Центральноукраїнський національний технічний університет
Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”
Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор
_____ Олексій СМІРНОВ
“____” _____ 2026 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему
“Дослідження та програмна реалізація системи клієнтського модуля ВКЗ з використанням технології WebRTC та алгоритмів адаптивного управління якістю потокового відео”

Виконав здобувач вищої освіти
ІІ курсу, групи КІ-25М
ОПП «Комп’ютерна інженерія»
спеціальності F7 «Комп’ютерна інженерія»
_____ Дроздов С.І.
«____» _____ 2026 р.

Керівник проекту
доктор філософії (PhD)
_____ Усік П.С.
«____» _____ 2026 р.

Нормоконтроль проекту
_____ Коваленко А.С.
«____» _____ 2026 р.

Рецензент _____
_____

Підписано КЕП
Центральноукраїнський національний технічний університет
Дроздов Сергій Ігоревич
Сертифікат: 514B5C86A1E5DA11040000003CD63700C032FE05
Дійсний з 29.07.2026 13:27:11 по 29.07.2027 13:27:10

Центральноукраїнський національний технічний університет
Факультет *Механіко-технологічний*
Кафедра *Кібербезпеки та програмного забезпечення*
Рівень вищої освіти *магістр*
Галузь знань _F_____*"*Інформаційні технології*"*_____
Спеціальність F7 *"*Комп'ютерна інженерія*"*
Освітньо-професійна (освітньо-наукова) програма _*"*Комп'ютерна інженерія*"*_

ЗАТВЕРДЖУЮ
Завідувач кафедри
д.т.н., проф.
_____ Олексій СМІРНОВ
« 1 » вересня 2026 року

ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ДРУГИМ (МАГІСТЕРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Дроздову Сергію Ігоревичу
(прізвище, ім'я, по батькові)

1. Тема роботи *Дослідження та програмна реалізація системи клієнтського модуля ВКЗ з використанням технології WebRTC та алгоритмів адаптивного управління якістю потокового відео*

2. Керівник роботи *Усік Павло Сергійович, доктор філософії (PhD)*
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 41-13 від 17.08.2026 року

3. Строк подання студентом роботи до захисту *2.12.2026 р.*

4. Мета та завдання випускної кваліфікаційної роботи*: Метою розробки є дослідження та програмна реалізація системи клієнтського модуля ВКЗ з використанням технології WebRTC та алгоритмів адаптивного управління якістю потокового відео*

5. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Призначення та область використання.	*5. Впровадження системи в промислову експлуатацію*
2. Перегляд аналогічних існуючих систем.	*6. Наукова новизна.*
3. Опис і обґрунтування проектних рішень.	*7. Заходи з охорони праці та техніки безпеки.*
4. Етапи програмування системи.	*8. Висновки.*

6. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Наукова новизна	*1 аркуш*
Структурна схема системи	*1 аркуш*
Функціональна схема системи	*1 аркуш*
Діаграма процесів	*1 аркуш*
Блок-схема алгоритму роботи додатку	*2 аркуша*

7. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	Марченко К.М., к.т.н., доцент	06.10.2026	16.11.2026

8. Дата видачі завдання « 1 » вересня 2026 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.10.2026 р.	
2.	Постановка задачі, оформлення ТЗ	15.10.2026 р.	
3.	Розробка моделі компонента	20.10.2026 р.	
4.	Розробка структур даних	25.10.2026 р.	
5.	Розробка алгоритмів системи	30.10.2026 р.	
6.	Програмування алгоритмів	10.11.2026 р.	
7.	Розрахунки з охорони праці та техніки безпеки	15.11.2026 р.	
8.	Оформлення ПЗ	17.11.2026 р.	
9.	Попередній захист роботи	2.12.2026 р.	

Дата видачі завдання
« 1 » вересня 2026 р.

Підпис керівника

_____ _____
(прізвище та ініціали)

Завдання прийнято до виконання
« 1 » вересня 2026 р.

Підпис здобувача

_____ _____
(прізвище та ініціали)

АНОТАЦІЯ

Дроздов С.І. Дослідження та програмна реалізація системи клієнтського модуля ВКЗ з використанням технології WebRTC та алгоритмів адаптивного управління якістю потокового відео. F7 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2026.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи клієнтського модуля ВКЗ з використанням технології WebRTC та алгоритмів адаптивного управління якістю потокового відео.

Метою розробки є дослідження та програмна реалізація системи клієнтського модуля ВКЗ з використанням технології WebRTC та алгоритмів адаптивного управління якістю потокового відео.

Об'єктом дослідження є процес клієнтського модуля ВКЗ з використанням технології WebRTC та алгоритмів адаптивного управління якістю потокового відео.

Предметом дослідження є методи клієнтського модуля ВКЗ з використанням технології WebRTC та алгоритмів адаптивного управління якістю потокового відео.

Методи дослідження базуються на методах теорії кодування, методах комп'ютерних мереж, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи клієнтського модуля ВКЗ з використанням технології WebRTC та алгоритмів адаптивного управління якістю потокового відео.

В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Розроблено зручний інтерфейс користувача. Наведені інструкції по роботі з програмними засобами.

Програма може використовуватися на ПЕОМ з ОС Windows 10/11.

Програму розроблено в середовищі Visual C#.

Ключові слова: комп'ютерна інженерія, WebRTC, відео-конференц зв'язок

ABSTRACT

Drozdov S.I. Research and software implementation of a video conferencing client module system using WebRTC technology and adaptive video streaming quality control algorithms. F7 Computer Engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2026.

This master's-level graduation thesis presents the development of software designed for a video conferencing client module system utilizing WebRTC technology and adaptive video streaming quality control algorithms.

The objective of the project is to research and implement a video conferencing client module system using WebRTC technology and adaptive video streaming quality control algorithms.

The object of the study is the process of the video conferencing client module utilizing WebRTC technology and adaptive video streaming quality control algorithms.

The subject of the study comprises the methods employed by the video conferencing client module using WebRTC technology and adaptive video streaming quality control algorithms.

The research methodology relies on coding theory, computer networking techniques, mathematical statistics, and software development practices.

The outcome of the work is the software implementation of a video conferencing client module system using WebRTC technology and adaptive video streaming quality control algorithms.

During the development of the software model, an analysis of existing hardware and software solutions was conducted. All components of the developed software are described in detail.

A user-friendly interface has been developed, and instructions for using the software are provided.

The program is designed for use on PCs running Windows 10/11. The program was developed in the Visual C# environment.

