезпечення медіасерверу компанії «Visionmark». Крім того, в рамках інноваційного удосконалення системи передбачена закупівля додаткового обладнання та ПЗ для організації відеомовлення з кругозором у 360 градусів. Основними задачами поточного розділу є визначення капітальних витрат проекту модернізації, а також економічного ефекту, що виникає внаслідок скорочення експлуатаційних витрат.

3.1 Розрахунок капітальних витрат на модернізацію системи онлайн-трансляцій

Капітальні інвестиції – це кошти, призначені для створення і придбання основних фондів і нематеріальних активів, що підлягають амортизації.

Капітальні (фіксовані) витрати (K) на проектування та впровадження проектного рішення модернізованої телекомунікаційної системи онлайн-трансляцій складають:

$$K = K_{пр} + K_{зпз} + K_{пз} + K_{аз} + K_{навч} + K_{н} + K_{с}, \quad (3.1)$$

де $K_{пр}$ – вартість розробки проекту та залучення до цього зовнішніх консультантів, грн.;

$K_{зпз}$ – вартість закупівель ліцензійного основного й додаткового програмного забезпечення (ПЗ), грн.;

$K_{пз}$ – вартість створення основного й додаткового програмного забезпечення, грн.;

$K_{аз}$ – вартість закупівель апаратного забезпечення та допоміжних матеріалів, грн.;

$K_{навч}$ – витрати на навчання технічних фахівців і обслуговуючого персоналу, грн.;

K_n – витрати на встановлення обладнання та налагодження модернізованої системи онлайн-трансляцій, грн.;

K_c – витрати на послуги фахівця зі встановлення обладнання, грн..

На практиці в компанії «Visionmark» послуги, для яких розраховують $K_{пз}$, $K_{навч}$, K_n і K_c , за замовчуванням входять до розробки проекту. І їхня спільна вартість складає: $K_{пр} = 28\,380$ грн.

Проектні капіталовкладення в апаратне забезпечення ($K_{аз}$) складаються лише з витрат на придбання відеокамери з круговим оглядом у 360 градусів. Адже нова програмна платформа встановлена на вже наявне серверне обладнання, а додаткові серверні потужності Wowza Streaming Cloud не можна брати до уваги при розрахунку капітальних витрат через те, що вони надаються компанії «Visionmark» в оренду, тобто відносяться до експлуатаційних витрат.

Усі витрати на придбання апаратного забезпечення ($K_{аз}$), ліцензійного основного й додаткового ПЗ ($K_{пз}$) визначені на основі цін, наведених у прайс-листах відповідних фірм, інших довідкових матеріалів або за фактичними витратами. Загальна вартість обраного обладнання і ПЗ наведена у табл. 3.1.

Таблиця 3.1 – Загальна вартість обладнання та ПЗ, обраного для реалізації проекту модернізації телекомунікаційної системи онлайн-трансляцій у компанії «Visionmark»

Назва обладнання чи ПЗ	Кількість, шт.	Ціна, грн.
Відеокамера «360 градусів» з можливістю передачі RTMP-потoku – Yi 360 VR	1	11 352
Безстрокове ліцензійне ПЗ Wowza Streaming Engine	1	56 618

Отже, кінцеву формулу капітальних витрат (K) на проектування та впровадження модернізованої телекомунікаційної системи онлайн-трансляцій можна записати так:

$$K = K_{\text{пр}} + K_{\text{зпз}} + K_{\text{аз}} \quad (3.2)$$

Тоді:

$$K = 28\,380 + 56\,618 + 11\,352 = 96\,350 \text{ (грн.)}$$

3.2 Визначення економічного ефекту, що виникає за рахунок зменшення експлуатаційних витрат

Експлуатаційні витрати – це поточні витрати на експлуатацію та обслуговування об'єкта проектування за визначений період (наприклад, рік), що виражені в грошовій формі.

