

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

ННК “Інститут прикладного системного аналізу”  
(повна назва інституту/факультету)

Кафедра Системного проектування  
(повна назва кафедри)

«До захисту допущено»

Завідувач кафедри

\_\_\_\_\_ А.І.Петренко  
(підпис) (ініціали, прізвище)

“ \_\_\_\_ ” \_\_\_\_\_ 2016 р.

**Дипломна робота**

першого (бакалаврського) \_\_\_\_\_ **рівня вищої освіти**  
(першого (бакалаврського), другого (магістерського))

зі спеціальності 7.05010102, 8.05010102 Інформаційні технології проектування  
7.05010103, 8.05010103 Системне проектування  
(код та назва спеціальності)

на тему: \_\_\_\_\_ Інтернет-радіо \_\_\_\_\_

Виконав: студент 4 курсу, групи ДА-21  
(шифр групи)

\_\_\_\_\_ Шеренковський Артем Олегович \_\_\_\_\_  
(прізвище, ім'я, по батькові) (підпис)

Керівник доцент, к.т.н Гіоргізова-Гай В.Ш. \_\_\_\_\_  
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант економічний професор, к.е.н. Семенченко Н.В. \_\_\_\_\_  
(назва розділу) (посада, вчене звання, науковий ступінь, прізвище, ініціали) (підпис)

Рецензент \_\_\_\_\_  
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Нормоконтроль \_\_\_\_\_ ст. викладач Бритов О.А. \_\_\_\_\_

Засвідчую, що у цій дипломній роботі  
немає запозичень з праць інших авторів  
без відповідних посилань.

Шеренковський А.О. \_\_\_\_\_  
(підпис)

Київ – 2016 року

**Національний технічний університет України  
«Київський політехнічний інститут»**

Факультет (інститут) ННК "Інститут прикладного системного аналізу"  
(повна назва)

Кафедра Системного проектування  
(повна назва)

Рівень вищої освіти Перший(Бакалаврський)  
(перший (бакалаврський), другий (магістерський) або спеціаліста)

Спеціальність 7.05010102, 8.05010102 Інформаційні технології проектування  
7.05010103, 8.05010103 Системне проектування  
(код і назва)

**ЗАТВЕРДЖУЮ**  
Завідувач кафедри  
А.І.Петренко  
(підпис) (ініціали, прізвище)

«    » \_\_\_\_\_ 2016 р.

**ЗАВДАННЯ**  
на дипломний проект (роботу) студенту  
Шеренковському Артему Олеговичу  
(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) ) Інтернет-радіо  
керівник проекту (роботи) ) Гіоргізова-Гай Вікторія Шалвівна доцент, к.т.н.  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від 12 травня 2016 р. № 50-ст

2. Строк подання студентом проекту (роботи) 08.06.2016

3. Вихідні дані до проекту (роботи)

Параметри фізичного серверу.

Сайт кафедри створений за допомогою CMS Joomla!

Підтримка розповсюджених браузерів та медіа плеєрів.

Максимальна кількість паралельних потоків  $\leq 100$ .

4. Зміст розрахунково-пояснювальної записки (перелік завдань, які потрібно розробити)

1. Розглянути принцип побудови Інтернет-радіо: архітектура, основні протоколи і формати даних, особливості передачі потокового трафіку.
2. Провести аналіз програмного забезпечення для Інтернет-радіо, порівняти ПЗ розповсюджених медіа серверів з метою обґрунтування його вибору.
3. Провести аналіз вимог до сервісу, розробити план робіт для його реалізації.
4. Реалізувати сервіс в мережі кафедри СП: встановити і налаштувати медіа сервер, реалізувати підключення до сайту кафедри.
5. Провести тестування роботи медіа серверу.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслеників, плакатів тощо)

1. Структурна схема організації Інтернет радіо – плакат.
2. Порівняльна характеристика розповсюджених безкоштовних медіа серверів – плакат.
3. Налаштування програмного забезпечення для реалізації сервісу - плакат.

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Семенченко Н.В., професор, д.е.н.		

7. Дата видачі завдання 01.02.2016

### Календарний план

№ з/п	Назва етапів виконання дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Отримання завдання	01.02.2016	
2	Збір інформації	28.02.2016	
3	Вивчення принципів побудови Інтернет радіо	15.03.2016	
4	Проведення порівняльної характеристики медіа серверів	30.03.2016	
5	Розробка технічного завдання	10.04.2016	
6	Встановлення програмних модулів на сервер	20.04.2016	
7	Тестування сервісу Інтернет радіо	25.04.2016	
8	Оформлення дипломної роботи	31.05.2016	
9	Отримання допуску до захисту та подача роботи в ДЕК	08.06.2016	

Студент

\_\_\_\_\_

(підпис)

А.О.Шеренковський  
(ініціали, прізвище)

Керівник проекту (роботи)

\_\_\_\_\_

(підпис)

В.Ш.Гіоргізова-Гай  
(ініціали, прізвище)

# АНОТАЦІЯ

бакалаврської дипломної роботи Шеренковського Артема Олеговича  
на тему «Інтернет радіо»

Дана дипломна робота присвячена реалізації медіа серверу для Інтернет радіомовлення у 26 корпусі НТУУ «КПІ», до якого можна підключати аудіо потоки, що йдуть від різних радіостанцій. Метою роботи є встановлення програмного забезпечення для радіомовлення на сервер кафедри та організація доступу до нього із сайту кафедри.

В роботі розглянуто особливості збереження та передачі аудіо контенту через мережу Інтернет. Проаналізовано принципи побудови Інтернет радіо. Проведено порівняння існуючих програмних засобів для створення аудіо потоку, реалізації медіа сервера та програвачів, що можуть відтворювати аудіо потік. В ході виконання роботи було встановлене та налаштоване програмне забезпечення медіа серверу на фізичний сервер кафедри, розглянуто методи підключення клієнта до медіа серверу як стандартними засобами HTML, так і за допомогою модулів на сайтах, створених засобами систем керування вмістом. Робота була апробована в одній міжнародній науково-технічній конференції «SAIT-2016».

Загальний обсяг роботи: 80 сторінки, 26 рисунків, 6 таблиці, 21 посилання.

Ключові слова: Інтернет радіо, медіа сервер, аудіо, потокова передача даних, плеєр.

# АННОТАЦИЯ

бакалаврской дипломной работы Шеренковского Артема Олеговича  
на тему «Интернет радио»

Данная дипломная работа посвящена реализации медиа сервера для Интернет радиовещание в 26 корпусе НГУУ «КПИ», к которому можно подключать аудио потоки, идущие от разных радиостанций. Целью работы является установка программного обеспечения для радиовещания на сервер кафедры и организация доступа к нему с сайта кафедры.

В работе рассмотрены особенности сохранения и передачи аудио контента через сеть Интернет. Проанализировано принципы построения Интернет радио. Проведено сравнение существующих программных средств для создания аудио потока, реализации медиа сервера и проигрывателей, которые могут воспроизводить аудио поток. В ходе выполнения работы было установлено и настроено программное обеспечение медиа сервера на физический сервер кафедры, рассмотрены методы подключения к медиа серверу как стандартными средствами HTML, так и с помощью модулей на сайтах, созданных средствами систем управления содержимым. Работа была апробированная в одной международной научно-технической конференции «SAIT-2016».

Общий объем работы: 80 страницы, 26 рисунков, 6 таблиц, 21 ссылка.

Ключевые слова: Интернет радио, медиа сервер, аудио, потоковая передача данных, плеер.

# ANNOTATON

a bachelor`s degree work of Artem Sherenkovskiy  
entitled “Internet radio”

This course is devoted to realization of media server for Internet broadcasting in NTUU “KPI” department 26, to which streams from different radio stations could connect. The aim is installing the software for the broadcast server at the department and organization of access to it from the department website.

The paper discusses the features of the preservation and transmission of audio content via the Internet. Were analyzed the principles of Internet radio structure. Were made comparison of existing software for creation audio stream, software for implementing a media server and player which can play the audio. During the performance media server software has been installed and configured on the physical server of the department. Were described methods of connecting to media server with standard HTML tools or with modules on websites created using content management systems. The work was tested in one international scientific and technological conference “SAIT-2016”.

Total volume of work: 80 pages, 26 figures, 6 tables, 21 links.

Keywords: Internet radio, media server, audio, streaming content transmission, player.

## ЗМІСТ

<b>ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ.....</b>	<b>10</b>
<b>ВСТУП.....</b>	<b>11</b>
<b>1 ЗАГАЛЬНІ ПРИНЦИПИ ПОБУДОВИ ІНТЕРНЕТ РАДІО .....</b>	<b>13</b>
1.1 Формати цифрового аудіо .....	13
1.2 Особливості передачі голосового трафіку .....	22
1.3 Архітектура Інтернет радіо. Протоколи, що використовуються .....	24
1.4 Висновки .....	33
<b>2 АНАЛІЗ І ПОРІВНЯЛЬНА ХАРАКТЕРИСТИКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ІНТЕРНЕТ РАДІО .....</b>	<b>34</b>
2.1 Програмне забезпечення для створення аудіо потоку .....	34
2.2 Програмне забезпечення серверів для Інтернет радіо .....	37
2.3 Програмне забезпечення клієнтів для Інтернет радіо .....	47
2.4 Висновки .....	48
<b>3 АНАЛІЗ ВИМОГ ДО СЕРВІСУ І ПЛАН РОБІТ ДЛЯ ЙОГО РЕАЛІЗАЦІЇ .....</b>	<b>49</b>
3.1 Вимоги до сервісу .....	49
3.2 План робіт для реалізації сервісу .....	49
3.3 Висновки .....	50
<b>4 РЕАЛІЗАЦІЯ СЕРВІСУ ІНТЕРНЕТ РАДІО В МЕРЕЖІ КАФЕДРИ СП.....</b>	<b>51</b>
4.1 Встановлення та налаштування медіа серверу .....	51
4.2 Підключення сервісу до сайту .....	52
4.3 Тестування .....	58



	9
4.4 Висновки .....	59
<b>5 ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ ПРОГРАМНОГО</b>	
<b>ПРОДУКТУ .....</b>	<b>60</b>
5.1 Постановка задачі техніко-економічного аналізу .....	61
5.1.1 Обґрунтування функцій програмного продукту .....	61
5.1.2 Варіанти реалізації основних функцій .....	62
5.2 Обґрунтування системи параметрів ПП .....	65
5.2.1 Опис параметрів.....	65
5.2.2 Кількісна оцінка параметрів .....	65
5.2.3 Аналіз експертного оцінювання параметрів.....	68
5.3 Аналіз рівня якості варіантів реалізації функцій.....	71
5.4 Економічний аналіз варіантів розробки ПП.....	73
5.5 Вибір кращого варіанта ПП техніко-економічного рівня.....	75
5.6 Висновки до економічного розділу розділу .....	75
<b>ВИСНОВКИ .....</b>	<b>77</b>
<b>ПЕРЕЛІК ПОСИЛАНЬ.....</b>	<b>79</b>

## **ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ**

ІКМ – імпульсно кодова модуляція

ПЕОМ – персональна електронна обчислювальна машина

ПК - персональний комп'ютер

ЦАП – цифро-аналогового перетворювача

АЦП - аналого-цифрового перетворювача

FEC - Forward Error Correction

CMS – content management system

AAC – Advanced Audio Config

WAV – Waveform Audio Format

AIFF – Audio Interchange File Format

FLAC – Free Lossless Audio Codec

WMA 9 – Windows Media Audio 9 Lossless

## ВСТУП

Незважаючи на велику кількість медіа контенту в мережі Інтернет, радіомовлення через всесвітню павутину у сьогоднішні не лише не втрачає своїх позицій, а й навпаки знаходить все більше й більше прихильників.

IP-радіо(інтернет радіо) – це група технологій передачі поточкових аудіо даних через мережу Інтернет для здійснення широкомовних передач. Також під терміном інтернет-радіо чи веб-радіо часто розуміють радіостанцію, що використовується для поточкового мовлення у глобальній мережі Інтернет.

Інтернет-радіоприймач – це новий клас пристроїв, котрий дає можливість прослуховувати радіостанції, що віщають таким чином через Інтернет. Із зростанням пропускної здатності каналів, що доступні звичайним користувачам, з'явилася можливість передачі поточкового медіа контенту через мережу Інтернет, аналогічно до звичайного радіомовлення. Проте на відмінну від звичайного радіо, його інтернет заміна повинна посилати поточкові данні персонально до кожного слухача, що створює великі проблеми в умовах обмеженої пропускної здатності. Також необхідно припиняти мовлення до відсутніх слухачів. Усі ці проблеми потребують спеціальних підходів до архітектури програмного забезпечення, що дозволяють фільтрувати користувачів, та віщати з мінімальним використанням апаратних ресурсів.

За даними статистики, кількість інтернет радіослухачів у США станом на 2014 рік була близькою до 160 мільйонів осіб, що складає близько 50% всього населення Сполучених Штатів, та близько 67% американців, що користуються інтернетом. Разом із тим зростає час, який користувачі проводять слухаючи онлайн-радіо [1, 4].

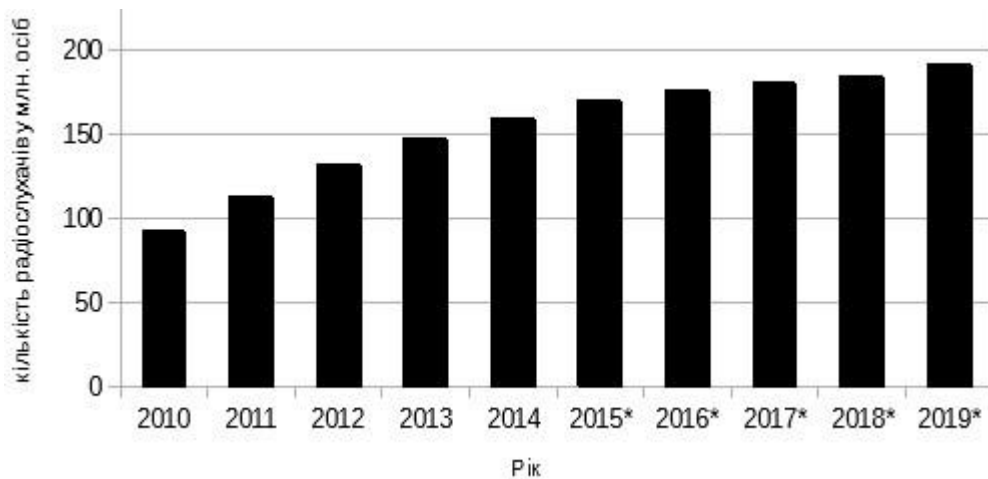


Рисунок 1 - Графік збільшення кількості інтернет-радіослухачів(\* - прогнозоване збільшення) [2]

Основною метою дипломної роботи є розробка працюючого інтернет-радіо медіа серверу, що буде приймати сигнал від певного джерела, та розсилати всім користувачам, що будуть до нього підключенні.

# 1 ЗАГАЛЬНІ ПРИНЦИПИ ПОБУДОВИ ІНТЕРНЕТ РАДІО

Зазвичай типовою схемою побудови Інтернет радіо є така: є одна або декілька радіостанцій(залежить від можливостей медіа серверу), на котрих відбувається запис та кодування аудіо потоків, що проходять через звукову карту ПЕОМ. Далі потік відправляється через мережу Інтернет на сервер мультимедіа, паралельно можливе зберігання на диск. Медіа сервер, що відповідає за підключення та підтримання зв'язку з користувачами сервісу, та відправку потокового аудіо до слухачів. Слухачі можуть відтворювати аудіо за допомогою певних спеціальних програм-програвачів.

Головною проблемою буде обмеження пропускної здатності каналу підключення до мережі Інтернету. При одночасному підключенні 100 слухачів до медіа серверу і якості потоку 160 kbps, мінімально допустима пропускна здатність підключення до Інтернету буде 16 Mbps.

Показником успіху роботи є готовий медіа сервер, який:

- Може підключати до себе клієнтів і надсилати ним потоки аудіо даних.
- Стабільна робота при максимальній завантаженості серверу, тобто час затримки відправлення пакету не повинен перевищувати певних норм, аби аудіо програвалось чітко на кінцевому програвачі.
- Можливість ручного регулювання певних параметрів серверу (кількість користувачів, завантаженість каналу доступу в мережу Інтернет та ін.)

## 1.1 Формати цифрового аудіо

Звукова хвиля представляє собою одновимірну акустичну хвилю (хвилю тиску). Коли така хвиля досягає людського вуха, барабана е починає вібрувати,

викликаючи вібрацію тонких кісток внутрішнього вуха, у результаті чого в мозок по нерву посилається сигнал, що розпізнається, як звук.

Цифрове аудіо - це цифрове представлення аудіо хвилі, за допомогою якого її можна відтворити. Звукові хвилі можна перетворювати у цифрову форму за допомогою АЦП. На вхід АЦП подається електрична напруга, у відповідності з частотою коливань хвилі, а на виході отримується бінарна послідовність. На рис. 2а зображено приклад синусоїдальної хвилі. Аби представити сигнал у цифровому вигляді, ми повинні заміряти значення сигналу (відліки) через рівні проміжки часу  $\Delta T$ , як показано на малюнку 2б. Якщо звукова хвиля не є чисто синусоїдальною, а представляє собою суму декількох синусоїдальних хвиль, а найбільша частота її складових дорівнює  $f$ , то згідно із законом Найквіста, для того щоб у майбутньому можна було відновити хвилю, частота дискретизації повинна бути не менша за  $2 \cdot f$ . При зворотному процесі, цифрові значення переводяться у аналогову електричну напругу за допомогою ЦАП, і вже потім можна аналоговий сигнал перевести у звукову хвилю.