Keywords: computer engineering, WebRTC, video conferencing

ВІДОМІСТЬ ПРОЕКТУ

		Текстові документи			
1	А4	Пояснювальна записка	111	–	
2	А4	Робоча програма	41	–	Додаток А
		Графічні документи			
3	А4	Наукова новизна	1	–	
4	А4	Структурна схема	1	–	
5	А4	Функціональна схема роботи системи	1	–	
6	А4	Діаграма взаємодії процесів	1	–	
7	А4	Блок-схема основної програми	1	–	
8	А4	Блок-схема роботи підпрограми	1	–	

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ 9
ВСТУП..... 10
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ 13
1.1 Призначення системи..... 13
1.2 Область застосування 13
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ 16
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти..... 16
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування 36
2.3 Розгорнута постановка завдання 37
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ 39
3.1 Опис функціонування системи 39
3.2 Розробка структурної схеми..... 43
3.3 Розробка функціональної схеми 55
3.4 Розробка діаграми процесів 58
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ 60
4.1 Розробка блок-схем та опис алгоритмів функціонування системи 60
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ..... 87
6 НАУКОВА НОВИЗНА 92

7 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ 93
7.1 Вступ 93
7.2 Шкідливі і небезпечні фактори при роботі з комп’ютером 94
7.3 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста ... 95
7.4 Розробка заходів з умов поліпшення охорони праці 98
7.5 Розрахункова частина 99
8 ОСНОВНІ ВИСНОВКИ 102
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 104

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

ДКП	–	дискретне косинусне перетворення
ПЗ	–	програмне забезпечення
ЛОМ	–	локальна обчислювальна мережа
CCTV	–	система замкнутого телебачення
IP	–	інтернет протокол
USB	–	універсальна серійна шина

ВСТУП

Актуальність теми. Якщо ви коли-небудь брали участь у дзвінках Google Meet, не завантажуючи нічого, або користувалися чатом підтримки клієнтів у браузері, який безпосередньо обробляв голосовий та відеозв'язок, ви вже користувалися WebRTC. Ви просто не знали, що в нього є назва.

WebRTC розшифровується як Web Real-Time Communication (веб-зв'язок у реальному часі). Це технологія, яка дозволяє веб-додаткам і сайтам захоплювати та транслювати аудіо- та відеомедіа, а також обмінюватися довільними даними між браузерами без посередника. Простіше кажучи: це дозволяє двом пристроям взаємодіяти один з одним безпосередньо через браузер, без плагіна, завантаження додатків та внутрішнє програмне забезпечення.

Спочатку розроблений Google, а згодом стандартизований Консорціумом Всесвітньої павутини (W3C) та Робочою групою з інтернет-інженерії (IETF), WebRTC став фактичним стандартом для зв'язку на основі браузера.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи клієнтського модуля ВКЗ з використанням технології WebRTC та алгоритмів адаптивного управління якістю потокового відео.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

– Огляд існуючих систем клієнтського модуля ВКЗ з використанням технології WebRTC та алгоритмів адаптивного управління якістю потокового відео.

– Дослідження системи клієнтського модуля ВКЗ з використанням технології WebRTC та алгоритмів адаптивного управління якістю потокового відео.

– Програмна реалізація системи клієнтського модуля ВКЗ з використанням технології WebRTC та алгоритмів адаптивного управління якістю потокового відео.

Об'єктом дослідження є процес клієнтського модуля ВКЗ з використанням технології WebRTC та алгоритмів адаптивного управління якістю потокового відео.

Предметом дослідження є методи клієнтського модуля ВКЗ з використанням технології WebRTC та алгоритмів адаптивного управління якістю потокового відео.

Методи дослідження базуються на методах теорії кодування, методах комп'ютерних мереж, методах математичної статистики, методах розробки програмного забезпечення.

Наукова новизна отриманих результатів. У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

– Удосконалено метод клієнтського модуля ВКЗ з використанням технології WebRTC та алгоритмів адаптивного управління якістю потокового відео.

– Розроблено вітчизняний продукт клієнтського модуля ВКЗ з використанням технології WebRTC та алгоритмів адаптивного управління якістю потокового відео, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі клієнтського модуля ВКЗ з використанням технології WebRTC та алгоритмів адаптивного управління якістю потокового відео.

Достовірність наукових результатів підтверджена теоретичними викладеннями, даними комп'ютерного моделювання, коректними дослідженнями параметрів на функціонуючій обчислювальній мережі, а також відповідністю отриманих результатів окремим результатам, наведеним у науковій літературі.

Робота апробована на LVIII науково-технічній конференції здобувачів вищої освіти «Наука-виробництву», основні положення випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти надруковані у статті збірника праць молодих науковців ЦНТУ, випуск №17.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи клієнтського модуля ВКЗ з використанням технології WebRTC та алгоритмів адаптивного управління якістю потокового відео, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

До появи WebRTC спілкування в реальному часі в Інтернеті було болісним. Потрібен був Flash. Потрібен був плагін. Потрібен був пропрієтарний клієнт, який ламався кожні кілька місяців. Розробники це ненавиділи. Користувачі ненавиділи це ще більше.

Google придбав цю технологію у 2010 році та розкрив її ядро у 2011 році. W3C та IETF підхопили її та розпочали процес стандартизації. За оцінками, до 2016 року було встановлено близько 2 мільярдів браузерів, які могли працювати з WebRTC. З точки зору трафіку, щотижня через зв'язок між браузерами передавалось понад мільярд хвилин і 500 терабайтів даних.

Перенесемося у 2026 рік, і WebRTC не просто зрілий, це інфраструктурний рівень сучасного інтернету в реальному часі. Відеоконференції, голосові агенти зі штучним інтелектом, телемедицина, прямі трансляції, автоматизація підтримки клієнтів, інструменти для спільної роботи – все це працює на ньому.

1.2 Область застосування

Розглянемо варіанти використання WebRTC у 2026 році. Технологія сама по собі не цікава. Важливо те, де вона проявляється.

Відеоконференції

Очевидний приклад. Google Meet, Microsoft Teams, Zoom (у браузері), Discord – усі вони або використовують WebRTC безпосередньо, або мають його вбудований на транспортному рівні. WebRTC забезпечує одноранговий зв'язок без зовнішніх плагінів, дозволяючи двом користувачам встановити безпечне та

пряме з'єднання, зменшуючи залежність від централізованих серверів, знижуючи використання пропускної здатності та підвищуючи ефективність.

Голосові агенти зі штучним інтелектом

Саме тут у 2026 році все стає по-справжньому захопливим. Голосовим агентам на базі штучного інтелекту потрібна наднизька затримка, щоб відчуватися природно. Навіть якщо між промовою користувача та відповіддю агента є затримка в 300 мс, взаємодія переривається. WebRTC вирішує цю проблему. Amazon Bedrock AgentCore Runtime додав підтримку WebRTC у березні 2026 року для двонаправленої потокової передачі в режимі реального часу між клієнтами та агентами, що забезпечує природний розмовний досвід у реальному часі.

У RTC League платформа голосових агентів TelEcho зі штучним інтелектом побудована саме на цьому фундаменті. WebRTC забезпечує транспортний рівень, який робить можливою затримку менше 200 мс у омніканальних розгортаннях, включаючи веб, WhatsApp, GSM та SIP/PSTN. Коли клієнт телефонує, а агент зі штучним інтелектом відповідає зі швидкістю, близькою до людської, WebRTC виконує всю роботу під ним.

Телемедицина

Безпечні, зашифровані відеоконсультації, що працюють безпосередньо у браузері. Жодних завантажень додатків, жодних дзвінків до IT-служби, жодних проблем із налаштуванням. Пацієнти підключаються з будь-якого браузера, лікарі приймають їх за лічені секунди. Якщо постачальник послуг телемедицини хоче провести зашифровану відеоконсультацію, WebRTC забезпечує транспортування.

Корпоративна телефонія

WebRTC перейшов від «приємного доповнення» до «очікуваного». У 2026 році справжнє питання для корпоративних команд полягає не в тому, чи є життєздатним браузерний зв'язок, а в тому, як розгорнути його безпечним, зручним та сумісним з існуючими системами SIP/PBX способом.