Річні експлуатаційні витрати по об'єкту проектування складають:

$$Z = C_a + C_z + C_c + C_m + C_e + C_i, \quad (3.3)$$

де C_a – амортизаційні відрахування, грн.;

C_z – заробітна плата обслуговуючого персоналу, грн.;

C_c – відрахування на соціальні заходи від заробітної плати, грн.;

C_m – витрати на технічне обслуговування і поточний ремонт устаткування та мереж, грн.;

C_e – вартість електроенергії, що споживається об'єктом проектування, грн.;

C_i – інші витрати, грн..

Головне завдання даного підрозділу – розрахувати різницю (ΔZ) між річними експлуатаційними витратами модернізованого медіасерверу (що працює на базі платформи Wowza Streaming Engine) і попереднього

медіасерверу (що працював на базі NGINX RTMP module). Для такого порівняння припустимо, що платформа NGINX RTMP module продовжила функціонувати у 2017 році. І розрахуємо річні експлуатаційні витрати для цього випадку паралельно із розрахунком експлуатаційних витрат модернізованої системи.

Внаслідок модернізації системи онлайн-трансляцій не зазнали змін наступні статті експлуатаційних витрат:

- C_z – заробітна плата обслуговуючого персоналу, грн.;
- C_c – відрахування на соціальні заходи від заробітної плати, грн.;
- C_m – витрати на технічне обслуговування і поточний ремонт устаткування та мереж, грн..

Отже, їх можна не брати до уваги при розрахунку ΔZ .

Тим часом амортизаційні відрахування у цьому випадку доцільно визначити за методом зменшення залишкової вартості. Суть даного методу полягає у визначенні річної суми амортизації об'єкта основних засобів виходячи із залишкової вартості такого об'єкта на початок звітного року. Річна сума амортизаційних відрахувань визначається шляхом множення залишкової вартості об'єкта основних засобів на початок звітного року або первинної вартості на дату початку нарахування амортизації на норму річної амортизації.

Норму річної амортизації при цьому визначають за наступною формулою:

$$H_a = 1 - \left(\frac{C_L}{C_n}\right)^{\frac{1}{n}}, \quad (3.4)$$

де H_a – річна норма амортизації, долі (або %);

n – термін корисного використання об'єкта основних засобів, роки;

C_L – ліквідаційна вартість об'єкта основних засобів, грн.;

C_n – первісна вартість об'єкта основних засобів, грн.

Місячна сума амортизації визначається діленням суми амортизації за повний рік корисного використання на 12 місяців.

Використання метода зменшення залишкової вартості при нарахуванні амортизації припускає наявність ліквідаційної вартості об'єкта основних засобів, що амортизується. Інакше об'єкт основних засобів буде повністю «амортизовано» протягом одного року.

Слід зазначити, що базою для нарахування амортизації при використанні даного метода є залишкова вартість на початок звітного року або первинна вартість на дату початку нарахування амортизації.

З цього виходить, що починаючи з дати введення в експлуатацію об'єкта основних засобів до кінця звітного року, в якому був введений такий об'єкт, базою для нарахування амортизації буде первинна вартість такого об'єкта. Іншими словами, якщо об'єкт основних засобів був введений в експлуатацію в липні, то з серпня по грудень (включно) річна сума амортизаційних відрахувань визначатиметься шляхом множення первинної вартості об'єкта на норму амортизації. З січня наступного року річна сума амортизації такого об'єкта основних засобів визначатиметься шляхом множення залишкової вартості такого об'єкта на норму амортизації.

Даний метод нарахування амортизації заснований на припущенні, що амортизація починає нараховуватися з початку звітного року, хоча практично це буває далеко не завжди. Тому у всіх випадках, коли амортизація нараховується не з початку звітного року, залишкова вартість в кінці запланованого терміну експлуатації не досягає ліквідаційної вартості. У зв'язку з цим підприємство може ухвалити рішення про зміну терміну амортизації і доамортизувати частину вартості, що залишилася, або ж вважати залишкову вартість об'єкта ліквідаційною вартістю.