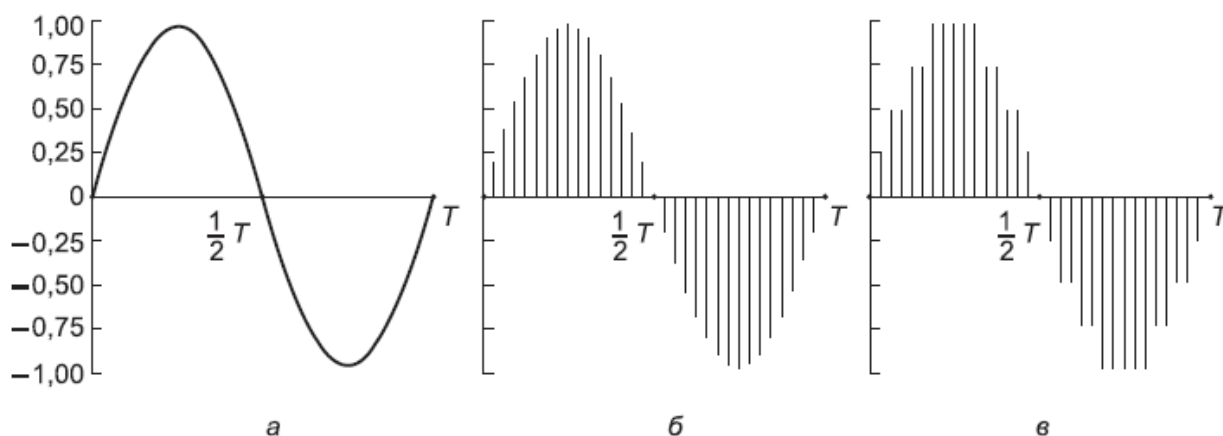


Рисунок 2 - Хвиля: а - синусоїдальна; б - дискретизація; в - квантування відліків 4 бітами [3]

Наприклад, відліки на малюнку 2в можуть приймати лише 9 значень - від -1 до 1 із кроком 0.25. Похибка, що виникає в результаті неточної відповідності дискретизованого сигналу аналоговому називають шумом квантування

(quantization noise). При малій кількості біт, котрими представляється кожен відлік сигналу, цей шум може ставати настільки великим, що наслух він буде здаватися наче спотворення вихідного сигналу, чи як сторонні шуми.

Найбільш відомі приклади використання цифрового звуку це:

- Телефон.
- Аудіо компакт-диски.

В імпульсно-кодовій модуляції, що використовується у телефонній системі, використовуються восьмибітні відліки, що заміряються 8000 разів у секунду, тобто гармоніки, що мають частоту вищу за 4000 Гц втрачаються. Шкала замірів нелінійна, це робиться заради мінімізації спотворення сигналу. У результаті це кодування забезпечує потік даних у 64000 біт за секунду.

Аудіо компакт-диски містять звуковий сигнал, відцифрований з частотою дискретизації 44100 Гц, у результаті чого вони можуть зберігати звук, що створюється гармоніками з частотою коливання більше за 22000 Гц. Відтворене звучання сприймається людиною, як достатньо якісний звук. Кожному відліку виділяється 16 біт, а його значення пропорційне амплітуді сигналу. Відлік величиною у 16 біт може приймати всього лише до  $2^{16} = 65536$  різних значень, у той час, коли людське ухо може чути діапазон більший ніж один мільйон значень. Таким чином, не дивлячись на те, що аудіо у CD-якості, значно краще за аудіо, що передається через телефон, використання 16 біт на відлік дає нам значний шум квантування, проте зазвичай якість такого звучання не викликає нарікань. Деякі цінителі гарного звучання обирають записи у форматі довгограючих платівок із 33 оборотами на хвилину, так як у платівок нема частотної межі у 22000 Гц та нема шуму квантування. При 44100 відліках за секунду по 16 біт кожен, для не стиснутого аудіо у CD-якості необхідна пропускна здатність у  $44100 * 16 = 705600$  біт/секунду = 705.6 кбіт/с, для монофонічного сигналу та 14.11 Мбіт/с для стереофонічного.

Аудіо контент часто стискається для того, щоб зменшити необхідну смугу пропускання каналу та час передачі даних. В усіх системах стиснення даних

повинно бути 2 алгоритми: один для стиснення, другий для їх подальшої розпаковки. Також ці алгоритми називаються алгоритмами кодування (encoding), та декодування (decoding), а пристрої що їх виконують, кодери та декодери відповідно.

В алгоритмах стиснення часто присутня певна асиметричність. Зазвичай контент кодується один раз і потім зберігається на сервері. Ця асиметричність означає, що алгоритм кодування може бути повільним та вимагати більш коштовного устаткування, при цьому алгоритм декодування повинен бути швидким и не вимагати великих ресурсних затрат. З іншого боку, для живого аудіо, такого як дзвінки через Skype або Viber, повільний процес кодування неприйнятний. Звідси можна зробити висновок, що мультимедіа реального часу використовує алгоритми та параметри, що відрізняються від аудіо, що зберігається на диску. Часто використовується менше стиснення. Друга ознака асиметрії це те, що процес кодування/декодування не завжди повинен бути зворотнім. Тобто при стисненні файлу, його передачі та наступній декомпресії користувач повинен отримувати усю інформацію до останнього біту. У випадку мультимедіа ця вимога необов'язкова. Зазвичай аудіо сигнал буде відрізнитись від початкового після декодування, проте людські органи сприйняття цього не будуть помічати. Коли після розпакування файл не повністю відповідає оригіналу, передача йде з втратою (lossy). Якщо вхідний та вихідний файли співпадають, передача йде без втрат (lossless). Системи з втратою важливі, бо при втраті незначної інформації ми отримуємо набагато краще стиснення.

При передачі даних у CD-якості теж бажано скоротити обсяг даних. Без стиснення для передачі аудіо в стерео якості потрібна ширина каналу в 1,411 Мбіт / с, що буде гальмувати широкосмугові канали, залишаючи менше місця для іншого веб-трафіку. Швидкість передачі даних можна зменшити на порядок завдяки стисненню, при цьому втрати якості будуть незначними або ж їх не буде зовсім.



Існують 2 основні концепції стиснення звуку:

- Кодування форми сигналів (waveform coding), коли сигнал розкладається на гармоніки за допомогою перетворення Фур'є. Амплітуда кожної компоненти кодується з мінімальними спотвореннями. Завданням є максимально акуратна передача форми сигналу з мінімально можливою витратою бітів.
- Перцептивне кодування (perceptual coding), метод, що базується на недоліках слухового апарату людини та науці психоакустиці. Ключовою властивістю даного алгоритму є те, що одні звуки можуть маскувати інші звуки. Властивість гучних звуків певного діапазону частот «приховувати» тихіші звуки інших діапазонів називається частотним маскуванням. Навіть після припинення гучного звуку, ще деякий час людина не буде чути тихіших звуків, це пов'язано з тим, що вухо має певну інерційність у процесі зміни чутливості. Цей ефект називається часовим маскуванням [3].

Як результат існує 3 варіанти збереження сигналу:

- Без стиснення – після перетворення аналогового сигналу у цифровий, він просто записується у файл, без попередніх стиснень. Найвідомішими форматами, що зберігають аудіо інформацію без стиснення є:
  - WAVE/WAV (Waveform Audio Format) – формат аудіо файлу розроблений компаніями Microsoft та IBM. WAVE базується на форматі RIFF (формат файлів-контейнерів, що використовується для збереження потокових даних), поширюючи його на інформацію про такі параметри аудіо, як застосований кодек, частота дискретизації та кількість каналів. WAV як і RIFF передбачався для комп'ютерів IBM PC, тому всі змінні записані у форматі little endian. Файли формату WAVE можуть записуватись за допомогою будь-якого кодеку, проте

зазвичай використовується ІКМ, що призводить до великих обсягів файлів. Також недоліком є обмеження розміру файлу до 4 ГБ, через 32-бітну змінну.

- AIFF (Audio Interchange File Format) – формат аудіо файлів, що був розроблений у 1988 компанією Apple Computers на основі формату IFF (формат файлів-контейнерів, що створений для спрощення обміну даними між ПЗ різних виробників; може містити в собі будь-які дані: звук, графіку, текст). Використовується для зберігання файлів на персональних комп'ютерах та інших електронних аудіо пристроїв. Як і WAV даний формат використовує ІКМ та має обмеження у 4 ГБ. Його особливістю є те, що він дозволяє разом із звуковою хвилею розміщувати додаткову інформацію, зокрема, приклади WaveTable (прикладі звучання інструментів разом із параметрами синтезатора), що покращує якість результату.
- Із стисненням без втрат – спосіб кодування, що дозволяє повністю відновити вихідні дані із стиснутого потоку. Найчастіше його використовують, коли необхідно повністю зберегти якість вихідної звукової інформації. Сучасні алгоритми стиснення без втрат дозволяють зменшити об'єм файлів більш ніж у два рази. Подібні кодери працюють за принципом звичайних архіваторів, проте використовують алгоритми, що адаптовані спеціально для роботи з аудіо. Найрозповсюдженими форматами, що використовують алгоритми стиснення без втрат є:
  - FLAC (Free Lossless Audio Codec) – формат, створений Xiph.Org Foundation у 2001 році. Даний кодек є безкоштовним і підходить для відтворення музики на апаратурі високої якості, а алгоритм кодування є повністю відкритим для доступу. Формат підтримує будь-які частоти дискретизації від 1Гц до 65535 Гц, а розрядність

аудіо може бути від 4-ох до 32-ох біт. Вважається, що FLAC у порівнянні з іншими lossless кодеками найбільш ефективно використовує ресурси системи при декодуванні. Нажаль це досягається за рахунок чималого часу кодування.

- WMA 9 (Windows Media Audio 9 Lossless) – розроблений компанією Microsoft для зберігання та трансляції аудіо інформації у 2003 році. Коефіцієнт стиснення варіюється від 1,75 до 3,5. Офіційно формат був випущений лише для ОС Windows та Mac OS X. Недоліком формату є те, що при пошкодженні певного байту, файл не можна прослуховувати в деяких випадках за 20 секунд до пошкодженої інформації та після неї.
- Із стисненням з втратами – використовується у випадках, коли необхідно якомога сильніше стиснути дані, зберегти при цьому якість на достатньому рівні. Зазвичай використовується перцептивне кодування. Найвідоміші формати, що використовують стиснення із втратами є:
  - MP3 (MPEG-1/2/2.5 Layer 3) – найпопулярніший формат стиснення на сьогоднішній день. Після видалення звуків, що не повинні чути, інформація кодується за методом Хаффмана. Особливістю є те, що програми, які кодують звук, не стандартизуються. Стандартам підпорядковуються лише декодери. Це призводить до того, що якість відтворення формату MP3 не завжди залежить від плеєра. Ще однією особливістю є JointStereo - замість кодування стерео, як двох незалежних каналів, відбувається кодування так званого центрального каналу та різниці, що відрізняє його від вихідних стереоканалів. Також цікавим є те, що кодер змінює ступінь стиснення «на льоту», в залежності від характеру звуку. Такий підхід призводить до зменшення розміру файлу.

- AAC (Advanced Audio Config) – формат аудіо файлу з меншою втратою якості при кодуванні, ніж MP3 при однакових розмірах файлу. Створювався як нащадок MP3 з покращеною якістю кодування.
- OOG Vorbis – вільний формат, що з'явився у 2002 році. Психоакустична модель за принципом дії схожа з MP3, проте математична обробка та програмна реалізація сильно відрізняються. Vorbis застосовує більш якісну психоакустичну модель, ніж конкуренти, що дає більшу чіткість відтворення при рівній густині потоку. Підтримує до 255 окремих каналів з частотою дискретизації до 192 кГц та розрядністю до 32 біт. Формат розроблявся з можливістю потокового мовлення. Це дає змогу в одному файлі зберігати декілька композицій із власними тегами. При завантаженні такого файлу у плеєр повинні відобразитися усі композиції, ніби їх завантажили із різних файлів. Використовується змінний бітрейт, що не має фіксованих значень, і може змінюватися навіть на 1 kbps. Наймолодший формат був створений аби замінити усі платні запатентовані аудіо формати та має підтримку на всіх відомих платформах: Windows, Linux, Symbian, DOS, FreeBSD та ін.

У файлах аудіо даних можуть також зберігатися певні додаткові відомості про композицію, такі як: ім'я композиції, її автор, обкладинка альбому та ін. Наприклад, для файлів розширення MP3 ця інформація зберігається у спеціальних тегах- мітках у межах файлу (на початку, або в кінці). У них можливе зберігання інформації про авторство, альбом, році випуску і інша інформація про трек. У пізніших версіях тегів можливе зберігання обкладинок альбомів і тексту пісні. Існують різні версії тегів:

- ID3 - формат метаданих. Найчастіше використовується разом з аудіоформатом MP3. Містить дані про назву трека, альбому, ім'я

виконавця тощо. Ця інформація може використовуватись, наприклад, програвачами мультимедіа для відображення відомостей про трек чи автоматичного сортування. Назва «ID3» — це скорочення «Identify a MP3». Існує дві несумісних версії ID3: ID3v1 та ID3v2. Хоч ID3 розроблявся для MP3, розробникам немає перешкод вбудувати його також і в інші формати.

- ID3v1 - після створення формату MP3 виникла проблема зберігання метаданих. MP3 цього не реалізовував. У 1996 році Еріку Кемпу прийшла ідея як вирішити цю проблему: додати маленький шматочок даних до файлу. Стандарт, тепер відомий як ID3v1 швидко став стандартом де-факто зберігання метаданих у MP3-файлі.
- ID3v2 - У відповідь на критику стандарту ID3v1 у 1998 році був розроблений стандарт ID3v2. Він мало схожий на першу версію. Існує три версії ID3v2: ID3v2.2 — перша публічна версія ID3v2. Використовує 3-символьний ідентифікатор кадру замість 4-символьного. Наприклад, TT2, а не TIT2. Тепер вважається застарілим. ID3v2.3 має 4-символьні ідентифікатори кадру. Кадр може мати кілька значень, розділених символом «/». На даний час, це найпопулярніший стандарт. ID3v2.4 — найновіша версія стандарту, представлена у листопаді 2000. Дозволяє зберігати текст у кодуванні UTF-8, замість UTF-16. Для розділення значень використовується нульовий байт, тому символ «/» можна вільно використовувати в тексті. Також додана можливість розміщати тег у кінці файлу, як у версії ID3v1.

Проаналізувавши методи створення аудіо файлів та визначивши їх переваги і недоліки можна сказати, що для Інтернет радіомовлення найбільш підходящими є формати із стисненням та втратами. Ці формати мають найбільшу

ступінь стиснення, що дуже важливо при передачі через мережу. Формат без стиснення можна використовувати для мовлення у локальній мережі проте він може дуже сильно її навантажити, а формат із стисненням без втрат можна використовувати при високошвидкісному підключенні до мережі Інтернет, бо його об'єм у декілька разів перевищує об'єм файлів створених із втратами. Найкращим із форматів, що стискають аудіо дані із втратами є OGG Vorbis, проте його недоліком є мала кількість використання порівняно з MP3 та AAC. Цей факт примушує до використання саме форматів MP3 та AAC.

## **1.2 Особливості передачі голосового трафіку**

Ідея відправки аудіо та відео через Інтернет, з'явилася ще у сімдесятих роках минулого століття, проте лише на початку XXI сторіччя передача мультимедіа у реальному часі (real-time audio) стала можливою. Трафік у режимі реального часу відрізняється від веб-трафіку тим, що передача повинна йти з певною швидкістю, аби мати сенс. Аби забезпечити такі вимоги відбулося дві події:

- По-перше комп'ютери стали потужніші, вони стали оснащуватися мікрофонами, що дало змогу легко вводити, обробляти і виводити аудіо дані.
- По-друге збільшилась пропускна здатність каналів, що з'єднують звичайних кінцевих користувачів із мережею Інтернет.

Враховуючи, що ширини полоси пропускання достатньо для передачі аудіо і навіть відео, ключовим питанням розробки додатків для проведення онлайн конференцій та передачі мультимедіа є мережева затримка та відсоток втрачених пакетів. Для аудіо необхідно, аби вони програвались з постійною швидкістю, аби бути корисним, адже запис що постійно переривається, або відтворюється у уповільненому режимі не принесе задоволення кінцевому користувачеві. При програванні музики через мережу абсолютна затримка не

грає ролі, адже вона впливає лише на те, коли почнеться програвання файлу. Проте зміна затримки, що називається нестійкою синхронізацією (jitter), все ще грає важливе значення. Це повинно маскуватися програвачем, у противному випадку аудіо буде звучати перериваючись [3]. Головною причиною постійності затримки є те, що при потоковій передачі даних, наприклад розмова через Skype, необхідно аби наступні дані надходили з однаковою швидкістю, аби у приймача вони відтворювалися плавно та без обривів. Для передачі без буферизації необхідно, аби затримка не перевищувала 150 мс, як це реалізовано у телефонній мережі (міжнародні виклики, що можуть мати час очікування до 400 мс викликають роздратованість користувача). Проте існує затримка, котру зменшити неможливо, відтак при передачі даних від Києва до Нью-Йорка, на відстань 8000 км, неможливо зробити затримку меншою за 30мс( час проходження світла між цими двома точками). В реальності затримка буде ще більшою через затримку маршрутизуючих пристроїв, та неідеальності середі через яку йде сигнал.

Другим фактором якості передачі є кількість загублених пакетів. При використанні протоколу UDP втрачені пакети не відправляються повторно, тому звук, що мав відтворюватись, буде відсутнім. Мовлення при якому постійно обривається сигнал не може задовільнити клієнта. Якщо ж живе мовлення транслюється з відставанням на 5-10 секунд від реального, то можна використовувати TCP, для коректної передачі даних без втрати пакетів. Якщо пакет втрачається, то інформація, що буферизована у програвачі клієнта і продовжує програватись дає час на повторну відправку втрачених пакетів. Проте в цьому випадку також є свої недоліки. По-перше на повтору відправку пакету може знадобитися декілька секунд, тому використання для онлайн розмов неможливе. По-друге при втраті лише 2-ох відсотків пакетів, завантаженість каналу виростає у декілька разів.

Отже при передачі аудіо сигналу основними факторами є:

- абсолютна затримка не важлива, вона впливає лише на те, коли почнеться програвання файлу.
- постійність затримки важлива аби потокові дані відтворювалися з постійною швидкістю, а звучання не обривалося.
- необхідно забезпечити надійність доставки пакетів для забезпечення безперервного звучання, або компенсацію невеликої кількості втрачених пакетів.