Пряма трансляція та інтерактивна трансляція

Традиційні протоколи потокового передавання, такі як HLS або DASH, вводять затримки 10-30 секунд. WebRTC зменшує цю затримку до менш ніж секунди. Це важливо для таких сценаріїв, як прямі трансляції спортивних та ігрових трансляцій, платформи фінансової торгівлі, онлайн-ставки та інструменти для спільної роботи, де затримки повинні бути менше секунди.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи клієнтського модуля ВКЗ з використанням технології WebRTC та алгоритмів адаптивного управління якістю потокового відео, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

2.1 Огляд існуючих систем, технологій, архітектур, програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти

Програмне забезпечення для відеоконференцій дозволяє командам проводити відеозустрічі в реальному часі, ділитися екранами, обмінюватися повідомленнями, записувати сеанси та співпрацювати з настільних комп'ютерів, мобільних пристроїв або браузерних програм.

Найкраща платформа залежить від того, як насправді відбуваються зустрічі в організації. Компанія, яка часто запрошує клієнтів, може надавати пріоритет простому доступу через браузер. Користувач Microsoft 365 або Google Workspace може надавати перевагу конференціям, вбудованим у існуюче середовище продуктивності. Підприємствам із налагодженими системами конференц-залів може знадобитися сумісність SIP або H.323, тоді як регульовані організації можуть вимагати, щоб інфраструктура та записи зустрічей залишалися під їхнім власним контролем.

У цьому розділі порівнюється дев'ять платформ для відеоконференцій за практичними критеріями покупки, включаючи досвід проведення зустрічей, доступ для гостей, розгортання, запис, можливості штучного інтелекту, адміністрування, сумісність, інтеграції та контроль інфраструктури.

Немає універсального переможця. Zoom – це практичний варіант для частих зовнішніх зустрічей, Microsoft Teams підходить для середовищ Microsoft 365, Google Meet природно працює з Google Workspace, Webex підходить для підприємств, орієнтованих на Cisco, а TrueConf виділяється, коли організації потрібні відеоконференції, обмін повідомленнями, сумісність кімнатних систем та обробка даних за допомогою штучного інтелекту на інфраструктурі, яку вона контролює.

Zoom

Найкраще підходить для: зовнішніх зустрічей, дзвінків клієнтам, вебінарів та організацій, які надають пріоритет простому гостьовому доступу.

Розгортання: Хмара під управлінням постачальника.

Безкоштовний варіант: доступний з обмеженнями щодо кількості учасників та тривалості зустрічі.

Доступ через браузер: Так.

Мобільні додатки: iOS та Android.

Запис: Локальні та хмарні опції залежно від плану.

Штучний інтелект: Zoom AI Companion на підтримуваних планах.



Рисунок 2.1 – Інтерфейс користувача Zoom

Zoom залишається однією з найбільш впізнаваних платформ для відеоконференцій і особливо практичний, коли зустрічі регулярно проводяться за участю людей поза організацією.

Учасники зазвичай можуть приєднатися через посилання на зустріч, не

вивчаючи попередньо повного середовища для співпраці. Це зменшує труднощі під час дзвінків клієнтам, співбесід з наймом, зустрічей з постачальниками, консультацій та інших зовнішніх робочих процесів.

Zoom також вийшов за межі окремих зустрічей завдяки Zoom Workplace, Zoom Phone, Zoom Rooms та Zoom Events. Таким чином, організації можуть розширити ту саму екосистему на обмін повідомленнями, телефонію, кімнати для переговорів, вебінари та масштабніші заходи.

AI Companion додає короткий зміст зустрічей, допомогу з відновленням, запропоновані наступні кроки та функції подальшого спостереження для відповідних планів.

Переваги:

– Приєднання гостей з низьким рівнем Інтернету.

– Можливості для проведення сесійних конференцій та вебінарів.

– Широка підтримка комп'ютерів, браузерів та мобільних пристроїв.

– Велика екосистема обладнання для конференц-залів.

– Підсумки зустрічей та подальші дії за допомогою штучного інтелекту.

– Широкий ринок інтеграції зі сторонніми розробниками.

Основні компроміси:

– Zoom залишається переважно хмарним сервісом, керованим постачальниками. Організаціям, яким необхідно самостійно керувати основною інфраструктурою конференцій, знадобиться інша модель розгортання.

– Вартість підписки також слід розраховувати з урахуванням фактичної кількості хостів, кімнат, вимог до сховища, вебінарів та доповнень, а не на основі найнижчого заявленого плану.

Оберіть Zoom, коли: основними вимогами є зовнішня участь та легкий доступ до зустрічей.

Перегляньте Zoom, коли: Інфраструктура зустрічей має залишатися під контролем клієнта або працювати в обмеженому середовищі.

Microsoft Teams

Найкраще підходить для: організацій, стандартизованих для Microsoft

365.

Розгортання: хмара під управлінням Microsoft.

Безкоштовний варіант: доступний з обмеженнями; бізнес-функціональність залежить від плану та регіону.

Доступ через браузер: Так.

Мобільні додатки: iOS та Android.

Запис: Доступно для підтримуваних бізнес-планів.

Штучний інтелект: Функціональність Microsoft Copilot та Intelligent Recap залежить від ліцензування.