У випадку модернізованої системи онлайн-трансляцій придбаними об'єктами основних засобів є програмне забезпечення для медіасерверу Wowza Streaming Engine, амортизація на яке не розраховується, і камера на 360 градусів, вартістю 11 352 грн.. Термін корисної дії камери – 5 років. Також до

розрахунків щорічної амортизації слід додати 1 сервер HP ProLiant DL380 компанії «Visionmark», вартість якого на початок 2013 року становила 90 тис. грн., а термін корисного використання складав 10 років. Очікується, що сервер HP ProLiant DL380 матиме ліквідаційну вартість, рівну 5 000 грн., а камера Yi 360 VR – рівну 500 грн.

Отже, амортизаційні відрахування модернізованої системи необхідно розрахувати так:

$$C_a = C_{a_K} + C_{a_C}, \quad (3.5)$$

де C_{a_K} – амортизаційні відрахування камери Yi 360 VR у 2017 році, грн.;
 C_{a_C} – амортизаційні відрахування серверу HP ProLiant DL380 у 2017 році, грн..

Річна норма амортизації серверу за звичайною методикою розрахунку складе:

$$H_{a_C} = 1 - \left(\frac{5\,000}{90\,000} \right)^{\frac{1}{10}} = 0,251 = 25,1(\%)$$

А річна норма амортизації камери становить:

$$H_{a_K} = 1 - \left(\frac{500}{11\,352} \right)^{\frac{1}{5}} = 0,464 = 46,4(\%)$$

Оскільки камера була введена в експлуатацію в травні, амортизація нараховуватиметься з червня. При цьому розрахована норма амортизації застосовуватиметься до первинної вартості об'єкта. Сума амортизаційних відрахувань за перші 7 місяців експлуатації складе:

$$C_{a_K} = \frac{11\,352 \text{ грн.} \cdot 0,464 \cdot 7 \text{ міс.}}{12 \text{ міс.}} = 3\,072,61 \text{ грн.}$$

Розрахунок амортизаційних відрахувань для двох об'єктів основних засобів в розрізі років представлений в табл. 3.2 і табл. 3.3.

Таблиця 3.2 – Розрахунок амортизації об'єкта основних засобів - сервера HP ProLiant DL380

Рік	Балансова (залишкова) вартість, грн.	Норма амортизації, долі	Річна сума амортизаційних відрахувань, грн.
2013	90 000	0,251	22 590
2014	67 410	0,251	16 919,91
2015	50 490,09	0,251	12 673,01
2016	37 817,08	0,251	9 492,09
2017	28 324,99	0,251	7 109,57
2018	21 215,42	0,251	5 325,07
2019	15 890,35	0,251	3 988,48
2020	11 901,87	0,251	2 987,37
2021	8 914,5	0,251	2 237,54
2022	6 676,96	0,251	1 675,92

Таблиця 3.3 – Розрахунок амортизації об'єкта основних засобів - камера «360 градусів» з можливістю передачі RTMP-потоків Yі 360 VR

Рік	Балансова (залишкова) вартість, грн.	Норма амортизації, долі	Річна сума амортизаційних відрахувань, грн.
2017 (7 місяців)	11 352	0,464	3 072, 61
2018	8 279,39	0,464	3 841,64
2019	4 437,75	0,464	2 059,12
2020	2 378,63	0,464	1 103,68
2021	1 274,95	0,464	591,58
2022 (5 місяців)	683,37	0,464	132,12

Для розрахунку різниці (ΔZ) між річними експлуатаційними витратами модернізованої платформи на базі Wowza Streaming Engine і попередньої платформи на базі NGINX RTMP module необхідно навести розрахунок амортизаційних відрахувань та щомісячних витрат для старої платформи. Для цього вважатимемо, що платформа лишилася працювати у 2017 році на базі чотирьох серверів HP ProLiant DL380. Тоді на початок року залишкова вартість кожного сервера складе 28324,99 грн. (табл. 3.2), а їхня загальна річна сума амортизаційних відрахувань становитиме:

$$C_{a_{4C}} = 7109,57 \text{ грн.} \cdot 4 \text{ сервери} = 28\,438,28 \text{ грн.}$$

Вартість електроенергії, що споживається об'єктом проектування протягом року, визначається за формулою:

$$C_e = W_r \cdot C_e, \quad (3.6)$$

де W_r – кількість спожитої за рік електроенергії, кВт·годин;

C_e – тариф на електроенергію, грн./кВт·годин.