### **1.3 Архітектура Інтернет радіо. Протоколи, що використовуються**

В мережі можна легко транслювати аудіо потоки, котрі можуть складатись, як із збереженого контенту (наприклад MP3-файли на вашому комп'ютері), так і з живого мовлення (голос через мікрофон). Основними компонентами Інтернет радіомовлення є:

- Джерело, або радіостанція. В якості радіостанції може виступати комп'ютер, що підключений до мережі Інтернет, який використовується для змішування усіх аудіо потоків, що проходять через звукову карту ПК. В якості джерел може виступати музикальний файл, що програватиметься, подкаст(англ. «podcast», записи певних програм, що йшли на цьому радіо, збереженні на диск), живий голос із мікрофону чи потік аудіо контенту з інших пристроїв – CD/DVD програвач та інші. Комп'ютер перетворює звукову інформацію із звукової карти у «потік» і «на льоту»(англ. On-the-fly) відправляє на медіа сервер.
- Сервер, або медіа сервер. Медіа сервером може бути сервер, або звичайний персональний комп'ютер, на якому встановлене спеціальне програмне забезпечення для віщання через мережу



Інтернет. Основним завданням серверу є прийом потоку від радіостанції та його переадресація слухачам, що підключенні до даного медіа серверу. Зазвичай програмне забезпечення дозволяє керувати якістю вихідних потоків, кількістю підключених користувачів та іншими параметрами.

- Клієнт – програма на стороні користувача для прослуховування аудіо контенту Інтернет радіостанції. В її основні функції входить: декодування аудіо потоку від медіа серверу у звук, управління інтерфейсом користувача, обробка помилок передачі. На даний момент існує дуже багато музикальних програвачів для різних операційних систем, що можуть приймати потік даних з мережі Інтернет, та програвати його через пристрої виводу. Деякі можуть не лише декодувати «на льоту», а й буферизувати дані. Це дозволяє боротися з флуктуаціями (джиттером), а також надає можливість перемотати подкаст, пісню, або мовлення назад, щоб прослухати його ще раз.

Структура взаємодії усіх компонентів показана на рис. 3:

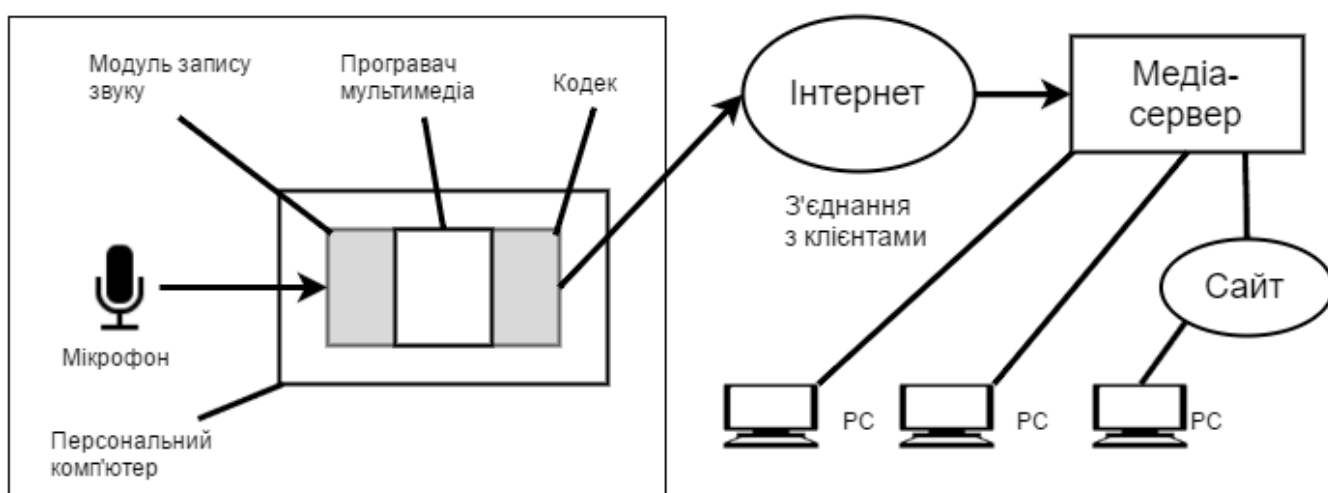


Рисунок 3 - Типова архітектура Інтернет радіо [3]

Популярністю користується не тільки збережене аудіо. Передача медіа у режимі реального часу також дуже популярна. Завдяки мовленню у режимі реального часу виникли нові стандарти та технології. Живе мовлення через Інтернет використовується основними телестанціями. Його зазвичай називають IPTV (IP TeleVision – IP-телебачення). Таке ж мовлення використовується і радіостанціями. Воно називається інтернет-радіо.

Основний підхід до інтернет радіомовлення полягає у тому, що слухачі підключаються до потоку, що віщається радіостанцією. Медіаплеєри клієнтів можуть давати такі можливості, як зупинка і перемотка назад, у той час, як живий медіа контент буде продовжувати передаватись і буферизуватись, доки користувач не буде готовий до прослуховування. Для програвача не важливо, відправляється контент у режимі реального часу, чи його джерелом є файл, що зберігається на диску.

Проте навіть при мовленні у режимі реального часу, все ще є необхідність у буферизації на стороні клієнта, аби згладити нестійкість синхронізації (jitter). Зазвичай, затримки початку програвання на 10-15 секунд достатньо.

Ще однією особливістю живого мовлення є те, що події, чи передачу, що передаються у реальному часі, зазвичай слухають чи дивляться сотні тисяч слухачів та глядачів відповідно. Передача великій кількості користувачів, таким чином, складається з багатьох окремих сесій передачі, що протікають в один і той же час.

Для цього було створено багато протоколів, серед яких є:

- RTP (Real-time Transport Protocol) – працює на прикладному рівні і використовується при передачі трафіку реального часу. Був розроблений в 1996 як RFC 1889 [11] та був замінений на RFC 3550 [12] у 2003 році. RTP не має стандартного зарезервованого номера порту. Єдиним обмеження полягає в тому, що з'єднання проходить з використання парного номеру порту, а наступний непарний використовується для зв'язку з протоколом RTSP. Встановлення і

розрив з'єднання не входить в список можливостей даного протоколу, такі дії зазвичай виконуються сигнальним протоколом (наприклад RTSP).

- RTSP (Real Time Streaming Protocol) – протокол розроблений у 1998 році й описаний у RFC 2326 [10]. Являється прикладним протоколом, що призначений для використання у системах, що працюють з мультимедіа даними, і дозволяє віддалено керувати потоком даних із серверу, даючи можливість виконання таких команд як «Старт», «Стоп», а також доступу за часом до файлів, розташованих на сервері. RTSP не виконує стиску, не визначає метод інкапсуляції мультимедійних даних та транспортні протоколи. Передача поточкових даних сама по собі не є частиною цього протоколу, для цього використовується RTP, проте деякі виробники реалізують власні транспортні протоколи.
- IGMP (Internet Group Management Protocol) – протокол керування груповою (multicast) передачею в мережах, базованих на протоколі IP. IGMP використовується маршрутизаторами для об'єднання мережевих пристроїв в групи. Протокол розташований вище мережевого рівня, хоча функціонує не як транспортний протокол. Він схожий з ICMP для односторонньої передачі.
- SDP (Session Description Protocol) – мережевий протокол, призначений для опису сеансу передачі поточкових даних, включаючи IP-телефонію, Інтернет радіо та програми мультимедіа. Не дивлячись на те, що SDP надає можливість опису мультимедіа даних, у ньому не вистачає механізмів узгодження параметрів сесії, котрі мають намір використовувати партнери. RFC 3264 представляє модель узгодження на основі механізму пропозиція / відгук.

Схема багатоадресної потокової трансляції працює наступним чином. Сервер один раз відправляє кожен пакет, використовуючи при цьому

багатоадресну передачу по IP групі адресатів (IP multicast). Усі клієнти, що хочуть підключитись до передачі, приєднуються до групи за допомогою IGMP, а не відправляючи RTSP-повідомлення на медіа сервер, так як медіа сервер вже відправляє живий потік (окрім випадку підключення першого клієнту).

Так як багатоадресна передача є сервісом доставки від одного до багатьох, медіа передається в RTP-пакетах через UDP-транспорт. TCP працює тільки між одним відправником і одним одержувачем. Так як UDP не забезпечує надійності, деякі пакети можуть бути втрачені. Щоб скоротити рівень втрат до прийняттого, ми можемо використовувати FEC і інтерлівінг.

У разі використання FEC виникає корисна взаємодія з багатоадресною передачею, як показано в прикладі з контрольним пакетом на рис. 4. Коли пакети передаються на багато адрес, різні клієнти можуть втрачати різні пакети. Наприклад, клієнт 1 втратив пакет В, клієнт 2 втратив контрольний пакет Р, клієнт 3 втратив пакет D, а клієнт 4 не втратив жодного пакета.

Однак, незважаючи на те що клієнти втратили різні пакети, в даному випадку всі користувачі зможуть відновити втрачену інформацію. Все що потрібно - це щоб жоден з клієнтів не втрачав більше одного пакета, тоді відсутній може бути відновлений за допомогою інформації з отриманих.

Для сервера з великою кількістю клієнтів багатоадресна передача медіа в RTP- і UDP-пакетах є найефективнішим способом. В іншому випадку сервер повинен передавати  $N$  потоків, коли у нього є  $N$  клієнтів, що потребують дуже великої пропускної здатності мережі з боку сервера. Проте на практиці все зовсім не так.

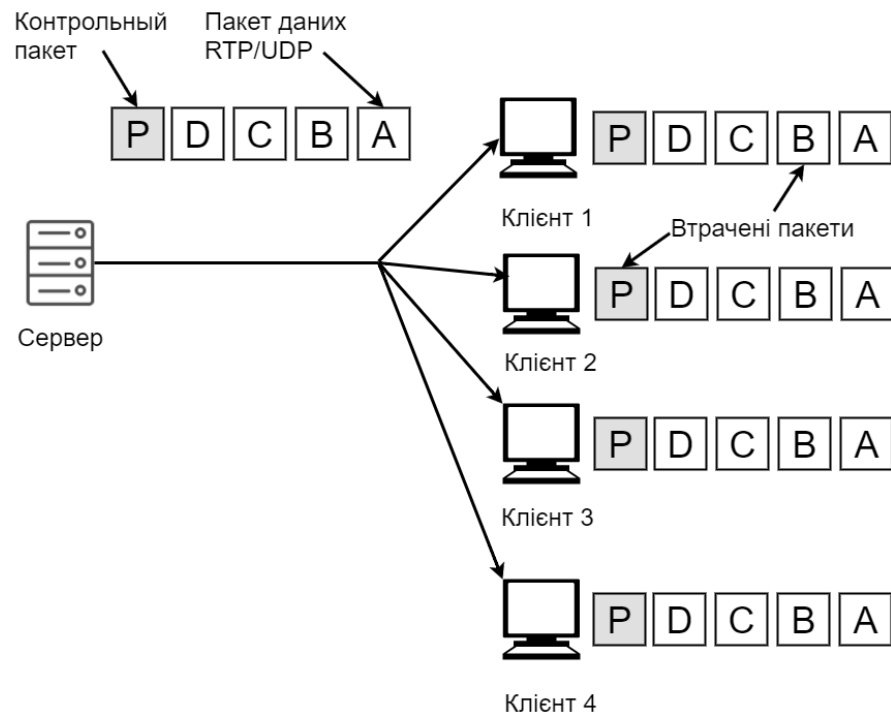


Рисунок 4 - Багатоадресна передача медіа з контрольним пакетом [3]

Зазвичай кожен користувач встановлює TCP-з'єднання з сервером, і медіа передається по цьому з'єднанню. Для клієнта це виглядає так само, як передача збережених медіафайлів. І так само як при ній, існує кілька причин для такого невдалого рішення. Перша причина полягає в тому, що багатоадресна передача по IP не дуже поширена в Інтернеті. Деякі інтернет-провайдери та мережі її підтримують, але зазвичай вона є недоступною за межами окремої мережі, що ставить хрест на глобальному мовленні. Іншою причиною є переваги TCP над UDP. Передачу за допомогою TCP зможуть прийняти практично всі користувачі Інтернету, особливо в тому випадку, якщо передача буде йти як HTTP для проходження через міжмережеві екрани, а надійна доставка медіа дозволить користувачам легко перемотувати контент назад.

Повертаючись до Інтернету, недолік передачі в режимі реального часу через TCP полягає в тому, що сервер повинен надсилати окрему копію медіа кожному клієнту. Це реально, якщо клієнтів не багато. Важливий момент полягає

в розміщенні сервера в місці з хорошим підключенням до Інтернету, щоб пропускної здатності каналу було досить.

Проте при великій кількості клієнтів використовувати TCP для передачі медіа кожному клієнту з одного сервера неможливо. Для цього буде не достатньо смуги пропускання каналу одного сервера. Для великих сайтів потокового мовлення використовується кілька медіа серверів, розосереджених географічно, так щоб клієнт міг підключитися до найближчого, для зменшення затримок мережі [3].

Отже основними компонентами Інтернет радіо є:

- Радіостанція, що створює потік та відправляє його на медіа сервер.
- Медіа сервер, що приймає потік від радіостанції та відправляє усім підключеним клієнтам.
- Програвач клієнта, що підключається до медіа серверу, отримує та відтворює аудіо потік.

При передачі аудіо даних використовується буферизація на стороні клієнта, що дозволяє знизити вплив джиттера. Буферизація 10-15 с звучання до початку програвання дозволяє зробити мовлення безперервним навіть за умови помітного джиттера у мережі. До тих пір, доки пакети прибувають раніше, ніж вони реально необхідні, немає ніякого значення, коли саме вони прибувають.

Для програвання ефіру при використанні програвача необхідно задати повну адресу потоку у форматі «`http://{serverIP}:{MediaServerPort}/{mountName}`», де `serverIP` – IP адреса серверу, або його домєне ім'я; `MediaServerPort` – це порт на якому працює медіа сервер; `mountName` – ім'я медіа потоку, до якого необхідно під'єднатись.

Якщо ж для під'єднання до серверу використовується сайт то, коли клієнт натискає клавішу «Грати» пересилається короткий метафайл, що містить URL серверу, звідки йде аудіо потік. Браузер, використовуючи MIME-тип чи розширення файлу, визначає підходящий допоміжний додаток (тобто програвач). Метафайл записується у тимчасовий файл, надалі відкривається

додаток-програвач, якому передається ім'я тимчасового файлу. Бачачи у ньому URL, програвач зв'язується із сервером і починає працювати як радіоприймач. Аудіо дані передаються у вигляді одного потоку, тому можлива робота по HTTP.

Незважаючи на велику кількість протоколів, створених для багатоадресної передачі потокового медіа контенту, радіомовлення через мережу використовує звичайний TCP для встановлення з'єднання між клієнтом та сервером та надійної передачі даних.

### Організація зв'язку із сайтом

Для програвання ефіру, що йде від медіа серверу усе що потрібно на сайті це тег `<audio></audio>`. У даного тегу є певні можливі атрибути, а саме:

- `autoplay` – при наявності даного атрибута аудіо потік буде відтворюватися одразу ж після завантаження сторінки. У якості значення вказується «autoplay», також можна взагалі не вказувати жодного значення.
- `controls` – атрибут, що додає панель керування до аудіо треку. Вид панелі та її зміст залежить від браузеру і може включати в себе: кнопку відтворення, паузи, полосу перемотки, повзунок для зміни рівня гучності та ін. У якості значення вказується «controls», також можна писати атрибут без значення.
- `loop` – зациклює аудіо, у випадку якщо програтється файл, аби воно нескінченно повторювалось після завершення. У якості значення вказується «loop», також можна писати атрибут без значення
- `muted` – вимикає звук при відтворенні музики. У якості значення вказується «muted», також можна взагалі не вказувати жодного значення.
- `preload` – атрибут використовується для завантаження аудіо файлу разом із завантаженням веб-сторінки. Ігнорується, якщо встановлений атрибут «autoplay». Можливі значення:

- none – не завантажувати аудіо файл.
- metadata – завантажити лише службову інформацію, таку як назва, тривалість аудіо файлу та ін.
- auto – завантажити аудіо файл при завантаженні веб-сторінки.
- src – вказує шлях до файлу, що відтворюється, або адрес джерела аудіо потоку. Для цієї ж цілі також можна використовувати тег <source> усередині тегу <audio>. В якості значення приймається повний або відносний шлях до файлу, або до джерела аудіо потоку [7].

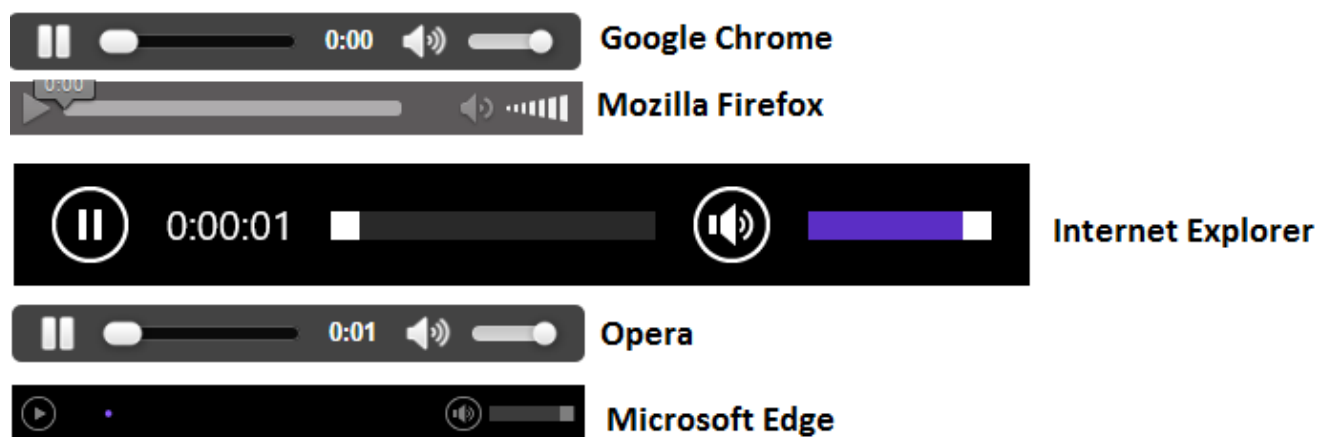


Рисунок 5 - Порівняння вигляду тегу <audio> у різних браузерах

### Сайти створенні за допомогою CMS

При необхідності підключення до сайтів, що створенні за допомогою систем керування вмістом у Інтернеті можна знайти безліч модулів, що працюють з різними варіантами медіа серверів.

У безкоштовних модулів зазвичай наявний мінімум налаштувань та можливостей. В них можна задати тип медіа серверу, та параметри необхідні для підключення до нього. Модулі зазвичай реалізовані на технології Flash, так як розробка модуля буде займати менше часу.