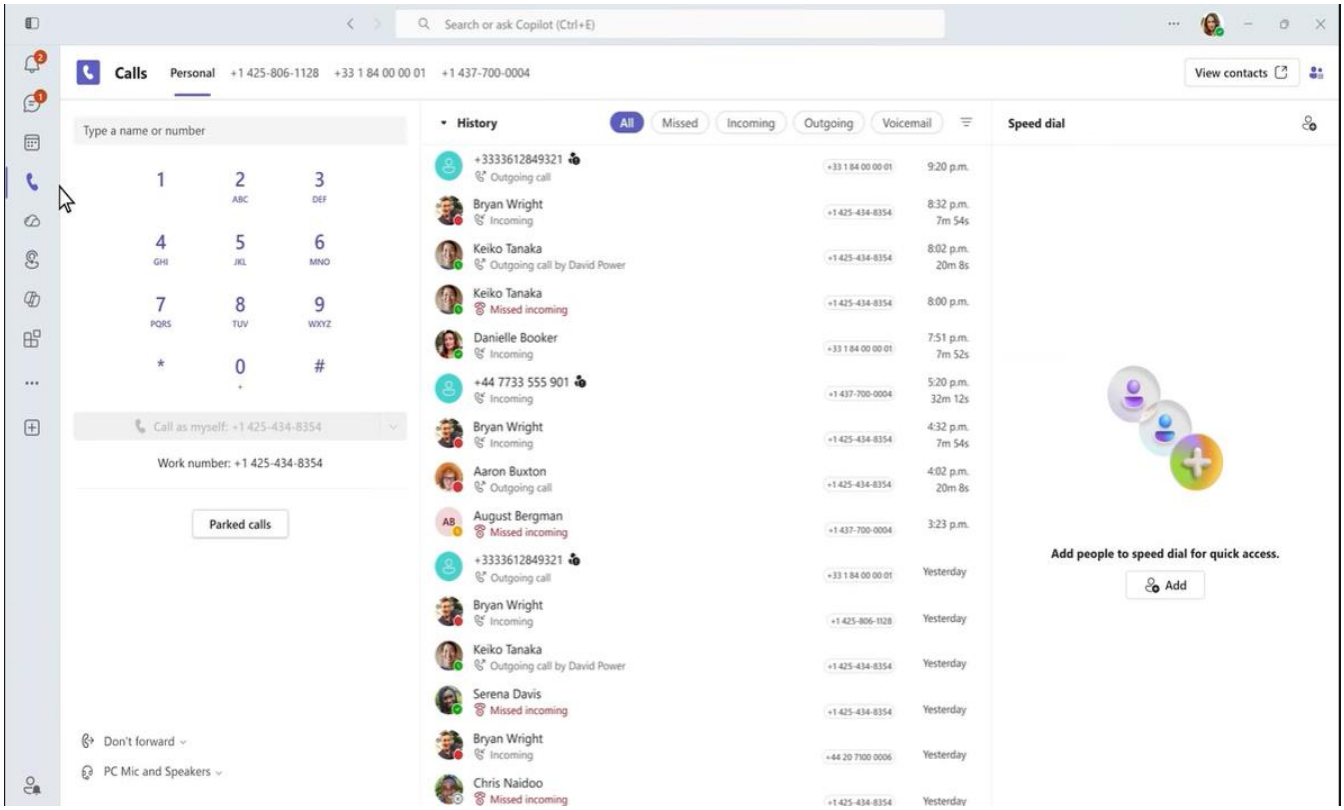


Рисунок 2.2 – Інтерфейс користувача Microsoft Teams

Microsoft Teams поєднує відеоконференції з чатом, каналами, календарями, документами, варіантами викликів і програмами Microsoft 365.

Тому його головна перевага полягає в тому, що відеоконференції не є ізольованими. Teams найбільш корисний, коли співробітники вже проводять свій

робочий день в Outlook, SharePoint, OneDrive, Word, Excel та інших сервісах Microsoft.

Розмова може перейти з чату на зустріч, а потім у спільне редагування документів, не виходячи з ширшого середовища Microsoft.

Copilot може підсумовувати обговорення, визначати рішення та пункти дій, допомагати користувачам відстежувати пропущені зустрічі та пов'язувати інформацію про зустрічі з іншими робочими процесами Microsoft 365.

Переваги:

– Нативна інтеграція з Microsoft 365.

– Зустрічі, чат, файли та канали в одному середовищі.

– Екосистема кімнат Teams.

– Ідентифікація та адміністрування підприємства.

– Запис і транскрипція на підтримуваних планах.

– Аналітика зустрічей на основі Copilot.

Основні компроміси:

– Teams забезпечує найбільшу цінність, коли Microsoft 365 вже є основною екосистемою продуктивності організації. Компанія, яка купує лише сервіс відеозустрічей, може вважати ширшу модель ліцензування та адміністрування надмірно складною.

– Основний сервіс також працює в хмарі, а не розміщується на клієнтському хостингу.

Оберіть Teams, якщо: Microsoft 365 вже надає ідентифікаційні дані, файли, календарі та можливості для співпраці.

Перегляньте Teams, коли: вам потрібна окрема платформа для зустрічей або інфраструктура конференцій, що керується клієнтом.

TrueConf Server

Найкраще підходить для: організацій, яким потрібні відеоконференції, обмін повідомленнями, сумісність систем кімнат та аналітика зустрічей на інфраструктурі, яку вони контролюють.

Розгортання: локальна або приватна інфраструктура, що керується

клієнтом, залежно від випуску та конфігурації.

Безкоштовний варіант: доступний TrueConf Server Free.

Доступ через браузер: Так.

Мобільні додатки: доступні на основних платформах для комп’ютерів та мобільних пристроїв.

Запис: варіанти на стороні клієнта та на стороні сервера.

Штучний інтелект: TrueConf AI Server забезпечує локальне розгортання аналітичних даних для проведення зустрічей.

Взаємодія: SIP/H.323 та пов'язані з ним корпоративні інтеграції є важливою частиною платформи.

TrueConf відрізняється від більшості хмарних продуктів у порівнянні тим, що організації можуть самостійно керувати основною системою конференц-зв'язку.

Платформа поєднує відеозустрічі з постійним обміном повідомленнями один на один та в групах, файлами, інформацією про присутність, участю браузера, мобільними додатками, каталогами та централізованим адмініструванням. Залежно від обраної версії та ліцензії, розгортання можуть підтримувати великі конференції з кількістю учасників до 2000.

Для організацій, які вже підтримують системи конференц-залів SIP або H.323 та інфраструктуру АТС, TrueConf може підключити існуюче обладнання, не вимагаючи повної заміни апаратного забезпечення кімнати. Інтеграція з Active Directory та LDAP також може підключити комунікаційну платформу до існуючого середовища корпоративної ідентифікації.

Ще однією відмінністю є TrueConf AI Server. Транскрипція зустрічей, ідентифікація доповідача, конспекти, протоколи та обробка записаного аудіо чи відео можуть бути розгорнуті як частина середовища, контрольованого клієнтом, а не покладатися виключно на публічний сервіс штучного інтелекту. В оригінальній статті також серед можливостей, пов'язаних зі штучним інтелектом, виділено шумозаглушення та візуальні ефекти.

Таке поєднання є найбільш актуальним, коли самостійне розміщення є не

просто уподобанням, а частиною інфраструктурної моделі організації. Урядовим, медичним, фінансовим, промисловим, юридичним та іншим організаціям може знадобитися жорсткіший контроль над інфраструктурою зустрічей, записами, інтеграцією ідентифікаційних даних або мережевими залежностями.

Щоб побачити, як повне локальне розгортання виглядає на практиці, перегляньте сторінку TrueConf про безпечний зв'язок – вона охоплює локальну мережу, VPN та роботу в ізоляції без впливу хмарних технологій.

Переваги:

– Розгортання, кероване клієнтом.

– Вбудовані відеоконференції та постійний обмін повідомленнями.

– Підтримка великих зустрічей.

– Участь у браузері та на мобільних пристроях.

– Інтеграція з Active Directory та LDAP.

– Взаємодія SIP/H.323.

– Запис на стороні клієнта та сервера.

– Локальна обробка ШІ за допомогою TrueConf AI Server.

– Можливість зберегти існуючі інвестиції в кімнатні системи.

– Модель ліцензування, яка відрізняється від чистих підписок SaaS на користувача.

Основні компроміси:

Запуск платформи внутрішніми силами означає, що організація повинна керувати серверною інфраструктурою, оновленнями, моніторингом, резервним копіюванням, резервуванням, конфігурацією мережі та підтримкою.

Оберіть TrueConf, коли: зустрічі, обмін повідомленнями, записи, корпоративні каталоги, сумісність кімнат та обробка даних за допомогою штучного інтелекту повинні працювати в межах інфраструктури, керованої клієнтом.

Перегляньте TrueConf, коли: Основною вимогою є прості зустрічі в публічній хмарі з мінімальним адмініструванням інфраструктури.

4. Cisco Webex

Найкраще підходить для: великих підприємств, середовищ Cisco, регульованих організацій та керованих розгортань конференц-залів.

Розгортання: переважно хмарне, з гібридною та іншими варіантами корпоративної архітектури залежно від обраного сервісу.

Безкоштовний варіант: доступний з обмеженнями.

Доступ через браузер: Так.

Мобільні додатки: iOS та Android.

Запис: Доступно залежно від послуги та плану.

Штучний інтелект: Cisco AI Assistant та пов’язана з ним аналітика зустрічей.

Webex має довгу історію в корпоративних конференціях і є особливо актуальним там, де програмне забезпечення для проведення зустрічей має співіснувати з мережевою, телефонною та конференц-інфраструктурою Cisco.