Однак витрати на електроенергію, що споживають сервери, бере на себе Інтернет-провайдер, на місці зосередження трафіку якого компанія “Visionmark” орендує стойки під сервери. При цьому витрати електроенергії не мають перевищувати 350 кВт·годин. Тоді їхня вартість за замовчуванням входить до сталої вартості щомісячної оренди. Таким чином, витрати на електроенергію можна не брати до розрахунку, а витрати на оренду каналів і стоек під сервери необхідно включити до інших експлуатаційних витрат.

Отже, до інших експлуатаційних витрат (C_i) можна віднести щомісячні витрати за статтями, вказаними в табл. 3.4:

Таблиця 3.4 – Статті інших експлуатаційних витрат

Витрати Системи	Оренда каналів і стоек під сервери на місці зосередження трафіку провайдера ТОВ “Фрегат”	Оренда CDN Wowza Streaming Cloud на основі середньомісячного трафіку (у 2017 році)
Модернізована	Оренда однієї стойки 2U з одним каналом, пропускною здатністю 1Гбіт/с. $(C_{i_1} = 2\,020 \text{ грн./місяць})$	$C_{i_2} = 2\,602 \text{ грн./місяць}$
Немодернізована	Оренда чотирьох стоек по 2U з чотирма каналами, пропускною здатністю 2 Гбіти/с. $(C_{i_3} = 13\,080 \text{ грн./місяць})$	—

Отже, інші річні експлуатаційні витрати модернізованої системи треба визначати як суму щомісячних витрат, помножену на 12 місяців:

$$C_{i_M} = 12 \cdot (C_{i_1} + C_{i_2}) = 12 \cdot (2\,020 + 2\,602) = 55\,464 \text{ (грн.)}$$

Тим часом інші річні експлуатаційні витрати немодернізованої системи дорівнюють:

$$C_{i_H} = 12 \cdot C_{i_3} = 156\,960 \text{ (грн.)}$$

На цьому етапі можна розрахувати річні експлуатаційні витрати модернізованого медіасерверу (що працює на базі платформи Wowza Streaming Engine) і попереднього (немодернізованого) медіасерверу (що працював на базі NGINX RTMP module):

$$Z_M = 7\,109,57 + 3\,072,61 + 55\,464 = 65\,646,18 \text{ (грн./рік)}$$

$$Z_H = 28\,438,28 + 156\,960 = 185\,398,28 \text{ (грн./рік)}$$

Звідси:

$$\Delta Z = Z_H - Z_M = 185\,398,28 - 65\,646,18 = 119\,752,1 \text{ (грн./рік)}$$

3.3 Висновки до третього розділу

В економічному розділі були проведені розрахунки:

– капітальних витрат проекту модернізації. До розрахунку включено: вартість розробки проекту та залучення до цього зовнішніх консультантів, вартість закупівель ліцензійного основного й додаткового програмного забезпечення і вартість закупівель апаратного забезпечення та допоміжних матеріалів. Капітальні витрати на модернізацію системи онлайн-трансляцій склали 96 350 грн.;

– економічного ефекту, що виникає внаслідок скорочення експлуатаційних витрат. До розрахунку експлуатаційних витрат включено: амортизаційні відрахування апаратного забезпечення, оренда каналів і стійок під сервери на місці зосередження трафіку провайдера ТОВ «Фрегат» і оренда CDN Wowza Streaming Cloud на основі середньомісячного трафіку (у 2017 році). Під час розрахунку було визначено, що експлуатаційні витрати скоротилися майже втричі, а економія склала 119 752,1 грн./рік. Причина - зменшення кількості працюючих серверів HP ProLiant DL380 з чотирьох до одного внаслідок модернізації системи.

Підсумуємо: економічний ефект, що виник внаслідок скорочення експлуатаційних витрат, дозволить окупити всі капітальні витрати на модернізацію системи вже протягом першого року її роботи.