Платні ж модулі мають той самий функціонал, що й безкоштовні, проте зазвичай наявні додаткові функції. Наприклад, вони можуть приймати та відображати метайнформацію, що надходить з медіа серверу. Також серед платних модулів можна знайти модулі реалізовані на HTML 5, а не на технології Flash компанії Adobe. Використання розширень на HTML 5 дає нам можливість доступу до плеєра на сайті не тільки з десктопних браузерів, а й з мобільних пристроїв, адже мобільні браузери не підтримують Flash, а HTML 5 у найближчому повністю витисне свого конкурента [8].

## 1.4 Висновки

Незважаючи на велику кількість протоколів, створених для багатоадресної передачі потокового медіа контенту, радіомовлення через мережу використовує звичайний TCP для встановлення з'єднання між клієнтом та сервером та надійної передачі даних.

Основними компонентами Інтернет радіо є:

- Радіостанція, що створює потік та відправляє його на медіа сервер.
- Медіа сервер, що приймає потік від радіостанції та відправляє усім підключеним клієнтам.
- Програвач клієнта, що підключається до медіа серверу, отримує та відтворює аудіо потік.

Проаналізувавши методи створення аудіо файлів та визначивши їх переваги і недоліки можна сказати, що для Інтернет радіомовлення найбільш підходящими є формати із стисненням та втратами, такі як MP3, AAC, OGG Vorbis.

## **2 АНАЛІЗ І ПОРІВНЯЛЬНА ХАРАКТЕРИСТИКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ІНТЕРНЕТ РАДІО**

Для кожного з компонентів Інтернет радіо використовується власне програмне забезпечення.

### **2.1 Програмне забезпечення для створення аудіо потоку**

Першим, що необхідно будь-якій радіостанції у мережі Інтернет це програмне забезпечення, що зможе віддавати свій аудіо потік у мережу. Існує багато програм для рішення даної задачі. Найбільш розповсюдженими є:

- Для комп'ютерів під управлінням ОС Windows:
  - RadioBOSS
  - Winamp + плагін до ShoutCast(програмне забезпечення медіа серверу)
- Для комп'ютерів під управлінням ОС Linux:
  - Ices
  - PulseAudio

Розглянемо спочатку програми для комп'ютерів під управлінням ОС Windows. RadioBOSS - це платна програма для автоматизації ефіру радіостанцій. Широкі можливості програми дозволяють використовувати її скрізь, де потрібно музичне оформлення. Програма може працювати місяцями без втручання людини. Простий і зручний інтерфейс дозволить почати роботу відразу ж після запуску програми. RadioBOSS включає:

- Стабільний плеєр, який контролює коли і які плейлисти відтворюються.
- Конвертер аудіо файлів, що дозволяє використовувати аудіо файли, які є на вашому диску.

- Музичну базу для пошуку, каталогізації, коментування та управління вашими аудіо файлами.
- Надійний генератор плейлистів з професійними функціями, що включає шаблони плейлиста і ін.
- Рекламний планувальник для легкого створення і планування рекламних блоків.
- Генератор звітів для створення докладних звітів про відтворені треках.

Другим варіантом для ОС від Microsoft є Winamp, хоча для мовлення треба встановити спеціальний плагін для SHOUTCast. Основною перевагою Winamp є те, що він безкоштовний. Також він має достатньо високу функціональність, а саме:

- Велика кількість форматів файлів, що можуть відтворюватися: MP3, ACC, ACC+, WMA, MIDI, AMF, WMV, ASF та інші.
- Підтримка модулів, що дозволяють розширити функціональність; відкритий формат модулів, що дозволяє створювати власні модулі.
- Підтримка безперервного відтворення для MP3 та AAC.
- Широкопasmовий еквалайзер, що має декілька десятків налаштувань для всіх жанрів музики.

Проте серед комп'ютерів, що використовуються для роботи, дуже часто використовуються Unix-подібні операційні системи, через їхню доступність та більшу ефективність.

Ices – це безкоштовне програмне забезпечення, що створене спеціально для програми IceCast (ПЗ для медіа серверу). IceCast можна використовувати на платформах:

- Linux (включаючи Redhat та Debian)
- FreeBSD
- OpenBSD

- Solaris

Також його великою перевагою є мала необхідність у ресурсах комп'ютера. На відміну від попередньо описаних аналогів, ця програма не може програвати MP3 та інші формати файлів, проте вона може повністю зчитувати усе, що проходить через звукову карту комп'ютера, включаючи сигнали з мікрофонів та інших програвачів.

Останнім у списку є PulseAudio. Це кросплатформений звуковий сервер, що може бути встановлений як на POSIX платформах, таких, як Linux, Solaris, FreeBSD, так і на платформі Microsoft Windows. Ця програма є абсолютно безкоштовною. Основними можливостями даного ПЗ є:

- Роздільна установка рівнів гучності для різних програм [5]
- Можливість роботи з декількома джерелами і приймачами звуку
- Мала затримка і можливість її точного вимірювання [6]
- Ефективний мікшер, навіть для звукових плат без мікшера, що запобігає «монополізації»/«крадіжці» звукового пристрою окремою програмою
- Вбудоване перетворення аудіо форматів
- Синхронізація потоків відтворення
- Модуль придушення акустичного відлуння для VoIP

Проаналізувавши варіанти програмного забезпечення для створення аудіо потоку, можна зробити висновки, що платний продукт має більше додаткових функцій, проте основні функції реалізовано скрізь у повній мірі. Для студентського радіо найбільш підходящим буде PulseAudio, оскільки він є безкоштовним та має достатню кількість налаштувань аби створювати якісне мовлення. А його можливість роботи з декількома джерелами звуку дозволяє вести ефір діджеєм з різних місць, зв'язавшись при цьому по Viber або Skype.

## 2.2 Програмне забезпечення серверів для Інтернет радіо

Найважливішою частиною Інтернет радіомовлення є медіа сервер – програмне забезпечення, що до якого під’єднуються усі потоки аудіо даних, як вхідні, так і вихідні. Існують два найпопулярніших медіа сервера для радіомовлення: SHOUTcast та IceCast 2. Обидва з них є безкоштовними та кросплатформними.

### Функціональність

Основною функціональністю, якою має володіти медіа сервер це здатність приймати потоки аудіо даних від станції радіомовлення та передавати їх до всіх клієнтів, що підключені до цього серверу. Звичайно, що обидва з варіантів можуть це робити. Як SHOUTcast так і IceCast 2 можуть отримувати метадані від станції та передавати їх кінцевим користувачам. Обидва варіанти дають можливість віддаленого доступу, за допомогою Інтернет браузерів (рис. 6 та 7).

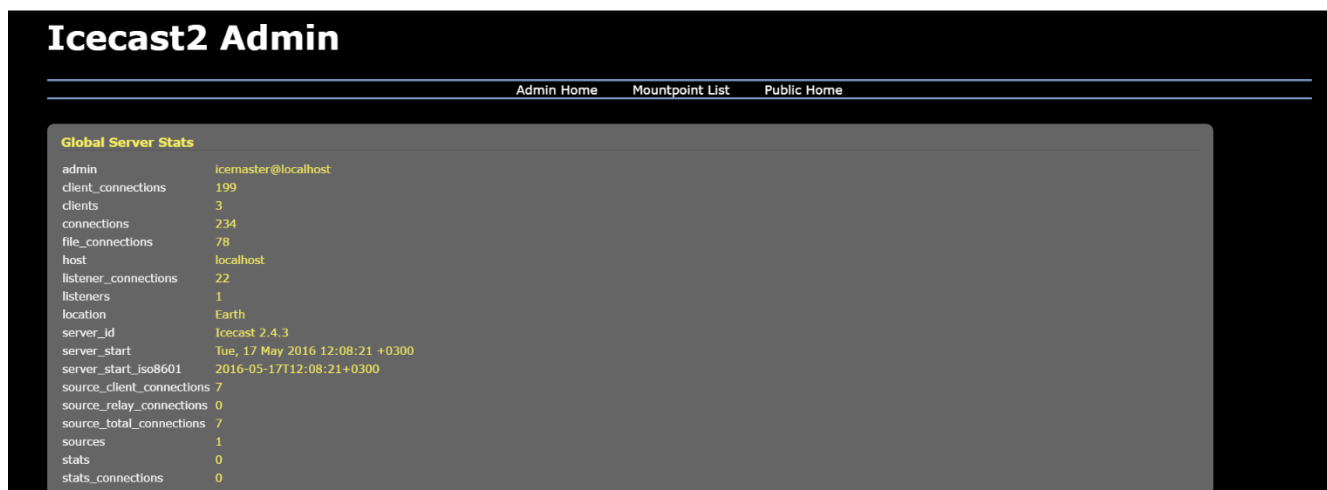


Рисунок 6 - Адмін палень IceCast

У додачу до веб-інтерфейсу SHOUTcast (у версії для ОС Windows) має графічний інтерфейс користувача що відкривається при запуску виконуваного файлу (рис. 8).

## SHOUTcast Server Summary

Uptime: 43 minutes 23 seconds

SHOUTcast Server v2.4.7.256/win32

[Help](#) | [Documentation](#) | [Bandwidth Usage](#) | [Log \(Tailing | Save\)](#) | [Summary](#) | [Ban List](#) | [Reserved List](#) | [User Agent List](#) | [Logout](#)

Available Streams: 1 | Server Listeners: 0 | Peak Server Listeners: 0 | Unique Listeners: 0

**Server Management**

Rotate Log File(s): [ All ] [ Log ] [ W3C ]

Log file: sc\_serv.log

Configuration file: sc\_serv\_basic.conf

---

View Source Connection Details: [ View ]

View Advert Group Details: [ View ]

**Stream #1 (Stream Login)**

Create Authhash | Manually Set Authhash

SKPI (MP3 @ 64 kbps)

Playing: **Hardwell & GTA & Trendsetter - Wild Animals (Bass Remix) - Hardwell & GTA & Trendsetter - Wild Animals (Bass Remix)**

Listeners: 0 of 512 | Status: Not Listed - Empty Authhash | Source: v1 [ kick ]

[Summary](#) | [Listeners](#) | [Played History](#) | [Metadata](#) | [Statistics ...](#)

Рисунок 7 - Адмін панель SHOUTcast

Nullsoft SHOUTCast Server 2 Win32 GUI

DONATE

Kill Server | Restart Server | Diagnose | ViewLog | Edit Config

```

24.05.2016 16:35:11 INFO Your Shoutcast sc_serv port is :8001
24.05.2016 16:35:11 INFO Server starting ...
2016-05-24 16:35:11 INFO *****
2016-05-24 16:35:11 INFO ** SHOUTcast Distributed Network Audio Server (DNAS) **
2016-05-24 16:35:11 INFO ** Copyright (C) 2014-2015 Radionomy SA, All Rights Reserved **
2016-05-24 16:35:11 INFO *****
2016-05-24 16:35:11 INFO [MAIN] SHOUTcast DNAS/win32 v2.4.7.256 (Mar 31 2015)
2016-05-24 16:35:11 INFO [MAIN] PID: 2948
2016-05-24 16:35:11 INFO [MAIN] Saving log output to 'C:\RADIO\SHOUTcastGUMlogs\sc_serv.log'
2016-05-24 16:35:11 INFO [MAIN] Automatic log rotation interval: 1 day
2016-05-24 16:35:11 INFO [MAIN] Loaded config from 'C:\RADIO\SHOUTcastGUMsc_serv_basic.conf'
2016-05-24 16:35:11 INFO [MAIN] Calculated CPU count is 4 --> using all available CPUs
2016-05-24 16:35:11 INFO [MAIN] Starting 4 network threads
2016-05-24 16:35:11 INFO [MICROSERVER] Listening for source and client connections on port 8001
2016-05-24 16:35:11 INFO [MICROSERVER] Listening for legacy source connections on port 8002
2016-05-24 16:35:11 INFO [MICROSERVER] Flash policy file server not enabled
24.05.2016 16:35:26 INFO Detected internal ip : 127.0.0.1 & External IP : 62.205.135.116
24.05.2016 16:35:26 INFO Loading a new settings.xml!
2016-05-24 16:36:20 INFO [SRC 127.0.0.1] SHOUTcast 1 source connection starting.
2016-05-24 16:36:20 ERROR [SRC 127.0.0.1] SHOUTcast 1 source connection denied. Bad password: pass
2016-05-24 16:36:35 INFO [SRC 127.0.0.1] SHOUTcast 1 source connection starting.
2016-05-24 16:36:35 ERROR [SRC 127.0.0.1] SHOUTcast 1 source connection denied. Bad password: pass
2016-05-24 16:36:50 INFO [SRC 127.0.0.1] SHOUTcast 1 source connection starting.
2016-05-24 16:36:50 ERROR [SRC 127.0.0.1] SHOUTcast 1 source connection denied. Bad password: pass
2016-05-24 16:37:05 INFO [SRC 127.0.0.1] SHOUTcast 1 source connection starting.
2016-05-24 16:37:05 ERROR [SRC 127.0.0.1] SHOUTcast 1 source connection denied. Bad password: pass
2016-05-24 16:37:20 INFO [SRC 127.0.0.1] SHOUTcast 1 source connection starting.
2016-05-24 16:37:20 ERROR [SRC 127.0.0.1] SHOUTcast 1 source connection denied. Bad password: pass
2016-05-24 16:37:35 INFO [SRC 127.0.0.1] SHOUTcast 1 source connection starting.
2016-05-24 16:37:35 ERROR [SRC 127.0.0.1] SHOUTcast 1 source connection denied. Bad password: pass
  
```

Status: Not Registered to YP | No external connectivity | Streams:1 | Running for 0 Days, 00:02:34.18 | About

Рисунок 8 - Графічний інтерфейс користувача SHOUTcast

Проте, якщо в SHOUTcast сервері можна додавати деякі IP адреси, або цілі підмережі, вказуючи маску підмережі, до списку заблокованих прямо із сайту, то у його конкурента така можливість відсутня, проте їх можна прописати безпосередньо до файлу, що лежить на сервері (шлях до файлу зі списком

заблокованих адрес необхідно вказувати у конфігураційному файлі “icecast.xml”).

```
<paths>
  <basedir>./</basedir>
  <logdir>./logs</logdir>
  <pidfile>./icecast.pid</pidfile>
  <webroot>./web</webroot>
  <adminroot>./admin</adminroot>
  <allow-ip>/path/to/ip_allowlist</allow-ip>
  <deny-ip>/path_to_ip_denylist</deny-ip>
  <alias source="/foo" dest="/bar"/>
</paths>
```

Рисунок 9 - Фрагмент коду конфігураційного файлу IceCast [18]

Обидва програмних забезпечення можуть показувати інформацію про споживання трафіку, кількість підключених користувачів, та видавати більш детальну інформацію з цього приводу (IceCast видає її в більш придатному для користувача форматі, SHOUTcast видає у вигляді відкритого XML файлу) (рис. 10).

```
<SHOUTCASTSERVER> SHOUTcast
  <LISTENERS>
    <LISTENER>
      <HOSTNAME>192.168.0.177</HOSTNAME>
      <USERAGENT>
        Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36
        (KHTML, like Gecko) Chrome/50.0.2661.102 Safari/537.36
      </USERAGENT>
      <CONNECTTIME>9</CONNECTTIME>
      <UID>7</UID>
      <TYPE>1024</TYPE>
      <REFERER/>
      <XFF/>
      <GRID>0</GRID>
      <TRIGGERS>0</TRIGGERS>
    </LISTENER>
  </LISTENERS>
</SHOUTCASTSERVER>
```

IP	Sec. connected	User Agent	Action
192.168.0.177	14	Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/50.0.2661.102 Safari/537.36	<a href="#">Kick</a>

Рисунок 10 - Порівняння інформації про користувача, що повертає графічний вивід SHOUTcast(зліва) та IceCast(справа) серверів.

Також корисною буде функція відключення користувача від серверу (рис. 9 справа, жовта клавіша «Kick»), або станції, що передає інформацію на сервер (рис. 10). У IceCast є можливість відключати кожного клієнта окрема, у той час, як його суперник надає лише можливість відключення усіх користувачів разом. Обидва мають можливість відключення потоку, що йде від станції.

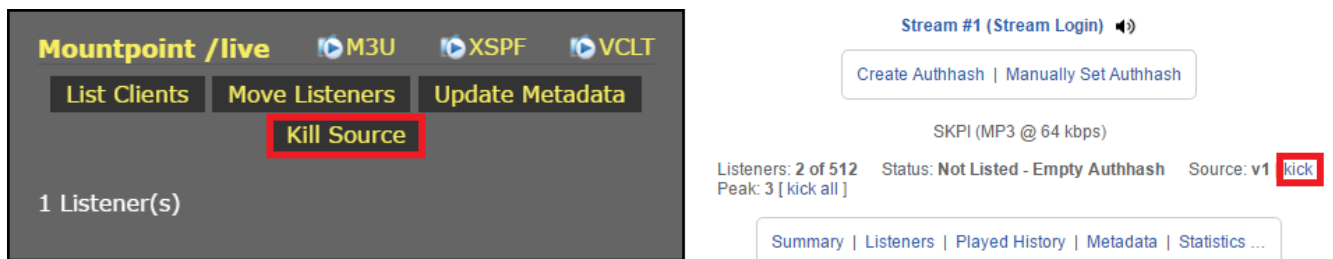


Рисунок 11 - Відключення потоків, що йдуть від радіо станцій, IceCast(зліва), SHOUTcast(справа)

У питанні потоків IceCast надає можливість підключення декількох потоків («маунтпоінтів»), та роботу з ними. Від так можна перенести всіх користувачів одного потоку на інший, так що користувач цього навіть не помітить, продивитись користувачів по кожному з потоків (рис. 12). SHOUTcast може працювати лише з фіксованою, заданою кількістю потоків одночасно, тобто якщо дозволено один потік і підключилася одна станція до серверу, то усі інші будуть отримувати помилку підключення, при спробі розпочати відправлення потоку за заданим адресом.