Організації, які вже використовують кінцеві точки Cisco, можуть інтегрувати наради в існуюче апаратне та адміністративне середовище, замість того, щоб розглядати конференції як повністю окрему систему.

Webex також надає корпоративні засоби керування ролями нарад, записом, дозволами учасників, пристроями кімнат та більшими форматами нарад.

Cisco AI Assistant може допомогти підбити підсумки зустрічей, визначити пункти завдань, відповідати на запитання, пов’язані з зустріччю, та допомогти учасникам надолужити згаяне.

Переваги:

– Адміністрування підприємства.
– Екосистема кімнатних пристроїв Cisco.
– Варіанти зустрічей, дзвінків та обміну повідомленнями.
– Можливості проведення вебінарів та заходів.
– Робочі процеси зустрічей із підтримкою штучного інтелекту.
– Чудова відповідність існуючій інфраструктурі Cisco.

Основні компроміси:

– Широкий спектр послуг Cisco для співпраці може зробити ліцензування, налаштування та адаптацію більш складними, ніж просто послуга, що передбачає проведення лише зустрічей.

– Покупцям також слід розрізняти хмарні, гібридні, урядові, дзвінкові, кімнатні та приватні архітектурні продукти, а не вважати кожне розгортання Webex технічно ідентичним.

Оберіть Webex, коли: інфраструктура Cisco та централізоване адміністрування підприємства є центральними вимогами.

Перегляньте варіант Webex, коли: Організації потрібен легкий додаток для проведення зустрічей з мінімальною конфігурацією.

Google Meet

Найкраще підходить для: організацій Google Workspace, освітніх закладів та команд, які надають перевагу зустрічам у браузері.

Розгортання: хмара під управлінням Google.

Безкоштовний варіант: доступний з обмеженнями щодо зустрічей.

Доступ через браузер: Так.

Мобільні додатки: iOS та Android.

Запис: Доступно для відповідних планів Workspace.

Штучний інтелект: Функції для проведення зустрічей на базі Gemini залежно від версії Workspace.

Google Meet наголошує на доступі через браузер і тісній інтеграції з Google Workspace.

Зустрічі можна створювати через Календар Google, запускати з Gmail та підключати до контенту, що зберігається на Диску та в Документах. Для організацій, які вже стандартизували Workspace, це зменшує потребу у впровадженні додаткового середовища для співпраці.

Учасники зазвичай можуть приєднатися з підтримуваного браузера без встановлення програми для комп’ютера, що корисно для організацій, які часто запрошують зовнішніх гостей.

Функції Gemini додають нотатки до зустрічей, підсумки та інші робочі процеси за допомогою штучного інтелекту в підтримуваних планах.

Переваги:

– Доступ до зустрічей через браузер.

– Інтеграція Календаря Google та Gmail.

– Мобільні додатки.

– Субтитри та функції спеціальних можливостей.

– Запис на підтримуваних планах Workspace.

– Інтеграція Gemini.

Основні компроміси:

Доступність функцій значно відрізняється між версіями Workspace, особливо для запису, адміністрування, штучного інтелекту, відвідуваності та більших форматів зустрічей.

Google Meet не пропонує клієнтом самостійно розміщений сервер для зустрічей.

Оберіть Google Meet, коли: Google Workspace вже є основним середовищем для продуктивності.

Перегляньте Google Meet, коли: потрібне володіння інфраструктурою, розширена сумісність кімнат або ізольована робота.

Jitsi Meet

Найкраще підходить для: розгортання програмного забезпечення з відкритим кодом, технічних команд, освіти та організацій, які хочуть контролювати саме програмне забезпечення для конференцій.

Розгортання: державна служба або власне розміщення.

Безкоштовний варіант: Так.

Доступ через браузер: Так.

Мобільні додатки: Доступні.

Запис: залежить від конфігурації та розгортання.

Штучний інтелект: головним чином через інтеграції та користувацькі робочі процеси, а не стандартизований комерційний пакет ШІ.

Jitsi Meet – один із найвідоміших проектів з відкритого коду для проведення конференцій.

Організації можуть використовувати публічний сервіс Jitsi для проведення простих зустрічей або самостійно розгортати програмне забезпечення, коли контроль інфраструктури та налаштування є важливішими.

Його модель з відкритим кодом надає технічним командам гнучкість щодо автентифікації, брендингу, інфраструктури, інтеграцій та допоміжних послуг.

Переваги:

– Програмне забезпечення з відкритим вихідним кодом.

– Самостійний хостинг.

– Приєднання до браузера.

– Відсутність звичайної ліцензії на робоче місце для ядра з відкритим кодом.

– Налаштоване розгортання.

– Можливість створювати власні інтеграції та робочі процеси обробки.

Основні компроміси:

– Самостійне розміщення Jitsi вимагає технічних знань. Розгортання на корпоративному рівні потребує планування автентифікації, масштабування, запису, моніторингу, оновлень, резервування та безпеки.

– Інтегрований набір функцій бізнес-комунікації також вужчий, ніж у комерційного пакету, що поєднує зустрічі, постійний обмін повідомленнями, каталоги, інтеграцію кімнат та підтримку постачальників.

Оберіть Jitsi, коли: Відкритий код та гнучкість розгортання важливіші за комплексне адміністрування підприємства.

Перегляньте Jitsi, коли: Вам потрібне комерційно підтримуване інтегроване комунікаційне середовище без складання додаткових компонентів.

8×8

Найкраще підходить для: організацій, які купують відеоконференції як частину ширшого середовища UCaaS або контакт-центру.

Розгортання: Хмара під управлінням постачальника.

Безкоштовний варіант: залежить від продукту та плану.

Доступ через браузер: Так.

Запис: залежить від плану.

Штучний інтелект: доступний у ширшій екосистемі комунікацій 8×8.

8×8 розглядає відео як один із компонентів більшого стеку бізнес-комунікацій.

Замість того, щоб просити клієнтів самостійно купувати зустрічі, ширше середовище 8×8 може об'єднати бізнес-телефонію, відеоконференції, обмін повідомленнями, послуги контакт-центру, адміністрування та аналітику.

Ця модель корисна для організацій, які намагаються об'єднати кількох постачальників комунікаційних послуг.

Переваги:

– Відео інтегровано з ширшим середовищем UCaaS.
– Бізнес-телефонія.
– Варіанти контакт-центру.
– Єдине адміністрування.
– Застосування функціональності ШІ в ширшій екосистемі.

Основні компроміси:

Ширший стек менш актуальний, коли організації потрібні лише відеозустрічі.

Покупцям, які зосереджені виключно на конференц-зв'язках, слід порівняти можливості 8×8 для проведення зустрічей зі спеціалізованими відеоплатформами, а не оцінювати його лише через загальний комунікаційний пакет.

Оберіть 8×8, коли: Відео є частиною ширшого проекту консолідації телефонії та контакт-центру.

Перегляньте 8×8, коли: Вимога полягає в окремому найкращому продукті для проведення зустрічей.

Рехір

Найкраще підходить для: сумісності підприємств, змішаних середовищ для зустрічей, інфраструктури конференц-кімнат та розгортання приватного відео.

Розгортання: Хмарне, приватне та контрольоване клієнтом розгортання залежно від продукту.

Доступ через браузер: Так.

Запис: Залежить від архітектури.

Штучний інтелект: Приватні можливості штучного інтелекту для вибраних робочих процесів мовлення та перекладу.