ВИСНОВКИ

У даній дипломній роботі були отримані наступні результати:

1 Визначено основні недоліки програмної платформи NGINX RTMP module, на базі якої працював медіасервер компанії «Visionmark» у 2012-2016 роках. Серед них: наявність великої затримки передачі відеопотоку (більше 5с), не якісна підтримка HLS-протоколу, неможливість переспрямування потоків на популярний сервіс YouTube, неможливість підтримки трансляції з мобільних пристроїв, використання потужностей одразу чотирьох апаратних серверів і, за рахунок не дуже якісної системи транскодування та трансрейту, неефективне використання процесорної потужності.

2 Визначені етапи передачі відеопотоку, які вносять найбільшу затримку, і етап, під час якого можна вплинути на її тривалість. Це етап проходження потоку через медіасервер, де виникає затримка трансмуksингу (перепакування пакетів з одного протоколу в інший).

3 Вивчена історія розвитку медіасерверних платформ і сформульовані технічні вимоги до нового ПЗ для медіасерверу. Найбільш вагомими серед них є:

- найменша часова затримка, що виникає під час трансмуksингу;
- висока відмовостійкість при навантаженні – 99, 8%;
- можливість автоматизації деяких процесів (зокрема, переспрямування потоків на YouTube і Facebook);
- відповідність критеріям безпеки;
- можливість розгортання системи та її масштабування за рахунок Cloud-технологій;
- підтримка більшості протоколів передачі.

4 Обрано платформи, що відповідають вимогами, емпірично виміряна затримка трансмуksингу для кожної з них. Платформа Wowza Streaming Engine має найменшу затримку (1,5с). Це ПЗ було обране для модернізації системи онлайн-трансляцій. У якості додаткових серверних

потужностей був обраний сервіс Wowza Streaming Cloud, який дозволив зменшити кількість апаратних серверів з чотирьох до одного.

5 Була налаштована і протестована платформа Wowza Streaming Engine. Зокрема, мінімізувати затримку до 1,5с дозволило «тонке» налаштування. При цьому частина користувачів (або всі користувачі при нормальному навантаженні) прийматиме відеопотік в режимі реального часу з мінімальною затримкою (1,5с) без втрати якості, а решта - із запізненням, але зі збереженням HD-якості.

6 На базі API було написане власне програмне доповнення, що автоматизує процес переспрямування відеопотоку в режимі реального часу на популярні платформи YouTube і Facebook. Крім того на базі API був створений додатковий функціонал для інтеграції в систему онлайн-трансляцій компоненту pay-per-view.

7 Були наведені рекомендації для організації онлайн-трансляцій з круговим оглядом 360 градусів.

8 Були проведені розрахунки капітальних витрат проекту модернізації, які склали 96 350 грн., а також розрахунки економічного ефекту, що виникає внаслідок скорочення експлуатаційних витрат. Визначено, що експлуатаційні витрати зменшилися майже втричі, а економія склала 119 752,1 грн./рік. Тому економічний ефект, що виник внаслідок скорочення експлуатаційних витрат, дозволив окупити всі капітальні витрати на модернізацію системи вже протягом 2017 року.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. WOWZA STREAMING ENGINE 4 API REFERENCE DOCUMENTATION [Електронний ресурс]. – Режим доступу: <https://www.wowza.com/resources/serverapi/4.7.1/>.
2. Official Community MySQL Website [Електронний ресурс]. – Режим доступу: <https://dev.mysql.com/downloads/connector/j/5.0.html>.
3. How to extend Wowza Streaming Engine using the Wowza IDE [Електронний ресурс]. – Режим доступу: <https://www.wowza.com/docs/how-to-extend-wowza-streaming-engine-using-the-wowza-ide>.
4. Official Website Advanced broadcasting electronics [Електронний ресурс]. – Режим доступу: <http://www.abe.it/default.aspx?lang=eng.&AspxAutoDetectCookieSupport=1>.
5. Вікіпедія – вільна енциклопедія [Електронний ресурс]. – Режим доступу: <https://en.wikipedia.org/>.
6. Мониторинг и аналитика стриминга видео [Електронний ресурс]. – Режим доступу: <http://itmultimedia.ru/monitoring-i-analitika-striminga-video/>.
7. Стриминг видео 360 градусов с помощью Wowza Streaming Engine. Как организовать? [Електронний ресурс]. – Режим доступу: <http://itmultimedia.ru/striming-video-360-gradusov-s-pomoshhyu-wowza-streaming-engine-kak-organizovat/>.
8. Экшен-камера, которая сама умеет стримить поток [Електронний ресурс]. – Режим доступу: <http://itmultimedia.ru/category/videotranslyacii/page/3/>.
9. Медиа сервера: Flash Media Server, Wowza, Erlyvideo, Red5 [Електронний ресурс]. – Режим доступу: <http://itmultimedia.ru/media-servera-flash-media-server-wowza-erlyvideo-red5/>.
10. Система онлайн видео-вещания. От постановки задачи, до ее решения [Електронний ресурс]. – Режим доступу: <http://itmultimedia.ru/sistema-onlajn-video-veshaniya-ot-postanovki-zadachi-do-ee-resheniya/>.