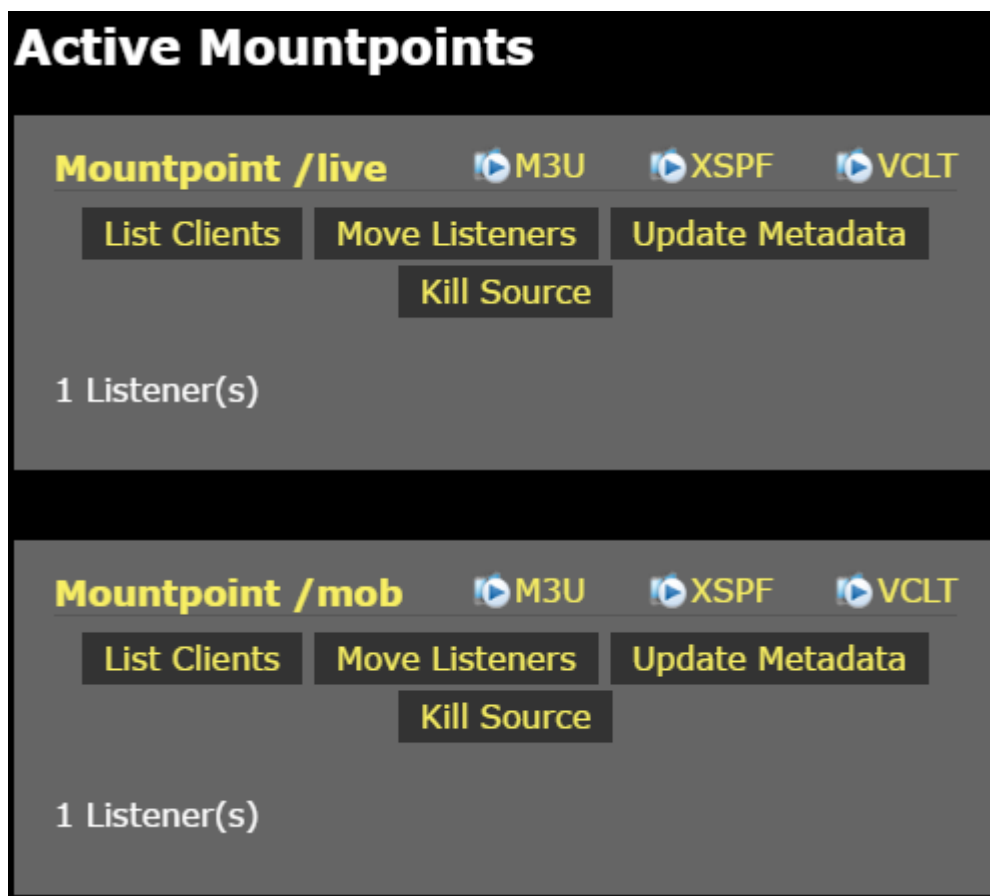


Рисунок 12 - Демонстрація роботи декількох вхідних потоків на IceCast сервері.

IceCast сервер дозволяє завантажити файли (рис. 12 підписані, як M3U, XSPF, VCLT), котрі можна відкрити програвачем, аби він (програвач) налаштувався на прослуховування аудіо даних, що йдуть з цього медіа серверу.

Тож бачимо що за функціональністю інтерфейсу користувача обидва продукти майже однакові, але можливість динамічного збільшення радіостанцій підключених до одного медіа серверу надає перевагу IceCast.

## Технічні характеристики

Однією з найважливіших характеристик для програмного забезпечення є платформа або операційна система, під яку його створено. Обидва продукти, що розглядаються є кросплатформними та мають відповідні пакети установки для UNIX-подібних систем та комп'ютерів, що користуються системою Windows (рис. 13 і 14).



Рисунок 13 - Сторінка завантаження IceCast [18]

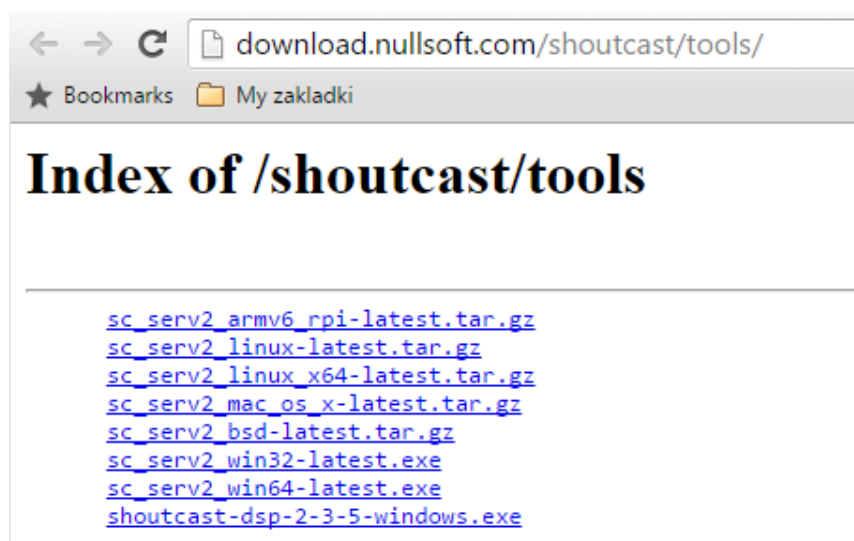


Рисунок 14 - Сторінка завантаження SHOUTcast [21]

Також важливим питанням є простота та час необхідний для налаштування серверу. Для налаштування IceCast використовується файл «icecast.xml», що знаходиться в корневому каталозі IceCast. Для запуску серверу необхідно лише задати hostname, за замовчанням «localhost», та порт, за замовчанням «8000». Також можна задати ім'я та пароль адміністратора, проте всі значення із самого

початку встановленні у стандартні. Тут зберігається пароль для потоків від радіостанцій. Також у цьому файлі можна встановити такі параметри, як: ліміт клієнтів, максимальний час очікування відгуку від клієнта, максимальний час очікування відгуку від радіостанції, заблоковані IP адреси та ін.

Для налаштування SHOUTcast використовується файл конфігурацій «sc\_serv\_basic.conf» де можна встановити лише адрес серверу, номер порту та паролі для джерел та адміністратора. Надалі настройки серверу можна проводити за допомогою графічного інтерфейсу користувача.

Не менш важливим показником є можливість передачі файлів різних форматів. Обидва медіа сервери можуть пересилати не тільки великий набір аудіо форматів, таких як: AAC, AAC+, MP3, WAV, WMA, NSV тощо; а й відео потоки.

Для систем з малою ресурсозабезпеченістю, такою як оперативна пам'ять, потужність процесора та ін., велику роль відіграє необхідність даного програмного засобу у ресурсах. Під час використання SHOUTcast разом із програмою створення аудіо потоку середня завантаженість ЦП доходила до 25,9%, проте спостерігалися постійні стрибки завантаженості до 100%. IceCast разом із програмою створення аудіо потоку Ices (програма спеціально створення для роботи з IceCast, тим самим розробником Xiph.Org Foundation) завантажували процесор максимум на 2.8% від максимально допустимого значення. Отже можна зробити висновок, що для комп'ютерів, що не мають великої ресурсозабезпеченості, IceCast є найкращим варіантом для реалізації свого медіа серверу [9].

### **Документація, підтримка, плагіни**

На сайті IceCast [18] наявна велика документація для кожної версії продукту, проте вона наявна лише англійською мовою (рис. 15). У ній можна знайти інструкції по базовому налаштуванню медіа сервера, інформацію про роботу з адміністративною панеллю, про роботу з файлом конфігурації та багато

іншого. Також на сайті можна продивитися перелік продуктів, за допомогою яких можна надсилати аудіо потік на сервер.

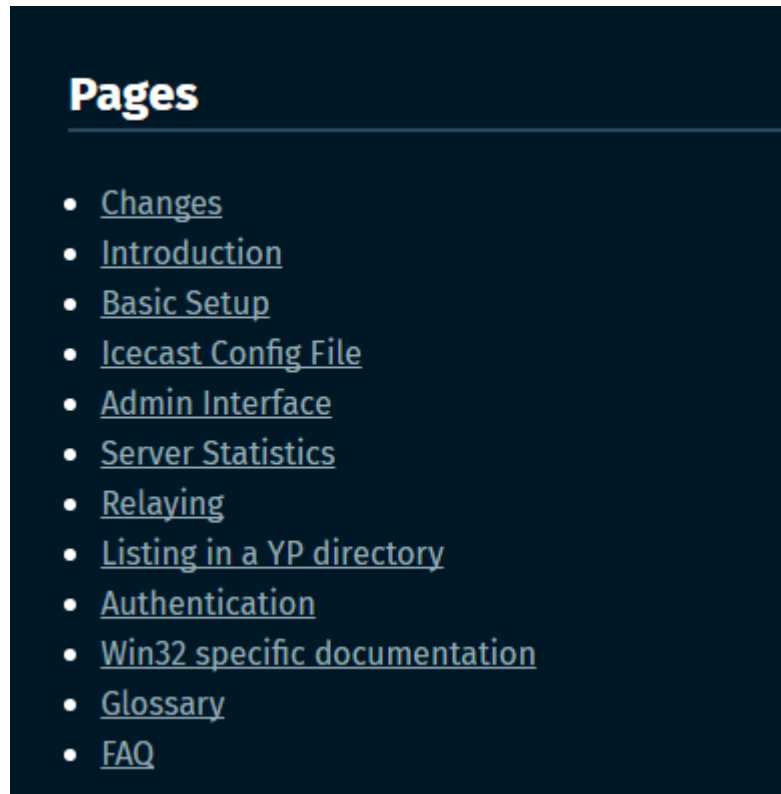


Рисунок 15 - Документація Icescast [18]

Для SHOUTcast офіційну документацію знайти можна на сайті SHOUTcast [19]. Також документація поставляється разом з продуктом і її можна знайти у папці «doc» усередині папки встановлення SHOUTcast (рис. 16).

<p style="text-align: center;">Contents [hide]</p> <ul style="list-style-type: none"> <li>1 Introduction</li> <li>2 Overview</li> <li>3 Getting Started <ul style="list-style-type: none"> <li>3.1 Running the Server</li> <li>3.2 Windows <ul style="list-style-type: none"> <li>3.2.1 Install as a Service</li> <li>3.2.2 Uninstall the Service</li> <li>3.2.3 Run in the Console</li> </ul> </li> <li>3.3 Linux / BSD / Raspbian / Mac OS X <ul style="list-style-type: none"> <li>3.3.1 Run as a Daemon</li> <li>3.3.2 End a Daemon</li> <li>3.3.3 Run as a Non-Daemon</li> <li>3.3.4 Immediate Segfault on Run</li> </ul> </li> <li>3.4 Run in Setup Mode</li> <li>3.5 Additional Signals</li> </ul> </li> <li>4 Configuration File <ul style="list-style-type: none"> <li>4.1 Banning</li> <li>4.2 Listener Behaviour</li> <li>4.3 Debugging</li> <li>4.4 Flash Policy Server</li> <li>4.5 Introduction and Backup Files</li> <li>4.6 Logging</li> <li>4.7 Miscellaneous</li> <li>4.8 Networking</li> <li>4.9 Network Buffers</li> <li>4.10 Relaying</li> <li>4.11 Reserved List</li> <li>4.12 Stream Configuration</li> <li>4.13 Web Connection (W3C) Logging</li> <li>4.14 YP Server Behaviour</li> <li>4.15 YP Server Errors</li> <li>4.16 Statistics</li> <li>4.17 Blocked User Agents</li> <li>4.18 Artwork (Experimental)</li> </ul> </li> <li>5 Administration <ul style="list-style-type: none"> <li>5.1 Administration Pages <ul style="list-style-type: none"> <li>5.1.1 Public Pages</li> <li>5.1.2 Private Pages <ul style="list-style-type: none"> <li>5.1.2.1 Configuration Reload</li> </ul> </li> </ul> </li> <li>5.2 XML / JSON / JSON-P Responses</li> </ul> </li> <li>6 Stream Addresses <ul style="list-style-type: none"> <li>6.1 HTTP Protocol Compatibility</li> </ul> </li> <li>7 Maximum Listener Connection Limits</li> <li>8 Example Configurations <ul style="list-style-type: none"> <li>8.1 sc_serv_basic</li> <li>8.2 sc_serv_public</li> <li>8.3 sc_serv_relay</li> <li>8.4 sc_serv_simple</li> </ul> </li> </ul>	<h2>1. Introduction</h2> <hr/> <p>The purpose of this document is to show you the different configuration options supported by the DNAS server (<code>sc_serv</code>) along with basic and more advanced example configurations to get started with using the DNAS server and the many features it can offer.</p> <p>The aim of the DNAS server is to provide enhanced serving features and also access to the new YP2 infrastructure whilst maintaining as much backward compatibility with previous versions of DNAS server as possible. The new features introduced include:</p> <ul style="list-style-type: none"> <li>▪ Serving multiple streams from a single server instance</li> <li>▪ Relaying multiple streams from a single server instance</li> <li>▪ Multiplexing all server activity through a single IP port</li> <li>▪ SHOUTcast 2 wire protocol support for sources, relays and listener clients</li> <li>▪ Repackaging of SHOUTcast 1.x and SHOUTcast 2.x stream data as needed for listener clients</li> <li>▪ Ability to use any SHOUTcast compatible source (1.x or 2.x) for any stream</li> <li>▪ v2 SHOUTcast Directory (YP2) infrastructure support</li> <li>▪ Real-time metadata and statistic reporting</li> <li>▪ Static station id support</li> <li>▪ Expanded in-stream metadata support including sending of artwork</li> <li>▪ UTF-8 and international character encoding</li> <li>▪ Improved server and stream security</li> <li>▪ Improved server support for using DNS addresses instead of raw IP</li> <li>▪ Backup url support to keep streams active when the main source fails</li> <li>▪ Improved support for intro and backup files over the 1.x support</li> </ul> <h2>2. Overview <a href="#">[top]</a></h2> <hr/> <p>To take full advantage of the features provided by the 2.x DNAS server, it is recommended to use a 2.x compatible source which can provide the required data.</p> <p>However this does not prevent you from using a 1.x compatible source with the 2.x DNAS, though you will not get the full benefit of all of the additional platform and DNAS features with the 2.x DNAS server.</p> <p>With the introduction of the v2.4.7 DNAS server, you are now also able to connect a 1.x source for any configured stream as long as you enter the correct password for the stream. See <a href="#">Server Source Support</a> for more information about this and the history of this support.</p> <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;"> <p><b>If a listener connection is detected as 'SHOUTcast compatible', the DNAS will repackage the stream and any metadata into the 2.x or 1.x SHOUTcast streaming format as applicable.</b></p> </div>
---	---

Рисунок 16 - вбудована документація SHOUTcast

Документація для обох серверів наявна лише на англійській мові.

Плагіни для інтернет радіо мовлення, чи то плагін для створення потоків на радіостанції, чи то плагін для додаткових можливостей медіа серверу, існують у великій кількості, як для IceCast, так і для SHOUTcast. Деякі плагіни, наприклад SAM Encoders DSP Plugin for Winamp 3.8.3 – плагін, що створює аудіо потік і відправляє його за заданою адресою, може працювати як із Icecast, так і з SHOUTcast медіа серверами.

## Можливості платних серверів

Не зважаючи на існування якісних безкоштовних серверів, існують сервіси [13,14], що дають змогу використання медіа серверів за платню. Замовивши на подібних сайтах медіа сервер можна отримати:

- Детальна статистика:
  - Кількість слухачів по місяцям, дням та годинам.
  - Трафік як загальний, так і по кожному слухачу окремо.
  - Середній час прослуховування.
  - Програми, якими користуються слухачі.
  - Географічні координати слухачів.
  - Ім'я, кількість та час звучання треків у ефірі.
- Сумісна робота декількох діджеїв. Можливо створити декілька діджеїв із різними пріоритетами, і при одночасній роботі декількох з них, виводитись буде лише потік, що йде від діджея з найвищим пріоритетом. При підключенні працівника з більш високим пріоритетом, потік плавно переключиться на нього.
- Можливість роботи із серверами SHOUTcast та IceCast.
- Наявні API для створення власних модулів, що будуть працювати із сервером.
- Одночасна трансляція одного й того ж потоку у різній якості. Це дасть можливість слухачеві обрати ту якість, яку дозволяє швидкість їх з'єднання і Інтернетом. Оскільки потоки синхронізовані, то можна переключатись між ними майже непомітно.

Отже платні сервіси дають додаткові функції для покращення якості роботи із медіа сервером як адміністратора, так і слухача. Детальна інформація може бути цікава рекламодавцям. Тобто цей вид реалізації потрібен комерційним користувачам, яким необхідний даний функціонал але вони не хочуть створювати його самостійно.

## **Висновки**

Отже проаналізувавши варіанти реалізацій медіа серверів можна сказати, що досить великих відмінностей у плані функціоналу та підтримки немає. Проте мала необхідність у ресурсах серверу IceCast робить його більш вигідним варіантом для використання. Ще однією перевагою є те, що даний сервер створений тією ж компанією, що й кодек OGG Vorbis, тому він підтримує цей поки що не досить широко використовуваний, але дуже якісний, формат. Платні сервери мають додаткові функції, що робить цей вид реалізації більш підходящим для комерційних користувачів.

## **2.3 Програмне забезпечення клієнтів для Інтернет радіо**

Існує багато програвачів, які можна встановити на ПК, що працюють під керування різних операційних систем. Проте не всі програвачі можуть відтворювати потік, що йде з мережі. Найбільш розповсюдженими є:

- Winamp – безкоштовний плеєр, що має реалізацію для ОС Windows, Mac OS, та Android. Плеєр відрізняється підтримкою великої кількості форматів та наявністю великої кількості плагінів, що дозволяють налаштувати функціонал плеєру під необхідності кожного користувача. Еквалайзер плеєру має широкий спектр налаштувань для різних видів музики, що покращує якість відтворення музики.
- AIMP – плеєр з підтримкою всіх відомих форматів аудіо. Має пакети для встановлення на ОС Windows та Android. Має у наявності велику кількість модулів. Дозволяє обробляти звук у якості 32 біти. Дозволяє як прослуховувати радіостанції, так і створювати власний аудіо потік для відправки на медіа сервер.
- Amarok – найкращий плеєр для ОС Linux, хоча має реалізації для ОС Windows та Mac OS. Відмінністю даного програвача є тісна

інтеграція з Вікіпедією [17], що дозволяє швидко знаходити детальну інформацію про виконавця та треки, що програвуться.

Проаналізувавши ПЗ, що дає можливість прослуховування онлайн радіомовлення, можна зробити висновок, що програвачі своїм основним функціоналом майже не відрізняються, а до вибору того чи іншого програмного продукту спонукає лише вподобання до інтерфейсу або до деяких додаткових функцій. Для ПК під керуванням ОС Linux слід обрати Amarok, як найбільш якісний програвач для цієї операційної системи.

## 2.4 Висновки

Отже ПЗ медіа серверів основну функцію виконують однаково, проте через низьку необхідність у ресурсах кращим вибором є IceCast. Серед програм, що можуть створювати аудіо потік можна виділити PulseAudio, що має такі особливості:

- Можливість роботи з декількома джерелами і приймачами звуку.
- Синхронізація потоків відтворення.
- Модуль придушення акустичного відлуння для VoIP.
- Кросплатформеність.