SIP/H.323: Центральна перевага.

Рехір вирішує іншу проблему, ніж більшість програм для загальних зборів.

Підприємство може вже використовувати Microsoft Teams для співробітників, водночас підтримуючи кінцеві точки конференц-залів Cisco або Poly, обладнання SIP, системи H.323 та кілька поколінь інфраструктури кімнат.

Рехір може виступати в ролі рівня сумісності між цими середовищами, а не вимагати від усіх користувачів та кімнат переходу на єдину платформу.

Варіанти приватного розгортання також роблять його актуальним там, де трафік зустрічей неможливо просто направити через звичайний публічний хмарний сервіс.

Рехір Private AI додає можливості розпізнавання та перекладу мовлення для вибраних приватних або суверенних середовищ. Його сфера застосування вужча, ніж у загальних помічників зі складання підсумків зустрічей, але архітектура може мати більше значення, ніж обсяг функцій, коли потрібна локальна обробка.

Переваги:

– Взаємодія SIP/H.323.
– Підтримка змішаних приміщень.
– Інтеграція з сучасними платформами для проведення зустрічей.

– Варіанти приватного розгортання.

– Можливість продовжити термін служби існуючого обладнання приміщення.

– Можливості приватного мовлення та перекладу в підтримуваних сценаріях.

Основні компроміси:

– Рехір вимагає більше корпоративних IT-експертиз, ніж типовий додаток для проведення зустрічей, що працює в першу чергу через браузер.

– Тому це менш актуально для невеликих організацій, яким просто потрібні прості онлайн-зустрічі.

Оберіть Рехір, коли: Існуючі платформи для проведення зустрічей та системи кімнат повинні бути сумісними.

Перегляньте Рехір, коли: Вам потрібен простий автономний додаток для проведення зустрічей, а не окремий рівень інфраструктури.

Secumeet

Найкраще підходить для: організацій, що оцінюють відеозв'язок для обмежених, регульованих або іншим чином жорстко контрольованих середовищ.

Розгортання: Розташування навколо контрольованої інфраструктури.

Безкоштовний варіант: Так

Доступ через браузер та мобільний пристрій: Так.

Запис: Так.

Штучний інтелект: Наразі не є центральною частиною позиціонування продукту.

Secumeet – це більш спеціалізований продукт, ніж універсальні платформи для конференцій, такі як Zoom або Teams.

Його позиціонування є актуальним для організацій, чий короткий список починається з питання інфраструктури: чи може комунікація залишатися в межах затвердженого середовища, а не оброблятися через публічний багатокористувацький сервіс конференцій.

Можливі сценарії включають конфіденційне спілкування керівництва,

обмежені мережі, регульовані робочі навантаження та організації зі суворими вимогами до розділення даних.

Компроміс полягає в тому, що загальнодоступна технічна інформація наразі менш обширна, ніж та, що доступна для великих постачальників послуг конференц-спілкування.

Перед закупівлею покупці повинні запросити документацію, що охоплює:

– архітектура розгортання;
– вимоги до сервера;
– механізми шифрування;
– робота з ключами;
– фіксація поведінки;
– залежності від браузера та мобільного пристрою;
– інтеграція з каталогами;
– зовнішні мережеві залежності;
– висока доступність;
– офлайн-ліцензування;
– функції аудиту;
– умови підтримки та оновлення.

Заяви безпеки, такі як наскрізне шифрування або повністю ізольована робота, повинні бути пов'язані з конкретною поточною технічною документацією та перевірені на відповідність запланованому розгортанню.

Оберіть Secumeet, коли: Обмежена інфраструктура є основною вимогою, і покупець може перевірити архітектуру шляхом технічних закупівель.

Перегляньте Secumeet, коли: Вам потрібна широка екосистема публічної хмари, обширна публічна документація або великий ринок інтеграції зі сторонніми розробниками.

Хмарні та власноруч організовані відеоконференції

Модель розгортання може виключати або відкидати продукти до порівняння окремих функцій. Жодна з моделей не є кращою за своєю суттю.

Конференц-спілкування в публічній хмарі мінімізує управління інфраструктурою та може бути найпростішим вибором для загальних ділових зустрічей.

Самостійний хостинг забезпечує більший контроль над серверами, сховищем, розміщенням у мережі, записами, інтеграціями та потенційно обробкою даних за допомогою штучного інтелекту. Цей контроль також передає операційну відповідальність клієнту.

Ключові висновки з порівняння

Дев'ять продуктів поділяються на кілька окремих груп, а не на одну єдину категорію.

Загального призначення хмарні конференції: Zoom, Microsoft Teams, Google Meet та Webex охоплюють більшість поширених робочих процесів ділових зустрічей, мінімізуючи управління інфраструктурою.

Інтегровані комунікації: 8×8 розглядає відео як частину ширшого середовища UCaaS та контакт-центру.

Самостійне розміщення конференцій: TrueConf та Jitsi дозволяють організаціям самостійно керувати основною інфраструктурою конференцій, але вони представляють різні моделі. Jitsi наголошує на гнучкості відкритого коду, тоді як TrueConf надає ширше комерційно інтегроване середовище, що поєднує відео, обмін повідомленнями, корпоративне адміністрування, сумісність та локально розгортаний штучний інтелект.

Відеоінфраструктура підприємства: Pexip зосереджується на сумісності та змішаних середовищах відеоспостереження.

Зв'язок в обмеженому середовищі: Secumeet займає вужчий сегмент, де архітектура розгортання може бути важливішою, ніж розмір екосистеми інтеграції.

Ось чому вибір програмного забезпечення для відеоконференцій лише на основі підрахунку функцій дає слабкі результати. Різні платформи оптимізовані для різних середовищ.

Як вибрати програмне забезпечення для відеоконференцій

1. Визначте, хто приєднується до зустрічей

Почніть зі справжньої аудиторії:

– працівники;

– клієнти;

– постачальники;

– пацієнти;

– студенти;

– зовнішні партнери;

– публічні учасники;

– обмежені внутрішні групи.

Організаціям з багатьма зовнішніми учасниками слід звернути особливу увагу на вимоги до приєднання через браузер та облікових записів.

2. Визначте формати зустрічей

Зустріч проекту з 10 особами та корпоративний захід з 1000 особами вимагають різних заходів контролю.

Перевірте:

– інтерактивна здатність учасників;

– режими вебінару або трансляції;

– кімнати для групових занять;

– елементи керування модератором;

– запис;

– інтерпретація;

– титри;

– реєстрація;

– взаємодія з аудиторією.

3. Перевірте доступ для гостей

Запитайте:

– Чи потрібен обліковий запис?

– Чи можуть гості приєднатися через браузер?

– Чи потрібна установка?

– Чи можуть господарі контролювати вхід?

– Чи є зали очікування?

– Чи працює приєднання мобільного гостя?

– Чи можуть зовнішні гості отримати доступ до записів або файлів?

Для організацій, орієнтованих на клієнтів, ці деталі можуть мати більше значення, ніж розширені функції штучного інтелекту.

4. Перевірте існуючу екосистему продуктивності

Користувачі Microsoft 365, природно, мають інші пріоритети, ніж користувачі Google Workspace.

Також перегляньте інтеграції з:

– корпоративні календарі;

– Active Directory або LDAP;

– Єдиний доступ;

– CRM;

– кімнатні системи;

– Інфраструктура АТС;

– SIP/H.323;

– системи управління навчанням;

– API.