11. Wowza Streaming Engine. Виртуальная реальность и камеры 360 градусов [Электронный ресурс]. – Режим доступа: <http://itmultimedia.ru/wowza-streaming-engine-virtualnaya-realnost-i-kamery-360-gradusov/>.
12. Возможности и функционал Wowza медиасервера [Электронный ресурс]. – Режим доступа: <http://itmultimedia.ru/vozmozhnosti-i-funktional-wowza-mediaservera/>.
13. Вещание онлайн-видео с помощью Nginx [Электронный ресурс]. – Режим доступа: <https://habrahabr.ru/post/145867/>.
14. HLS против RTMP — сухая статистика [Электронный ресурс]. – Режим доступа: <https://habrahabr.ru/post/232297/>.
15. Шуйтенов Г.Ж. Технологии передачи данных в системах видеоконференции /Г.Ж.Шуйтенов, Т.Ж.Айдынбай // Научный журнал Евразийского национального университета имени Л.Н.Гумилева. – 2015. – №4.- с.77-83.
16. Official Website Nimble Streamer [Электронный ресурс]. – Режим доступа: <https://wmspanel.com/nimble>.
17. Трансляция онлайн-видео с минимальной задержкой [Электронный ресурс]. – Режим доступа: <https://habrahabr.ru/post/265675/>.
18. Steven P. Reiss. Visualizing the Java heap to detect memory problems / Steven P. Reiss // Visualizing Software for Understanding and Analysis, 2009. VISSOFT 2009. 5th IEEE International Workshop, 25-26 Sept, 2009. – Edmonton: 2009.
19. J. Barros. Effective Delay Control in Online Network Coding / J. Barros, R.A.Costa, D.Munaretto // INFOCOM 2009, IEEE, 19-25 April, 2009. – Rio de Janeiro: 2009.
20. Eustathia Ziouva. CSMA/CA performance under high traffic conditions: throughput and delay analysis / E.Ziouva, T. Antonakopoulos // Computer Communications, 15 February, 2002., Pages 313-321.
21. S. Sen. Online smoothing of variable-bit-rate streaming video / S. Sen, J.L.Rexford, J.K.Dey // IEEE Transactions on Multimedia, 1 March, 2000. Pages 37-

48.

22. Bolun Wang. Anatomy of a Personalized Livestreaming System / B. Wang, G.Wang, X.Zhang // IMC '16 Proceedings of the 2016 Internet Measurement Conference, 14-16 Nov, 2016. - Santa Monica:2016. Pages 485-498.

23. Matti Siekkinen. A First Look at Quality of Mobile Live Streaming Experience: the Case of Periscope / M.Siekkinen, E.Masala // IMC '16 Proceedings of the 2016 Internet Measurement Conference, 14-16 Nov, 2016. - Santa Monica:2016. Pages 477-483.

24. R. Otsuka. Transpost: a novel approach to the display and transmission of 360 degrees-viewable 3D solid images / R. Otsuka, T. Hoshino // IEEE Transactions on Visualization and Computer Graphics, 2 April, 2006. Pages 178-185.