Усі існуючі програвачі, що можуть відтворювати аудіо потоки, які йдуть з мережі, за своїм функціоналом є приблизно однаковими.



## **3 АНАЛІЗ ВИМОГ ДО СЕРВІСУ І ПЛАН РОБІТ ДЛЯ ЙОГО РЕАЛІЗАЦІЇ**

Змістом дипломної роботи є організація трансляції студентського радіо КПІ або інших передач у 26 корпусі. Для цього необхідно встановити медіа сервер на сервер кафедри та організувати доступ до нього, також необхідно організувати можливість звернення користувачів до сервісу через сайт кафедри.

### **3.1 Вимоги до сервісу**

Даний пункт диплому описує вимоги, що ставляться до можливостей медіа серверу Інтернет радіо та зв'язок із ним. Кінцевий результат повинен відповідати усім підпунктам описаним в даному розділі.

Програмне забезпечення повинно мати можливість встановлення на ЕОМ під керівництвом ОС Linux. На сайт кафедри необхідно додати модуль, що зможе по запиті користувача підключатись до серверу, та відтворювати аудіо потік, що йде від серверу. Модуль повинен бути сумісним із CMS Joomla! Версії 1.5.25. Сервер повинен надійно працювати при одночасному підключенні  $\leq 100$  слухачів.

Результатом роботи повинен бути встановлений на фізичний сервер кафедри та налаштований медіа сервер. Також на сайт кафедри повинен бути встановлений та налаштований модуль для підключення до серверу та відтворення аудіо потоку.

### **3.2 План робіт для реалізації сервісу**

Для реалізації сервісу необхідно виконати роботи за наступним планом:

1. Спираючись на аналіз проведений у підрозділі 2.2 обрати найкращий варіант медіа серверу.
2. Завантажити пакет для встановлення на ЕОМ під керування ОС Linux.

3. Встановити пакет та створити службу, яка буде вмикати та вимикати дане ПЗ.
4. Налаштувати конфігураційний файл, в якому обов'язково треба задати номер порту, на котрому буде працювати служба, логін та пароль адміністратора.
5. Знайти та завантажити модуль для Joomla! версії 1.5.25
6. Встановити модуль на сайт кафедри СП.
7. Налаштувати модуль та відобразити його на сайті кафедри СП.
8. Протестувати роботу сервера та модуля.

### **3.3 Висновки**

У даному розділі було проаналізовано вимоги до сервісу:

- Медіа сервер повинен мати можливість встановлення на ОС Linux.
- Модуль повинен бути сумісним із Joomla! Версії 1.5.25.
- Сервер повинен надійно функціонувати при одночасному підключенні  $\leq 100$  слухачів.

Було розроблено план реалізації сервісу:

- Завантажити та встановити медіа сервер.
- Налаштувати конфігураційний файл медіа серверу.
- Завантажити та встановити радіо модуль на сайт кафедри.
- Налаштувати модуль на встановлений медіа сервер.

## 4 РЕАЛІЗАЦІЯ СЕРВІСУ ІНТЕРНЕТ РАДІО В МЕРЕЖІ КАФЕДРИ СП

Даний параграф описує роботи, що були проведенні для реалізації вимог, що поставлені до сервісу.

### 4.1 Встановлення та налаштування медіа серверу

Спираючись на порівняльну аналіз безкоштовних медіа серверів для Інтернет радіомовлення, що був проведений у підрозділі 2.2, можна сказати, що сервери за набором своїх можливостей майже не відрізняються. Проте мала необхідність в ресурсах серверу IceCast робить його найкращим варіантом для встановлення на сервер кафедри.

Для завантаження встановлювального пакету необхідно у командний рядок ввести команду: «`wget http://downloads.xiph.org/releases/icecast/icecast-2.4.3.tar.gz`».

Надалі аби дістати усі файли з архіву необхідно ввести команду: «`tar -xvzf icecast-2.4.3.tar.gz`». Буде створена директорія з іменем ідентичним до назви пакету.

Зайшовши до цієї директорії необхідно виконати наступні команди: «`make`», «`make install`»

Після проведених дій сервіс під назвою `icecast` буде успішно встановлено.

Конфігураційний файл лежить у папці «`etc/icecast`» під назвою «`icecast.xml`». Необхідно ввести дані у такі теги:

- У тег `<clients>` треба ввести обмеження на максимальну кількість слухачів 100.
- У тег `<port>` необхідно ввести номер порту на якому буде працювати сервер і до якого буде підключатись клієнт 8000.

- У тег <source-password> необхідно ввести пароль для джерел (радіостанцій)
- У тег <admin-user> необхідно ввести логін адміністратора.
- У тег <admin-password> необхідно ввести пароль адміністратора.

Після внесення змін конфігураційного файлу необхідно запустити службу. Для цього треба ввести команду «service icecast start».

## 4.2 Підключення сервісу до сайту

Оскільки на кафедрі СП використовується сайт створений за допомогою CMS Joomla!, то є можливість встановити спеціальний модуль для прослуховування Інтернет радіостанцій. Встановлення модуля дає додаткові можливості при прослуховуванні ефіру, такі як кешування аудіо потоку, що програвється, для подальшої можливості відмотування назад. Надалі розглядатиметься встановлення модуля для Інтернет радіомовлення для CMS Joomla! Версії 1.5.25, оскільки саме така версія системи керування вмістом використовується на сайті кафедри.

Знайти існуючі модулі можна на сайті розширень Joomla! [16].

Обравши модуль, що підходить для заданої версії Joomla! Треба його скачати, зазвичай він постачається у вигляді ZIP архіву. У випадку нашої версії Joomla! та необхідності використання безкоштовного програмного забезпечення, було обрано модуль «Shoutcast Radio Player».

**Dimedia** Radio Player for Joomla! Home About **Download** FAQ PRO Contact / Support

**Try Premium!**  
Upgrade your Free Player and show up the current track title (click the play button):

**Buy Now - \$1.80 or 1.50€**

Need a player that works with Android and iOS devices? Try our HTML5 based radio player V2:

**Buy Now - \$2.00 or 1.75€**

## Download

### Latest version

Here you can download our latest version of the Radio Player Module for Joomla!

**FREE**

Version: 2.6  
Released: 22.10.2013  
Release log: Code cleaning

**Download V2.6 FREE** →

Based on [Muses Radio Player](#)  
More skins can be found [here](#) (Open ticket for instructions).

Download Radio Player 2.5 for Joomla! 1.5 **DOWNLOAD**

Рисунок 17 - Сторінка завантаження модуля Joomla!(червоним виділено кнопку завантаження модуля для Joomla! 1.5) [20]

Надалі за допомогою менеджера розширень необхідно встановити модуль. Для цього на сторінці «[http://{ServerDomain}/administrator/index.php?option=com\\_installer](http://{ServerDomain}/administrator/index.php?option=com_installer)», де ServerDomain – це адреса сайту у мережі інтернет, або його IP адреса, у пункті «Завантажити файл пакета» обрати архів модуля та натиснути клавішу «Завантажити файл та встановити».

Після отримання повідомлення про успішне встановлення модуля, необхідно перейти у вкладку «Розширення->Керування модулями». Тут треба знайти модуль, який щойно був встановлений, в нашому випадку це «Radio Player Joomla FREE», та натиснути на його назву.

Потрапивши до панелі налаштувань модуля Joomla!, необхідно задати розташування модуля на сторінці (на рис. 18 «left»), сторінки на яких даний модуль буде відображатись (на рис. 18 «Всі»), також у пункті включено необхідно поставити маркер на варіант «Так», аби наш модуль відображався на сайті, ще можна виключити відображення назви модуля.

The screenshot shows the Joomla! administrator interface for configuring a module. At the top, there is a header with a Joomla! logo and the text 'Модуль: [ Редагувати ]'. Below this is a blue notification bar that says 'Пункт збережено'. The main content area is titled 'Деталі' and contains the following configuration options:

- Тип Модуля:** mod\_radioplayerjoomla-free
- Заголовок:** Radio Player Joomla FREE
- Показати Заголовок:**  Ні  Так
- Включено:**  Ні  Так
- Розташування:** left
- Сортувати:** 2::Radio Player Joomla FREE
- Права Доступу:** Загальнодоступний, Зареєстрований, Особливий
- ID:** 16
- Опис:** Free Radio Player for Joomla by Dimedia! If you need help, open support ticket: <http://support.radioplayerjoomla.tk!> Free ShoutCast Hosting: <http://www.bumhosting.net>

Below the 'Деталі' section is the 'Призначення Меню' section:

- Меню:**  Всі  Ні  Обрати Пункт(и) Меню зі Списку
- Вибір Меню:** mainmenu, Home

Рисунок 18 - Стандартні налаштування модуля у CMS Joomla!

Для модуля «Radio Player Joomla FREE» у правій частині екрану (рис. 19) можна задати такі параметри як:

**Налаштування**

▼ **Параметри Модуля**

<b>Radio Title</b>	<input type="text" value="MyRadio"/>
<b>Skin</b>	<input type="text" value="McClean [FREE] ▼"/>
<b>Auto Start</b>	<input type="text" value="Так ▼"/>
<b>Volume</b>	<input type="text" value="80 ▼"/>
<b>SHOUTCast Stream URL</b>	<input type="text" value="http://s1.bumhosting.net:8000"/>
<b>Stream type</b>	<input type="text" value="Shoutcast ▼"/>
<b>Intro Audio File (optional)</b>	<input type="text"/>
<b>Fallback File (optional)</b>	<input type="text"/>
<b>Message (optional)</b>	<input type="text" value="Welcome"/>
<b>Codec</b>	<input type="text" value="mp3 ▼"/>
<b>Allow tracking?</b>	<input type="text" value="Так ▼"/>

Рисунок 19 - налаштування модуля «Radio Player Joomla FREE»

- **Radio Title** (назва радіо) – назва радіостанції, яка буде виводитися у верхній частині модуля на сайті.
- **Skin** (зовнішній вигляд) – вигляд що матиме модуль, у даній версії модуля доступно 7 варіантів зовнішнього вигляду (рис. 20).

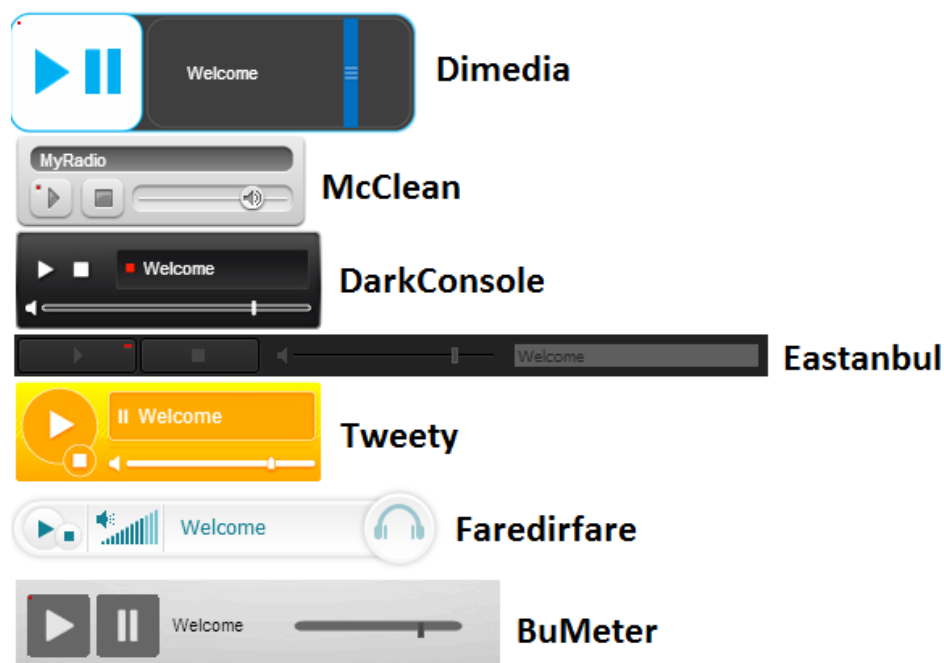


Рисунок 20 - приклади зовнішнього вигляду модуля «Radio Player Joomla FREE»

- **Auto Start** – визначає чи буде починатись відтворювання аудіо потоку одразу ж після завантаження сторінки.
- **Volume** – визначає гучність що буде встановлюватися за замовчуванням при завантаженні сторінки (надалі її можна буде змінювати).
- **SHOUTCast Stream URL** – це адреса нашого медіа серверу разом із портом. Адреса задається у форматі «`http://{serverAdress:serverPort/streamName}`», де
  - `serverAdress` - IP чи домена адреса медіа серверу, якщо сервер на тому ж комп'ютері, то можна вказати адресу 127.0.0.1, або localhost.
  - `serverPort` - номер порту, що вказаний у налаштуваннях у файлі «`icestream.xml`», якщо використовується сервер IceCast, або файлі



«sc\_serv\_basic.conf», якщо використовується сервер SHOUTcast.

- streamName - ім'я джерела, що вказується на радіостанції, як параметр передачі аудіо потоку на медіа сервер.
- Stream type (вид потоку) – визначає вид медіа серверу, від якого приймається потік. Можливе підключення лише до серверів IceCast та SHOUTcast обранням відповідного пункту.
- Intro Audio File (optional) - за бажанням можна вказати уніфікований локатор ресурсів, що вказує на файл, який необхідно відтворити перед відтворенням потоку радіостанції.
- Fallback File (optional) - за бажанням можна вказати уніфікований локатор ресурсів, що вказує на файл, який необхідно відтворювати в разі припинення надходження аудіо потоку від медіа серверу.
- Message (optional) – за бажанням можна вказати повідомлення, що буде виводитися по черзі разом із назвою радіостанції вказаної у пункті «Radio Title».
- Codec – кодек, що буде використовуватись для декодування та відтворення аудіо потоку, що йде від медіа серверу. Можливе використання двох кодеків: MP3 чи OGG.
- Allow tracking? – пункт, що дозволяє модулю відправляти певні данні розробнику для аналізу та покращення роботи модуля.

Після усіх налаштувань треба натиснути кнопку зберегти і результат буде видно на сторінках сайту вказаних у лівій частині сторінки налаштування модуля.

### 4.3 Тестування

Після реалізації даного сервісу було проведено тестування роботи сервісу. Було проведено підключення до адміністративної панелі медіа серверу. Також було перевірено отримання аудіо потоку із сайту кафедри. Отримані результати повністю відповідають поставленим задачам.

Зовнішній вигляд сторінки сайту, з якої можна звернутися до сервісу показано на рис.21

**Усім**

- Новини
- Попередній запис абітурієнтів

**Абітурієнтам**

- Чому слід обирати СП (САПР)?
- Бакалавр (на базі 11 кл.)
- Вступ на 5 курс
- Заочна форма навчання
- Друга вища освіта
- Працевлаштування
- Відлуки випускників

**Студентам**

- Розклад занять
- Навчальна документація
- Студентські публікації
- Дипломні проекти
- Опис дисциплін

**Викладачам**

- Графік зайнятості
- Публікації
- Сайти викладачів
- Викладачі

**Корисні посилання**

- Бібліотека НТУУ "КПІ"
- Студентська рада
- Конференція САІТ
- NetALLTED Моделювання
- ГРІД Україна
- Сумісна лабораторія Melexis
- E-campus

**Віталій Петрович Сігорський**  
22.11.1922 – 13.08.2007

Ім'я Сігорського Віталія Петровича є одним з найвизначніших імен в історії кафедри Системного проектування. Сігорський Віталій Петрович був видатним вченим, що стояв у витоків ідеї САПР. Саме під його керівництвом протягом 1964-1970 років формувалась кафедра. Кафедра здійснювала підготовку спеціалістів по спеціалізації "електронні пристрої систем управління" спеціальності 0612 ("Промислова електроніка") у складі однієї академічної групи (25 студентів). Почались і активно проводились дослідження методів і алгоритмів аналізу лінійних електронних схем, перетворювачів графічної інформації для ЕОМ, масштабно-часових перетворювачів сигналів та іншого. У цей час формувався колектив кафедри і створювалися його традиції. Слід відзначити, що це був період, по суті, відродження спеціалізації "Технічної електроніки", створеної у 1956 р. доц., д.т.н. С.В. Свечніковим на кафедрі "електронних приладів" (сьогодні - академік НАН, директор Інституту напівпровідників). В 1962 р. (після переходу Свечнікова С.В. у НАН України) була створена кафедра "Промислова електроніка" (завідувач доц., к.т.н Руденко С.В.), в складі якої базовий кістяк попередньої спеціалізації (доц. Петренко А.І., доц. Калніболотський Ю.М., ст. викл. Вакуленко Е.М., асс. Денбіновецький С.В., асс. Абакумов В.Г., асс. Будняк А.А.) продовжував працювати в галузі сигнальної електроніки і систем керування. Коли до КПІ прийшов проф. Сігорський В.П., відроджена спеціалізація на новій кафедрі у порівняно короткий термін досягла істотних успіхів за усіма напрямками своєї діяльності: навчально-методичної, наукової та організаційно-виховної.

2012 © Кафедра СП  
Про кафедру | Контакти | НТУУ "КПІ" | ННК "ІПСА"

Рисунок 21 - сторінка сайту з модулем «Radio Player Joomla FREE»

#### **4.4 Висновки**

У даному розділі було описано реалізацію сервісу Інтернет радіо. Детально описано процес встановлення та налаштування медіа серверу на фізичний сервер кафедри. На сайт кафедри СП було додано радіо модуль, що відтворює ефір, котрий надходить із медіа серверу.

За результатами тестування медіа сервер і модуль на сайті кафедри СП працюють відповідно з вимогами поставленими до сервісу Інтернет радіо.

## 5 ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ ПРОГРАМНОГО ПРОДУКТУ

У даному розділі проводиться оцінка основних характеристик програмного продукту, призначеного для створення інтернет радіомовлення у 26 корпусі НТУУ «КПІ».

Програмний продукт є кросплатформеним і може використовуватись, як на персональних комп'ютерах під управлінням операційної системи Windows, так і на ПЕОМ на яких встановленні UNIX-подібні операційні системи.

Нижче наведено аналіз різних варіантів реалізації модулю з метою вибору оптимальної, з огляду при цьому як на економічні фактори, так і на характеристики продукту, що впливають на продуктивність роботи і на його сумісність з апаратним забезпеченням. Для цього було використано апарат функціонально-вартісного аналізу.