5. Виберіть прийнятну модель розгортання

Основні моделі включають:

– публічний SaaS;

– приватні середовища, що управляються постачальниками;

– гібридні архітектури;

– приватна хмара;

– самостійно розміщена інфраструктура;

– обмежені або відключені середовища.

Не варто вважати, що кожна платформа, яку називають «приватною» або «гібридною», забезпечує однаковий рівень контролю над клієнтами.

6. Порівняйте адміністрування

У масштабі підприємства адміністративні функції стають такими ж важливими, як і функції, орієнтовані на учасників.

Перевірте:

– налаштування користувачів;

– єдиний доступ;

– Active Directory та LDAP;

– ролі;

– політика проведення зустрічей;

– елементи керування гостями;

– утримання;

– дозволи на запис;

– дані аудиту;

– звітність;

– API;

– моніторинг.

7. Оцінюйте ШІ незалежно від конференцій

Штучний інтелект стає все більш поширеним, але архітектура та ліцензування різняться.

Питання, які слід поставити, включають:

– Чи включено штучний інтелект до плану зустрічі?

– Де обробляється аудіозапис зустрічі?

– Де зберігаються транскрипти?

– Чи можна вимкнути ШІ?

– Чи можуть адміністратори контролювати, які користувачі мають до нього доступ?

– Чи доступні етикетки для динаміків?

– Чи можна редагувати резюме?

– Чи потребує ШІ сторонніх послуг?

– Чи може обробка даних ШІ виконуватися в інфраструктурі,

контрольованій клієнтом?

Хмарні помічники, такі як Zoom AI Companion, Microsoft Copilot та Google Gemini, оптимізовані для інтеграції з відповідними хмарними екосистемами.

TrueConf AI Server представляє іншу архітектуру, оскільки аналітичні засоби для зустрічей можна розгорнути як частину середовища конференцій, керованого клієнтом.

8. Розрахуйте загальну вартість володіння

Найдешевший щомісячний план не обов'язково є найдешевшим розгортанням.

Витрати на хмарні технології можуть включати:

– підписки користувачів або хостів;
– доповнення до вебінарів;
– місткість для великих зустрічей;
– Штучний інтелект;
– зберігання записів;
– телефонні послуги;
– кімнати для переговорів;
– преміум-підтримка.

Витрати на самостійне розміщення можуть включати:

– ліцензії на програмне забезпечення;
– сервери або віртуальна інфраструктура;
– зберігання;
– пропускна здатність мережі;
– резервне копіювання;
– моніторинг;
– адміністрація;
– оновлення;
– підтримка.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Для розробки програмного забезпечення клієнтського модуля відеоконференц-зв'язку (ВКЗ) було обрано мову програмування C# у середовищі .NET. Цей вибір зумовлений високими вимогами до продуктивності, безпеки, ефективного керування пам'яттю та багатопотоковості, що є критично важливими під час обробки потокового аудіо- та відеоконтенту в реальному часі.

C# – це строго типізована об’єктно-орієнтована мова, яка забезпечує зручну розробку високопродуктивних клієнтських додатків, а також гнучку інтеграцію з мережевими протоколами й низькорівневими бібліотеками, задіяними у реалізації стека WebRTC. Завдяки виразному синтаксису, підтримці асинхронного програмування (async/await), делегатів, подій та лямбда-виражень, розробник може ефективно керувати асинхронними мережевими подіями, станами медіапотоків та викликами зворотного зв'язку (callbacks), які виникають під час передачі даних у реальному часі.

Об'єктно-орієнтована архітектура C# (інкапсуляція, спадкування, поліморфізм) дозволяє чітко модулювати архітектуру клієнтського модуля ВКЗ. Це дає змогу відокремити бізнес-логіку інтерфейсу від модулів захоплення медіаданих, кодування, мережевої передачі через WebRTC та компонентів реалізації алгоритмів адаптивного управління якістю потокового відео. Використання властивостей, атрибутів та структур сприяє написанню чистого й підтримуваного коду для обробки телеметричних метрик потоку (таких як втрати пакетів, затримка RTT та джиттер).

Для взаємодії з нативними бібліотеками WebRTC (наприклад, написаними на C/C++) у середовищі C# використовується механізм взаємодії (Interop). Це дозволяє клієнтському модулю ефективно задіювати нативні компоненти для кодування/декодування медіаданих (VP8, VP9, AV1, H.264), керування мережевими з'єднаннями (ICE, STUN, TURN) та забезпечення низької

затримки (ultra-low latency), зберігаючи при цьому переваги безпечного керованого середовища виконання.

Архітектура платформи .NET у контексті клієнтського модуля ВКЗ

Програмне забезпечення клієнтського модуля виконується в середовищі .NET, яке містить віртуальну систему виконання (CLR) та уніфікований набір бібліотек класів.

Вихідний код на C# компілюється в проміжну мову (IL), яка під час запусків виконується за допомогою JIT-компіляції (Just-In-Time) у машинні інструкції. Для систем відеоконференц-зв'язку критично важливим є ефективне управління ресурсами, апаратне прискорення та мінімізація пауз, спричинених збиранням сміття (Garbage Collection). Середовище CLR забезпечує надійне керування пам'яттю та безпеку виконання "керованого коду", що важливо для стабільної роботи медіапакетів під час високого навантаження на мережу.

Завдяки потужній бібліотеці класів .NET та підтримці багатопотоковості реалізуються фонові задачі моніторингу пропускної здатності каналу зв'язку. Це дозволяє динамічно застосовувати алгоритми адаптивного управління якістю потокового відео – оперативно змінювати роздільну здатність, бітрейт чи частоту кадрів залежно від поточного стану мережевого середовища, забезпечуючи високу плавність та неперервність сеансу відеозв'язку.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускну кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи клієнтського модуля ВКЗ з використанням технології WebRTC та алгоритмів адаптивного управління якістю потокового відео.

В процесі розробки випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти необхідно виконати наступний обсяг роботи:

а) провести аналіз існуючих систем-аналогів для виявлення їх позитивних і негативних якостей. Результати аналізу врахувати в подальших розробках;

б) вибрати та обґрунтувати методику побудови системи контролю роботи технологічного обладнання на виробництві в автоматизованому режимі. Розробити функціональну та структурну схеми системи;

в) розробити програмне забезпечення системи, що дозволить реалізувати поставлену технічним завданням задачу. Побудувати блок-схеми алгоритмів програми та підпрограми;

г) організувати інтерфейс користувача з метою формування та виводу на екран ЕОМ повідомлень про некоректні дії користувача та нестандартні ситуації в роботі технологічного обладнання;

д) розробити рекомендації по організаційних та методичних заходах, які забезпечать впровадження системи в промислову експлуатацію та її подальшу успішну експлуатацію;

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

ж) розробити заходи по охороні праці при впровадженні та експлуатації системи, а також розробити заходи з цивільного захисту;

з) сформувати висновки про виконаний обсяг робіт та одержані результати.

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

WebRTC – це не окремий протокол. Це набір існуючих протоколів зв'язку з низькою затримкою, які були вдосконалені та об'єднані з використанням JavaScript API. Коли ви ініціюєте сеанс WebRTC, відбувається кілька речей у швидкій послідовності.