25. Teddy Mantoro. Live video streaming for mobile devices: An application on android platform / T.Mantoro, D.Jatikusumo // Uncertainty Reasoning and Knowledge Engineering (URKE), 2012 2nd International Conference on, 14-15 Aug, 2012. - Jalarta:2012.

26. A.Bechtolsheim. Media server system / A.Bechtolsheim, David R. Cheriton // Patent, 24 April, 2007.

27. Mengjou Lin. Method and system for supporting constant bit rate encoded MPEG-2 transport over local ATM networks / Mengjou Lin, D. Singer // Patent, 21 April, 1998.

28. Mark Kressin. Method and apparatus for video conferencing system with 360 degree view / M.Kressin // Patent, 23 May, 2002.

29. B. Edelman. Real-time priority-based media communication / B. Edelman, J.Gay, S.Lozen // Patent, 18 Sept, 2007.

30. A.Aloman. Performance Evaluation of Video Streaming Using MPEG DASH, RTSP, and RTMP in Mobile Networks / A.Aloman, M.D.Canno, A.I.Ispas // IFIP Wireless and Mobile Networking Conference (WMNC), 2015 8th, 5-7 Oct, 2015. – Munich:2015.

ДОДАТОК А. Перелік документів на оптичному носії

- 1 Пояснювальна записка на 88 сторінках «Пояснювальна записка Бублик Ю.О.docx»
- 2 Копія наукової публікації «Копія наукової публікації.pdf»
- 3 Презентація.pptx

ДОДАТОК Б. Перелік публікацій з досліджуваної тематики

- 1 Y. Krasovska Minimum delay achievement in video online broadcasting / Krasovska Y.O., Magro V. I., Zuyenok I. I. // WIDENING OUR HORIZONS : The 12 th International Forum for Students and Young Researchers, April 20-21, 2017, Volume 2 – Dnipro, 2017. - P. 32-33.

ДОДАТОК В. Лістинг проекту модуля для інтеграції в систему онлайн-трансляцій компоненту pay-per-view

```

package com.produck.wowza;

import java.sql.*;

//додаємо необхідні класи

import com.wowza.wms.application.*;
import com.wowza.wms.amf.*;
import com.wowza.wms.client.*;
import com.wowza.wms.module.*;
import com.wowza.wms.request.*;
import com.wowza.wms.stream.*;
import com.wowza.wms.rtp.model.*;
import com.wowza.wms.httpstreamer.model.*;

import
com.wowza.wms.httpstreamer.cupertinostreaming.httpstreamer.*;
import com.wowza.wms.httpstreamer.smoothstreaming.httpstreamer.*;

//додаємо необхідні класи

public class ModuleSecureRtmpConnect extends ModuleBase {

    public void onAppStart(IApplicationInstance appInstance)
    {
        // preload the driver class

        //підключення драйверу з'єднання з MySQL-базою

try
    {
        Class.forName("com.mysql.jdbc.Driver").newInstance();
    }

    catch (Exception e)

```

```

    {
        getLogger().error("Error loading: com.mysql.jdbc.Driver:
"+e.toString());

        //у разі успіху додаємо запис про успішне підключення до лог-файлу
сервера
    }
}

// власноруч описуємо метод onConnect – подія спрацьовує при підключенні
користувача

public void onConnect(IClient client, RequestFunction function,
AMFDataList params)
{
String secretKey = client.getQueryStr();

//отримуємо параметр перевірки, який передавали у посиланні
rtmp://origin.visionmark.com/9tv?123456/9tv

getLogger().info("MyModule" + client.getQueryStr());

//записуємо його до логфайлу

Connection conn = null;

try
{
    conn =
    DriverManager.getConnection("jdbc:mysql://localhost/wowza?use
r=XXXXXX&password=XXXXXXXX");

//з'єднуємося з дб

    Statement stmt = null;

    ResultSet rs = null;

//виставляємо змінні до первинного вигляду

    try
    {
        stmt = conn.createStatement();
    }
}
}