Функціонально-вартісний аналіз (ФВА) – це технологія, яка дозволяє оцінити реальну вартість продукту або послуги незалежно від організаційної структури компанії. Як прямі, так і побічні витрати розподіляються по продуктам та послугам у залежності від потрібних на кожному етапі виробництва обсягів ресурсів. Виконані на цих етапах дії у контексті метода ФВА називаються функціями.

Мета ФВА полягає у забезпеченні правильного розподілу ресурсів, виділених на виробництво продукції або надання послуг, на прямі та непрямі витрати. У даному випадку – аналізу функцій програмного продукту й виявлення усіх витрат на реалізацію цих функцій.

Фактично цей метод працює за таким алгоритмом:

– визначається послідовність функцій, необхідних для виробництва продукту. Спочатку – всі можливі, потім вони розподіляються по двом групам: ті, що впливають на вартість продукту і ті, що не впливають. На цьому ж етапі

оптимізується сама послідовність скороченням кроків, що не впливають на цінність і відповідно витрат.

- для кожної функції визначаються повні річні витрати й кількість робочих часів.

- для кожної функції на основі оцінок попереднього пункту визначається кількісна характеристика джерел витрат.

- після того, як для кожної функції будуть визначені їх джерела витрат, проводиться кінцевий розрахунок витрат на виробництво продукту.

## **5.1 Постановка задачі техніко-економічного аналізу**

У роботі застосовується метод ФВА для проведення техніко-економічний аналізу розробки.

Відповідно цьому варто обирати і систему показників якості програмного продукту.

Технічні вимоги до продукту наступні:

- програмний продукт повинен функціонувати на персональних комп'ютерах із стандартним набором компонент;

- забезпечувати стабільне, безперебійне мовлення до клієнтів;

- забезпечувати зручність і простоту взаємодії з користувачем або з адміністратором програмного забезпечення;

- передбачати мінімальні витрати на впровадження програмного продукту.

### **5.1.1 Обґрунтування функцій програмного продукту**

Головна функція  $F_0$  – обрання програмного продукту для встановлення на фізичний сервер 26 корпусу НТУУ «КПІ». Виходячи з конкретної мети, можна виділити наступні основні функції ПП:

$F_1$  – вибір програмного забезпечення радіостанції;

$F_2$  – вибір програмного забезпечення медіа серверу;

$F_3$  – вибір операційної системи на яку буде встановлюватися програмне забезпечення.

Кожна з основних функцій може мати декілька варіантів реалізації.

Функція  $F_1$ :

- а) програвач PulseAudio;
- б) програвач RadioBOSS;

Функція  $F_2$ :

- а) медіа сервер IceCast;
- б) медіа сервер SHOUTcast.

Функція  $F_3$ :

- а) операційна система Red Hat Enterprise Linux;
- б) операційна система Windows Server 2012.

### 5.1.2 Варіанти реалізації основних функцій

Варіанти реалізації основних функцій наведені у морфологічній карті системи (рис. 22). На основі цієї карти побудовано позитивно-негативну матрицю варіантів основних функцій (таблиця 1).

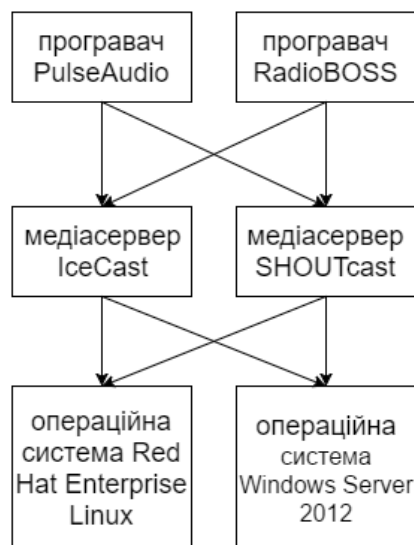


Рисунок 22 - Морфологічна карта

Таблиця 1 – Позитивно-негативна матриця

Основні функції	Варіанти реалізації	Переваги	Недоліки
<i>F1</i>	<i>A</i>	Кросплатформний, безкоштовний, може працювати з декількома джерелами, ефективний мікшер, вбудоване перетворення аудіо форматів, наявний модуль придушення акустичного відлуння	Складніший інтерфейс ніж у аналога.
	<i>B</i>	Має надійний генератор плейлистів, має вбудований генератор звітів, можливість вставляння рекламних, або новинних блоків	Платний, не кросплатформний.
<i>F2</i>	<i>A</i>	Кросплатформний, можливе приймання декількох потоків, підтримка багатьох форматів як аудіо, так і відео, можливість блокування IP адресів, можливість відключення користувачів та перекидання їх на інший аудіо потік. Наявна повноцінна документація.	Неможливо заблокувати підмережу IP адрес.
	<i>B</i>	Кросплатформний, підтримка багатьох форматів як аудіо, так і відео, можливість блокування IP адресів та одного підмереж IP адрес, користувача. Немає можливість відключення усіх користувачів.	Неможливе приймання декількох потоків. Неможливе відключення певного користувача. Немає гарної документації.

Таблиця 1 (продовження)

Основні функції	Варіанти реалізації	Переваги	Недоліки
<i>F3</i>	<i>A</i>	Безкоштовна, високопродуктивна система.	Немає графічного інтерфейсу користувача.
	<i>B</i>	Найявний графічний інтерфейс користувача.	Платна, менш продуктивна за аналог

На основі аналізу позитивно-негативної матриці робимо висновок, що при обрані програмного продукту деякі варіанти реалізації функцій варто відкинути, тому, що вони не відповідають поставленим перед програмним продуктом задачам. Ці варіанти відзначені у морфологічній карті.

#### Функція *F1*:

Оскільки радіостанція може бути встановлена на різні сервери, та повинна бути безкоштовною, слід обрати варіант А, оскільки він є кросплатформеним.

#### Функція *F2*:

Обидва сервери можуть бути встановлені на різні операційні системи та різниця у їх функціональності не є дуже великою, тому обидва варіанти А і Б гідні розгляду.

#### Функція *F3*:

Оскільки завдання повинно потребувати мінімуму затрат, а для медіа серверу доступне віддалене адміністрування, тому обираємо варіант А.

Таким чином, будемо розглядати такі варіанти реалізації ПП:

1. *F1a – F2a – F3a*
2. *F1a – F2б – F3a*

Для оцінювання якості розглянутих функцій обрана система параметрів, описана нижче.



## 5.2 Обґрунтування системи параметрів ПП

### 5.2.1 Опис параметрів

На підставі даних про основні функції, що повинен мати програмний продукт, вимог до нього, визначаються основні параметри виробу, що будуть використані для розрахунку коефіцієнта технічного рівня.

Для того, щоб охарактеризувати програмний продукт, будемо використовувати наступні параметри:

- $X1$  – завантаженість підключення до Інтернету;
- $X2$  – об'єм оперативної пам'яті, що використовується;
- $X3$  – час необхідний на підключення до медіа серверу;
- $X4$  – кількість функцій, доступних для віддаленого доступу.

$X1$ : Відображає кількість трафіку, що споживається під час роботи медіа серверу.

$X2$ : Відображає об'єм пам'яті в оперативній пам'яті персонального комп'ютера, необхідний для стабільної, безперебійної роботи програми.

$X3$ : Відображає час, який необхідно користувачу, для підключення до медіа серверу.

$X4$ : Показує кількість функцій, що доступні для адміністратора при віддаленому підключенні до серверу.

### 5.2.2 Кількісна оцінка параметрів

Гірші, середні і кращі значення параметрів вибираються на основі вимог замовника й умов, що характеризують експлуатацію ПП як показано у таблиці 2.

Таблиця 2 – Основні параметри ПП

Назва Параметра	Умовні позначення	Одиниці виміру	Значення параметра		
			гірші	середні	кращі
Завантаженість підключення до Інтернету	X1	Мб/с	1000	100	50
Об'єм оперативної пам'яті, що використовується	X2	МБ	1000	750	400
Час необхідний на підключення до медіа серверу	X3	с	10	7	5
Кількість функцій, доступних для віддаленого доступу	X4	Шт.	2	5	10

За даними таблиці 2 будуються графічні характеристики параметрів – рис.

23 – рис. 26

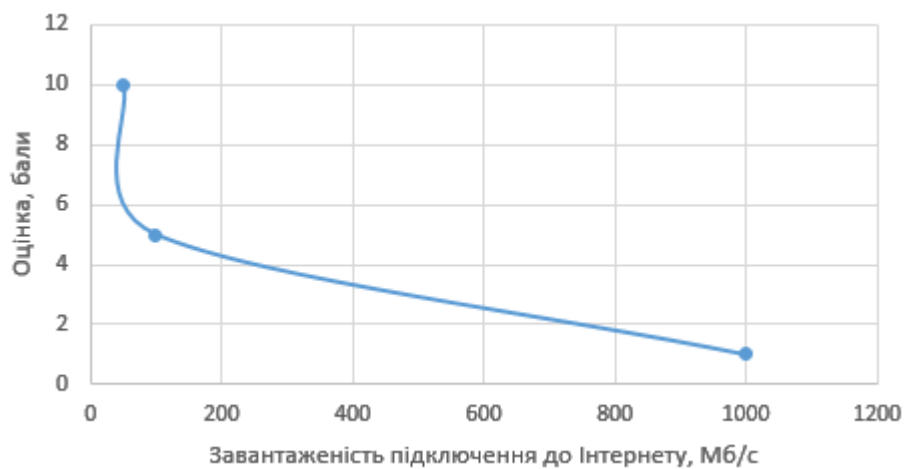


Рисунок 23 – X1, завантаженість підключення до Інтернету

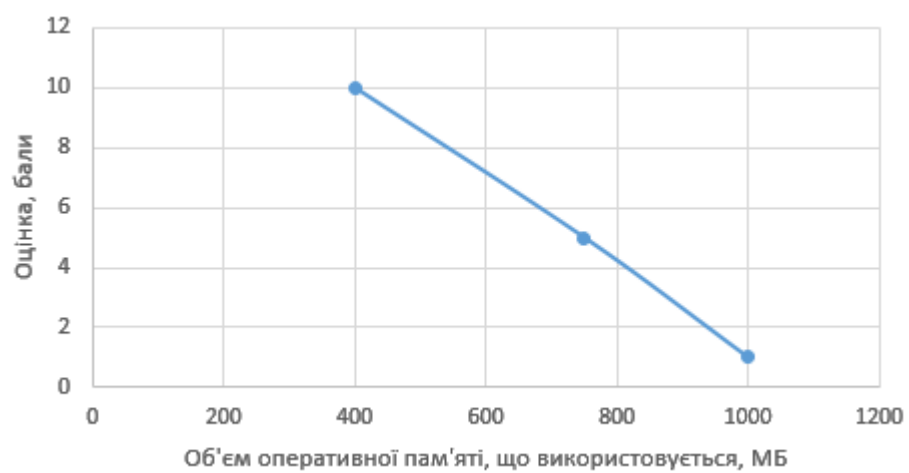


Рисунок 24 – X2, об'єм оперативної пам'яті, що використовується

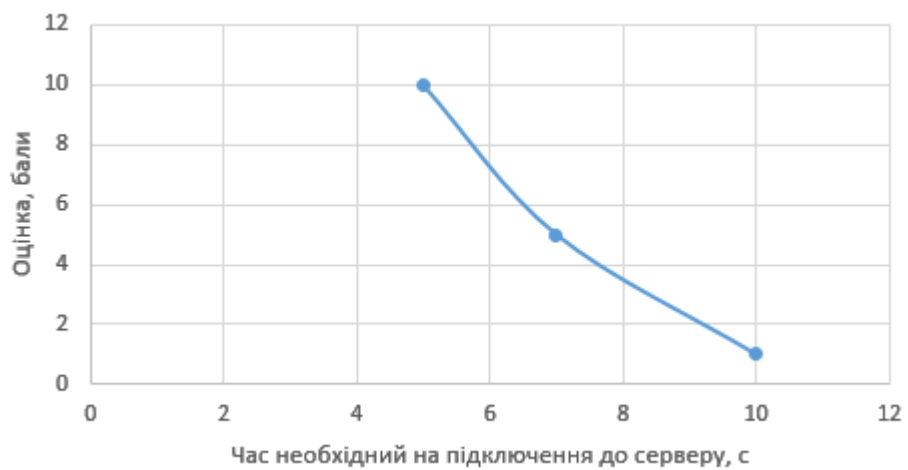


Рисунок 25 – X3, час необхідний на підключення до медіа серверу

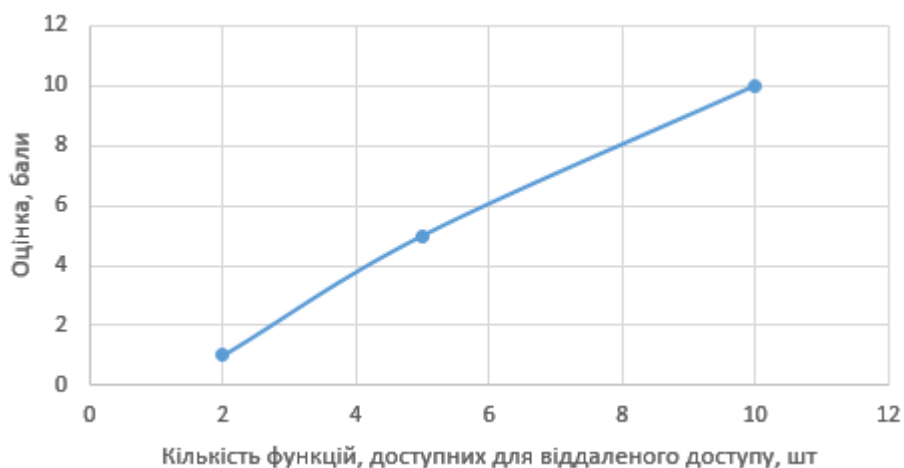


Рисунок 26 – Х4, кількість функцій, доступних для віддаленого доступу

### 5.2.3 Аналіз експертного оцінювання параметрів

Після детального обговорення й аналізу кожний експерт оцінює ступінь важливості кожного параметру для конкретно поставленої цілі – вибір програмного продукту, який має найбільш зручний та функціональний набір можливостей при низькій необхідності у ресурсах.

Значимість кожного параметра визначається методом попарного порівняння. Оцінку проводить експертна комісія із 7 людей. Визначення коефіцієнтів значимості передбачає:

- визначення рівня значимості параметра шляхом присвоєння різних рангів;
- перевірку придатності експертних оцінок для подальшого використання;
- визначення оцінки попарного пріоритету параметрів;
- обробку результатів та визначення коефіцієнту значимості.

Результати експертного ранжування наведені у таблиці 3.

Таблиця 3 – Результати ранжування параметрів

Позначення параметра	Назва параметра	Одиниці виміру	Ранг параметра за оцінкою експерта							Сума рангів $R_i$	Відхилення $\Delta_i$	$\Delta_i^2$
			1	2	3	4	5	6	7			
X1	Завантаженість підключення до Інтернету	Мб/с	1	2	2	2	1	2	2	12	-5,5	30,25
X2	Об'єм оперативної пам'яті, що використовується	МБ	2	1	1	1	2	1	1	9	-8,5	72,25
X3	Час необхідний на підключення до медіа серверу	с	3	3	4	3	4	4	3	24	6,5	42,25
X4	Кількість функцій, доступних для віддаленого доступу	Кількість функцій	4	4	3	4	3	3	4	25	7,5	56,25
	Разом		10	10	10	10	10	10	10	70	0	201

Для перевірки степені достовірності експертних оцінок, визначимо наступні параметри:

а) сума рангів кожного з параметрів і загальна сума рангів:

$$R_i = \sum_{j=1}^N r_{ij} R_{ij} = \frac{Nn(n+1)}{2} = 70,$$

де  $N$  – число експертів,  $n$  – кількість параметрів;

б) середня сума рангів:

$$T = \frac{1}{n} R_{ij} = 17,5$$

в) відхилення суми рангів кожного параметра від середньої суми рангів:

$$\Delta_i = R_i - T$$

Сума відхилень по всім параметрам повинна дорівнювати 0;

г) загальна сума квадратів відхилення:

$$S = \sum_{i=1}^N \Delta_i^2 = 201.$$

Порахуємо коефіцієнт узгодженості:

$$W = \frac{12S}{N^2(n^3 - n)} = \frac{12 \cdot 201}{7^2(4^3 - 4)} = 0,82 > W_k = 0,67$$

Ранжування можна вважати достовірним, тому що знайдений коефіцієнт узгодженості перевищує нормативний, котрий дорівнює 0,67.

Скориставшись результатами ранжирування, проведемо попарне порівняння всіх параметрів і результати занесемо у таблицю 4.

Таблиця 4 – Попарне порівняння параметрів

Параметри	Експерти							Кінцева оцінка	Числове значення
	1	2	3	4	5	6	7		
X1 і X2	>	<	<	<	>	<	<	<	0,5
X1 і X3	>	>	>	>	>	>	>	>	1,5
X1 і X4	>	>	>	>	>	>	>	>	1,5
X2 і X3	>	>	>	>	>	>	>	>	1,5
X2 і X4	>	>	>	>	>	>	>	>	1,5
X3 і X4	>	>	<	>	<	<	>	>	1,5

Числове значення, що визначає ступінь переваги  $i$ -го параметра над  $j$ -тим,  $a_{ij}$  визначається по формулі:

$$a_{ij} = \begin{cases} 1,5 \text{ при } X_i > X_j \\ 1,0 \text{ при } X_i = X_j \\ 0,5 \text{ при } X_i < X_j \end{cases}$$

З отриманих числових оцінок переваги складемо матрицю  $A = \| a_{ij} \|$ .

Для кожного параметра зробимо розрахунок вагомості  $K_{ei}$  за наступними формулами:

$$K_{Vi} = \frac{b_i}{\sum_{i=1}^n b_i}, \text{ де } b_i = \sum_{i=1}^N a_{ij}.$$

Відносні оцінки розраховуються декілька разів доти, поки наступні значення не будуть незначно відрізнятись від попередніх (менше 2%). На другому і наступних кроках відносні оцінки розраховуються за наступними формулами:

$$K_{Vi} = \frac{b'_i}{\sum_{i=1}^n b'_i}, \text{ де } b'_i = \sum_{i=1}^N a_{ij} b_j.$$

Як видно з таблиці 5, різниця значень коефіцієнтів вагомості не перевищує 2%, тому більшої кількості ітерацій не потрібно.

Таблиця 5 – Розрахунок вагомості параметрів

Параметри $x_i$	Параметри $x_j$				Перша ітер.		Друга ітер.		Третя ітер	
	X1	X2	X3	X4	$b_i$	$K_{Vi}$	$b_i^1$	$K_{Vi}^1$	$b_i^2$	$K_{Vi}^2$
X1	1,0	0,5	1,5	1,5	4,5	0,281	16,25	0,275	59,125	0,274
X2	1,5	1,0	1,5	1,5	5,5	0,344	21,25	0,360	77,875	0,361
X3	0,5	0,5	1,0	1,5	3,5	0,219	12,25	0,208	44,875	0,207
X4	0,5	0,5	0,5	1,0	2,5	0,156	9,25	0,157	34,125	0,158
Всього:					16	1	59	1	216	1

### 5.3 Аналіз рівня якості варіантів реалізації функцій

Визначаємо рівень якості кожного варіанту виконання основних функцій окремо.

Абсолютні значення параметрів X1 (Завантаженість підключення до Інтернету) та X3 (Час необхідний на підключення до медіа серверу) відповідають технічним вимогам умов функціонування даного ПП.

Абсолютне значення параметра X2 (Об'єм оперативної пам'яті, що використовується) буде найкращим у випадку обрання у F2 варіанта А і становитиме 400, для варіанту Б це значення буде 1000.

Абсолютне значення параметра X4 (Кількість функцій, доступних для віддаленого доступу) буде найкраще при обрані варіанту Б 10, а при обрані варіанту А воно буде середнім 5.

Коефіцієнт технічного рівня для кожного варіанта реалізації ПП розраховується так (таблиця 6):

$$K_K(j) = \sum_{i=1}^n K_{ei,j} B_{i,j},$$

де  $n$  – кількість параметрів;  $K_{ei}$  – коефіцієнт вагомості  $i$ -го параметра;  $B_i$  – оцінка  $i$ -го параметра в балах.

Таблиця 6 – Розрахунок показників рівня якості варіантів реалізації основних функцій ПП

Основні функції	Варіант реалізації функції	Параметри	Абсолютне значення параметра	Бальна оцінка параметра	Коефіцієнт вагомості параметра	Коефіцієнт рівня якості
F1	A	X1	100	5	0,274	1,37
F2	A	X2	500	8,6	0,361	3,1
		X4	4	3,7	0,207	0,77
	B	X2	950	1,8	0,361	0,65
		X4	9	9	0,207	1,863
F3	A	X3	6	4,2	0,158	0,66

За даними з таблиці 6 за формулою

$$K_K = K_{TY}[F_{1k}] + K_{TY}[F_{2k}] + \dots + K_{TY}[F_{zk}],$$

визначаємо рівень якості кожного з варіантів:

$$K_{K1} = 1,37 + 3,1 + 0,77 + 0,66 = 5,9$$

$$K_{K2} = 1,37 + 0,65 + 1,863 + 0,66 = 4,543$$

Як видно з розрахунків, кращим є перший варіант, для якого коефіцієнт технічного рівня має найбільше значення.



## 5.4 Економічний аналіз варіантів розробки ПП

Програмний продукт, що буде встановлюватись на сервер кафедри вже розроблено, проте необхідно увесь рік тримати під контролем сервер, на якому буде встановлено програмне забезпечення. Отже

$$T_1 = 365 \text{ людино-днів на рік.}$$

$$T_1 = 365 \cdot 8 = 2920 \text{ людино-годин;}$$

Обидва варіанти однаково трудомісні

В адмініструванні беруть участь два адміністратори з окладом 5000 грн.

Визначимо зарплату за годину за формулою:

$$C_{\text{ч}} = \frac{M}{T_m \cdot t} \text{ грн.,}$$

де  $M$  – місячний оклад працівників;  $T_m$  – кількість робочих днів тиждень;  $t$  – кількість робочих годин в день.

$$C_{\text{ч}} = \frac{5000 + 5000}{2 \cdot 21 \cdot 8} = 29,76 \text{ грн.}$$

Тоді, розрахуємо заробітну плату за формулою

$$C_{\text{зп}} = C_{\text{ч}} \cdot T_i \cdot K_{\text{д}},$$

де  $C_{\text{ч}}$  – величина погодинної оплати праці програміста;  $T_i$  – трудомісткість відповідного завдання;  $K_{\text{д}}$  – норматив, який враховує додаткову заробітну плату.

Зарплата розробників за варіантами становить:

$$C_{\text{зп}} = 29,76 \cdot 2920 \cdot 1,2 = 104285,71 \text{ грн на рік.}$$

Відрахування на єдиний соціальний внесок незалежно від групи професійного ризику становить 22%:

$$I. \quad C_{\text{від}} = C_{\text{зп}} \cdot 0,22 = 104285,71 \cdot 0,22 = 22,942,86 \text{ грн.}$$

Тепер визначимо витрати на оплату однієї машино-години. ( $C_{\text{м}}$ )

Працюватиме одна електронна обчислювальна машина, котра працюватиме цілодобово:

$$C_{\text{г}} = 12 \cdot M \cdot K_3 = 12 \cdot 5000 \cdot 0,2 = 12000 \text{ грн.}$$

З урахуванням додаткової заробітної плати:

$$C_{ЗП} = C_{Г} \cdot (1 + K_3) = 12000 \cdot (1 + 0.2) = 14400 \text{ грн.}$$

Відрахування на єдиний соціальний внесок:

$$C_{ВІД} = C_{ЗП} \cdot 0.22 = 14400 \cdot 0.22 = 3168 \text{ грн.}$$

Амортизаційні відрахування розраховуємо при амортизації 25% та вартості ЕОМ – 25000 грн.

$$C_A = K_{ТМ} \cdot K_A \cdot C_{ПР} = 1.15 \cdot 0.25 \cdot 25000 = 7187,5 \text{ грн.,}$$

де  $K_{ТМ}$  – коефіцієнт, який враховує витрати на транспортування та монтаж приладу у користувача;  $K_A$  – річна норма амортизації;  $C_{ПР}$  – договірна ціна приладу.

Витрати на ремонт та профілактику розраховуємо як:

$$C_P = K_{ТМ} \cdot C_{ПР} \cdot K_P = 1.15 \cdot 25000 \cdot 0.05 = 1437,5 \text{ грн.,}$$

де  $K_P$  – відсоток витрат на поточні ремонти.

Ефективний годинний фонд часу ПК за рік розраховуємо за формулою:

$$T_{ЕФ} = D_K \cdot t_3 \cdot K_B = 365 \cdot 24 \cdot 1 = 8760 \text{ годин,}$$

де  $D_K$  – календарна кількість днів у році;  $t_3$  – кількість робочих годин електронної обчислювальної машини в день;  $K_B$  – коефіцієнт використання приладу у часі протягом зміни.

Витрати на оплату електроенергії розраховуємо за формулою:

$$C_{ЕЛ} = T_{ЕФ} \cdot N_C \cdot K_3 \cdot C_{ЕН} = 8760 \cdot 0,6 \cdot 1 \cdot 2,0218 = 10626,58 \text{ грн.,}$$

де  $N_C$  – середньо-споживча потужність приладу;  $K_3$  – коефіцієнтом зайнятості приладу;  $C_{ЕН}$  – тариф за 1 кВт-годин електроенергії.

Накладні витрати розраховуємо за формулою:

$$C_H = C_{ПР} \cdot 0.67 = 25000 \cdot 0,67 = 16750 \text{ грн.}$$

Тоді, річні експлуатаційні витрати будуть:

$$C_{ЕКС} = C_{ЗП} + C_{ВІД} + C_A + C_P + C_{ЕЛ} + C_H$$

$$C_{ЕКС} = 14400 + 3168 + 7187,5 + 1437,5 + 10626,58 + 16750 = 53569,58 \text{ грн.}$$

Собівартість однієї машино-години ЕОМ дорівнюватиме:

$$C_{М-Г} = C_{ЕКС} / T_{ЕФ} = 53569,58 / 8760 = 6,12 \text{ грн/час.}$$

Оскільки в даному випадку всі роботи, які пов'язані з розробкою програмного продукту ведуться на ЕОМ, витрати на оплату машинного часу, в залежності від обраного варіанта реалізації, складає:

$$C_M = C_{M-G} \cdot T$$

$$C_M = 7,26 * 8760 = 53569,58$$

Накладні витрати складають 67% від заробітної плати:

$$C_H = C_{ЗП} \cdot 0,67$$

$$C_H = 104285,71 * 0,67 = 69871,43 \text{ грн.};$$

Отже, вартість розробки ПП за варіантами становить:

$$C_{ПП} = C_{ЗП} + C_{Вид} + C_M + C_H$$

$$C_{ПП} = 104285,71 + 22942,86 + 53569,58 + 69871,43 = 250669,58 \text{ грн.};$$

## 5.5 Вибір кращого варіанта ПП техніко-економічного рівня

Розрахуємо коефіцієнт техніко-економічного рівня за формулою:

$$K_{TEPj} = K_{Kj} / C_{Фj},$$

$$K_{TEP1} = 5,9 / 250669,58 = 2,35 \cdot 10^{-5};$$

$$K_{TEP2} = 4,543 / 250669,58 = 1,81 \cdot 10^{-5};$$

Як бачимо, найбільш ефективним є перший варіант програми з коефіцієнтом техніко-економічного рівня  $K_{TEP1} = 2,35 \cdot 10^{-5}$ .

## 5.6 Висновки до економічного розділу розділу

В даному розділі було проведено функціонально-вартісний аналіз програмного забезпечення, що буде встановлено у 26 корпусі НТУУ «КПІ». Процес аналіз складався з двох частин.

У першій проведено дослідження програмний продукт з технічної точки зору: були поставлені основні параметри, що повинні бути головними при обранні кращої реалізації. На основі отриманих значень параметрів, оцінок

експертів було обчислено коефіцієнт технічного рівня, який надалі у другій частині допоміг обрати найкращий варіант з техніко-економічної точки зору.

У другій частині виконувалося економічне обґрунтування альтернативних варіантів реалізації. Порівняння робились з урахуванням витрат на заробітні плати, електроенергії, накладні витрати.

Після виконання функціонально-вартісного аналізу програмного комплексу, що буде встановлюватися на сервер 26 корпусу НТУУ «КПІ», що залишилися після проведення першої частини аналізу було виявлено, що перший варіант є найбільш оптимальним для реалізації. Його показник техніко-економічного рівня якості  $K_{\text{ТЕР1}} = 2,35 \cdot 10^{-5}$ ;

Цей варіант реалізації програмного продукту має такі параметри:

- Програваач PulseAudio;
- Медіа сервер IceCast;
- Операційна система Red Hat Enterprise Linux.

Даний варіант реалізації є найбільш продуктивним, за рахунок використання більш оптимальної операційної системи та медіа серверу. Набір функцій доступних користувачу є достатнім. Програваач є оптимальним рішенням, за рахунок його безкоштовності.

## ВИСНОВКИ

В ході даної дипломної роботи було досліджено принципи побудови Інтернет радіо, основними частинами якого є радіостанція, медіа сервер та клієнт, що прослуховує ефір.

Проаналізувавши формати аудіо трафіку можна зробити висновок, що для радіомовлення найбільш ефективними є алгоритми стиснення з втратою даних, адже вони дають найбільше стиснення при належній якості звучання. Найкращим форматом є OGG Vorbis, проте його мале застосування порівняно з MP3 та AAC змушує застосовувати саме їх.

Було розглянуто протоколи передачі даних від медіа сервера до клієнта через мережу Інтернет. За розповсюдженою схемою для кожного клієнта з сервера відправляється окремий потік даних, за допомогою протоколу TCP. Покращити продуктивність роботи можна за допомогою протоколу RTP/UDP та багатоадресної передачі по IP групам адресатів. Проте через те, що не всі маршрутизатори на шляху аудіо потоку можуть підтримувати багатоадресну передачу, використання більш оптимальних технологій проблематичне.

Не дивлячись на те, що для передачі медіа в реальному часі використовують протокол UDP, при радіомовленні можливе використання саме TCP, що дає гарантію доставки усіх аудіо даних. Можливість цього досягається за допомогою затримки мовлення на 10-15 секунд від реального ефіру. Затримка дає змогу приховувати «джитери», збої синхронізації, що виникають як результат не ідеальності мережі, через які передаються дані, завдяки частковій буферизації даних на комп'ютері слухача.

Було проаналізовано найпопулярніші безкоштовні варіанти реалізації медіа серверів: IceCast та SHOUTcast. З точки зору можливостей, великої різниці між даними програмними засобами немає. Обидва медіа сервери можуть підтримувати мовлення одночасно від декількох радіостанцій, показувати кількість спожитого трафіку, відображати метадані, що поступали від

радіостанцій та мали гарну повноцінну документацію англійською мовою. Проте з точки зору споживання ресурсів медіа сервер IceCast виявився на порядок економнішим, що й стало головним критерієм обрання саме цього програмного продукту для подальшого встановлення на фізичний сервер кафедри.

Після вибору реалізації медіа серверу, було досліджено можливість підключення до цього програмного модулю із сайту кафедри. Було проаналізовано спосіб отримання аудіо потоку, як з використанням стандартних HTML тегів, так і з використанням спеціального модуля для системи керування вмістом Joomla! версії 1.5.

Було сформульовано план робіт, необхідний для встановлення та налаштування програмного забезпечення на сервері кафедри виходячи із необхідності підтримки роботи при кількості підключень  $\leq 100$  та можливості установки на сервер під керуванням операційної системи Linux.

Медіа сервер був встановлений на фізичний сервер кафедри, після чого було проведено його налаштування. Програма була встановлена як демон (сервіс), що буде працювати на задньому фоні увесь час, доки працює сервер, з можливістю перезапуску.

Результати тестування роботи серверу показали, що він коректно передає аудіо потік від радіостанції до користувачів, що підключаються. Також виводиться метадані музичного файлу, що програвється. У адміністративній панелі серверу доступна більш детальна інформація про використання мережі, кількості радіостанцій, що віщають через медіа сервер, та кількість клієнтів, що зараз прослуховують кожну радіостанцію.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Internet Radio Trends Report 2015. – Режим доступу:  
[http://xappmedia.com/wp-content/uploads/2015/01/Internet-Radio-Trends-Report-2015\\_january.pdf](http://xappmedia.com/wp-content/uploads/2015/01/Internet-Radio-Trends-Report-2015_january.pdf). – Дата доступу: 31.05.2016
2. Number of monthly digital radio listeners in the United States from 2010 to 2019.  
– Режим доступу: <http://www.statista.com/statistics/191657/us-weekly-internet-radio-listeners-since-2009/>. – Дата доступу: 31.05.2016
3. Таненбаум Э. Компьютерные сети 5-е издание / Эндрю Таненбаум. – Санкт-Петербург : «Питер», 2012. -С. , 739-745, 763-767
4. RadioBOSS Описание программы. – Режим доступу:  
[http://www.radioboss.ru/rus/description\\_radioboss.htm](http://www.radioboss.ru/rus/description_radioboss.htm). – Дата доступу:  
31.05.2016
5. Interviews/LennartPoettering — FedoraProject. – Режим доступу:  
<http://fedoraproject.org/wiki/Interviews/LennartPoettering>. – Дата доступу:  
31.05.2016
6. Lennart Pottering. PulseAudio and Jack. – Режим доступу:  
<http://0pointer.de/blog/projects/when-pa-and-when-not>. – Дата доступу:  
31.05.2016
7. Tutorialspoint simple easy learning. – Режим доступу:  
[http://www.tutorialspoint.com/html/html\\_tutorial.pdf](http://www.tutorialspoint.com/html/html_tutorial.pdf). – Дата доступу:  
31.05.2016
8. Будник С.А. Який вибір ми зробимо: HTML 5 чи Flash? / Будник С.А. // Системний аналіз та інформаційні технології : 18-а Міжнародна науково-технічна конференція «SAIT 2016», Київ, 30 травня – 2 червня 2016 р. / ННК “ІПСА” НТУУ “КПІ”. – К.: ННК “ІПСА” НТУУ “КПІ”, 2016. – С. 330
9. Шеренковський А.О. Програмне забезпечення медіа-серверу Інтернет радіо / Шеренковський А.О. // Системний аналіз та інформаційні технології : 18-

а Міжнародна науково-технічна конференція «SAIT 2016», Київ, 30 травня – 2 червня 2016 р. / ННК “ІПСА” НТУУ “КПІ”. – К.: ННК “ІПСА” НТУУ “КПІ”, 2016. – С. 429

10. Request For Comments RTSP. – Режим доступу:  
<https://tools.ietf.org/html/rfc2326>. – Дата доступу: 31.05.2016
11. Request For Comments RTP 1996. – Режим доступу:  
<https://tools.ietf.org/html/rfc1889>. – Дата доступу: 31.05.2016
12. Request For Comments RTP 2003. – Режим доступу:  
<https://tools.ietf.org/html/rfc3550>. – Дата доступу: 31.05.2016
13. Радіо точка, офіційний сайт. – Режим доступу: <https://radio-tochka.com>. –  
Дата доступу: 31.05.2016
14. Radioheart, офіційний сайт. – Режим доступу: <http://radioheart.ru/>. – Дата  
доступу: 31.05.2016
15. Сайт кафедри системного проектування. – Режим доступу:  
<http://cad.kpi.ua/>. – Дата доступу: 31.05.2016
16. Joomla! Extensions. – Режим доступу: <http://extensions.joomla.org/extension>.  
– Дата доступу: 31.05.2016
17. Головна сторінка Вікіпедії. – Режим доступу: <https://uk.wikipedia.org>. –  
Дата доступу: 31.05.2016
18. Документація медіа серверу IceCast 2. – Режим доступу  
<http://icecast.org/docs/>. – Дата доступу: 31.05.2016
19. Документація медіа серверу SHOUTcast. – Режим доступу  
[http://wiki.shoutcast.com/wiki/SHOUTcast\\_Broadcaster](http://wiki.shoutcast.com/wiki/SHOUTcast_Broadcaster). – Дата доступу:  
31.05.2016
20. Dimedia radio player for Joomla. – Режим доступу  
<http://radioplayer.dimediakd.com/download.html>. – Дата доступу: 31.05.2016
21. Сторінка завантаження SHOUTcast. – Режим доступу  
<http://download.nullsoft.com/shoutcast/tools/>. – Дата доступу: 31.05.2016