```

```
rs = stmt.executeQuery("SELECT count(*) as userCount FROM
ppv where secret_key = '"+secretKey+"' and
allow='true'");
```

//робимо запит до БД, відповідь пишемо до змінної rs

//запит означає: обрати користувачів, у яких key=secretKey та дозвіл у параметрі true

//якщо результат більше 0, такі користувачі є і можна надати права на перегляд, якщо ні, то від'єднуємо

```
if (rs.next() == true)
{
if (rs.getInt("userCount") > 0)
{
client.acceptConnection();
```

//безпосередній дозвіл на підключення

```
getLogger().info(" MyModule: " + secretKey + " connect
client: " + client.getClientId());
```

//запис цієї події до лог-файлу

```
    }
}
}

catch (SQLException sqlEx)
{
getLogger().error("sqlexecuteException: " +
sqlEx.toString());
}

finally
```

```
{
```

//закриваємо створені конекти до бази

```
if (rs != null)
{
```

```
        try
        {
            rs.close();
        }
        catch (SQLException sqlEx)
        {
            rs = null;
        }
    }

    if (stmt != null)
    {
        try
        {
            stmt.close();
        }
        catch (SQLException sqlEx)
        {
            stmt = null;
        }
    }
}

conn.close();
}
catch (SQLException ex)
{
    // handle any errors
    System.out.println("SQLException: " + ex.getMessage());
}
```

```

        System.out.println("SQLState: " + ex.getSQLState());
        System.out.println("VendorError: " + ex.getErrorCode());
    }

    getLogger().info("onConnect MyModule: " + secretKey + "-" +
client.getClientId());
}

static public void onConnectAccept(IClient client)
{
    getLogger().info("onConnectAccept MyModule: " +
client.getClientId());
}

static public void onConnectReject(IClient client)
{
    getLogger().info("onConnectReject MyModule: " +
client.getClientId());
}

static public void onDisconnect(IClient client)
{
    getLogger().info("onDisconnect MyModule: " +
client.getClientId());
}
}

```


ДОДАТОК Е. Відгук керівника дипломної роботи

ВІДГУК

на дипломний проект

студентки Бублик Ю.О., гр. 172м-16-1

на тему: «Модернізація телекомунікаційної системи онлайн-трансляцій, що працює у режимі реального часу»

Актуальність теми: вже багато років спортивні події по всьому світу пов'язані з необхідністю онлайн-трансляцій в режимі реального часу. У цій галузі надзвичайно популярним є тоталізатор. При цьому ставки роблять люди з різних куточків планети і кожна секунда затримки онлайн-трансляцій може коштувати їм великих грошей. Тому мінімізація затримки і своєчасна модернізація систем онлайн-трансляцій - актуальні задачі.

Повнота розкриття теми: тема дипломної роботи розкрита повністю.

Теоретичний рівень дипломної роботи - достатньо високий.

Практична значущість: у роботі запропонована модернізація системи онлайн-трансляцій, яка успішно втілена на практиці у двох мас-медіа компаніях Дніпропетровщини.

Самостійність виконання роботи: робота виконана самостійно.

Якість оформлення, загальна та спеціальна грамотність: оформлення роботи відповідає вимогам, які до неї пред'являються.

Переваги на недоліки роботи: робота має практичне застосування та вагоме економічне обґрунтування, до недоліків роботи слід віднести одночасне переслідування декількох цілей замість однієї, чого не можна уникнути при практичному застосуванні здобутих за спеціальністю знань.

Загальна оцінка роботи та висновок, щодо рекомендацій до захисту в ДЕК: в цілому дипломна робота справляє дуже позитивне враження та задовольняє вимогам, що пред'являються. Вважаю, що дипломна робота заслуговує оцінки «відмінно», а її автор, Бублик Ю.О., присвоєння кваліфікації «інженера в галузі електроніки та телекомунікацій» за напрямом 172 — телекомунікації та радіотехніка.

Науковий керівник

К. ф.-м. н. доцент кафедри БІТ

_____ (Магпо В.І.)

“15” січня 2018 р.