Сигналізація

Перш ніж два браузери зможуть спілкуватися один з одним, їм потрібно обмінятися інформацією про себе. Це називається сигналізацією. WebRTC навмисно не визначає, як працює сигналізація, що насправді є розумним дизайном, він дає розробникам свободу. На практиці більшість систем використовують WebSocket. Два вузли обмінюються повідомленнями протоколу опису сеансу (SDP), які містять інформацію про підтримувані кодеки, медіа-можливості та конфігурацію мережі.

Проходження NAT через ICE, STUN та TURN

Ось тут і починається цікаве, і саме тут криється найбільша складність.

Ваш пристрій майже напевно знаходиться за рівнем NAT (перетворення мережевих адрес). Ваш маршрутизатор надає вам приватну IP-адресу, але Інтернет бачить іншу публічну IP-адресу. Це створює проблеми для однорангових з’єднань. ICE (встановлення інтерактивного з’єднання) – це протокол, який дозволяє одноранговим вузлам знаходити та встановлювати прямі мережеві шляхи.

STUN-сервери допомагають пристрою виявити його публічну IP-адресу. Якщо пряме підключення можливе, чудово. Якщо ні, сервери TURN виступають у ролі точок ретрансляції. TURN (Traversal Using Relays around NAT) забезпечує сервери ретрансляції, коли пряме підключення неможливе.

Транспортування медіа через RTP та SRTP

Після встановлення з’єднання починається медіапотік. RTP (протокол транспортування в реальному часі) передає медіапотоки. SRTP захищає їх. Таким чином аудіо та відео передаються між вузлами з мінімальною затримкою. Шифрування не є необов’язковим, воно вбудоване за замовчуванням.

Безпека за замовчуванням

WebRTC розроблено з урахуванням надійних заходів безпеки, включаючи обов'язкове наскрізне шифрування за допомогою SRTP/DTLS для всіх медіа та потоків даних. Цю функцію не вмикаєте. Це базовий стандарт. Кожен сеанс WebRTC шифрується без винятків.

Які основні API у WebRTC?

Три JavaScript API виконують важку роботу:

– getUserMedia() – Доступ до камери та мікрофона користувача. Це точка входу для будь-якого захоплення медіафайлів у WebRTC-застосунку.

– RTCPeerConnection – центральний API. Він представляє собою WebRTC-з’єднання між локальним комп’ютером та віддаленим вузлом і використовується для ефективної потокової передачі даних між двома вузлами. Він обробляє все: від узгодження кодеків до керування кандидатами ICE.

– RTCDataChannel – дозволяє передавати довільні дані між вузлами. Не лише аудіо та відео. Файли, повідомлення, стан гри – все, що потрібно вашій програмі. DataChannels розширює WebRTC за межі аудіо та відео, забезпечуючи обмін файлами, метадані, функції співпраці в реальному часі та сигналізацію програм.

Різниця між WebRTC та традиційним VoIP

Питання, яке виникає постійно, особливо від компаній, що оцінюють свій комунікаційний стек.

Традиційний VoIP працює на SIP (протокол ініціалізації сеансу) і був розроблений для світу стаціонарних телефонів, статичних офісів та фіксованої пропускної здатності. Він працює, але вимагає спеціалізованого обладнання, клієнтського програмного забезпечення та складної конфігурації.

WebRTC був розроблений для браузера, для роботи в режимі реального часу та для peer-to-peer з самого початку. Застарілий VoIP був створений для статичних кінцевих точок. Підприємство 2026 року живе у світі бездротових мереж 5G, віддалених перших команд, периферійних пристроїв Інтернету речей та хмарних мікросервісів.

Однак, ці два поняття не є взаємовиключними. Більшість корпоративних середовищ у 2026 році не є повністю новостворюваними. Вони залежать від SIP-транкових ліній, інфраструктури АТС та шлюзів PSTN. Добре спроектована платформа WebRTC повинна підтримувати гібридну роботу через шлюз WebRTC-SIP.

RTC League виконує саме цю інтеграцію через свою корпоративну SIP-транкінгову інфраструктуру, поєднуючи WebRTC-нативні додатки зі застарілими телефонними системами, щоб компаніям не довелося розривати та замінювати існуючі інвестиції.

Кодеки WebRTC

Кодеки визначають якість аудіо та відео. WebRTC підтримує кілька кодеків, кожен з яких має свої недоліки.

WebRTC підтримує адаптивні кодеки, включаючи VP8, VP9, H.264 та новий AV1. Гнучкість кодеків забезпечує сумісність між різними браузерами та пристроями, а також оптимізує якість відео залежно від використання пропускної здатності.

Щодо аудіо, кодек Opus є золотим стандартом у WebRTC. Він динамічно адаптується до мережевих умов, добре обробляє як голос, так і музику, і працює з надзвичайно низькими бітрейтами без погіршення якості. Це надзвичайно важливо для голосових агентів зі штучним інтелектом, де чистий аудіовхід безпосередньо впливає на точність розпізнавання намірів.

WebRTC у великих масштабах: коли peer-to-peer недостатньо

Модель однорангового зв'язку WebRTC чудово працює для з'єднань один на один. Але що відбувається, коли у відеодзвінку 50 учасників або 10 000 глядачів дивляться пряму трансляцію?

Відповіддю є архітектура медіасервера. Хоча peer-to-peer ефективний для комунікації малого масштабу, масштабні розгортання, такі як вебінари з тисячами учасників, вимагають медіасерверів для обробки транскодування, адаптивної потокової передачі бітрейту та масштабованості.

Блоки вибіркового пересилання (SFU) є найпоширенішим підходом. Замість того, щоб кожен учасник надсилав відео кожному іншому учаснику, кожен надсилає його до SFU, який вибірково направляє потоки. LiveKit є одним з провідних рішень SFU з відкритим кодом у цій галузі, а RTC League керує керованою інфраструктурою LiveKit для клієнтів, яким потрібен масштаб корпоративного рівня без накладних витрат на самостійне управління.

WebRTC повністю безкоштовний та має відкритий вихідний код. Ця безкоштовна реалізація вбудована в усі сучасні браузери, що робить її використання безкоштовним як для розробника, так і для користувача. Річ у тім, що якщо ви хочете створити з його допомогою додаток, вам доведеться за щось заплатити в певний момент. Для значущих додатків у WebRTC потрібна серверна інфраструктура.

Ця серверна інфраструктура включає сервери TURN (особливо критично важливі для корпоративних мереж за суворими брандмауерами), медіа-сервери для масштабованого розгортання, сигнальні сервери та інфраструктуру моніторингу. Сама технологія є безкоштовною. Витрати накопичуються саме на розгортанні виробничого рівня навколо неї.

WebRTC у 2026 році

Кілька подій змінюють те, як зараз використовується WebRTC.

Інтеграція штучного інтелекту поглиблюється. Транскрипція в реальному часі, придушення фонового шуму, аналіз настроїв під час дзвінків та голосові агенти на базі штучного інтелекту – все це вимагає транспортного рівня з низькою затримкою, який забезпечує WebRTC. Конвергенція інфраструктури WebRTC з моделями штучного інтелекту є найбільшою тенденцією в комунікації в реальному часі у 2026 році.

WebTransport стає все більш популярним. Побудований на HTTP/3 та QUIC, WebTransport пропонує деякі переваги для певних випадків використання потокової передачі. Він не замінює WebRTC, але додає додаткову опцію для певних сценаріїв.

Впровадження WebRTC на підприємствах є мейнстрімним. Існує чотири фактори, що сприяють впровадженню WebRTC замість VoIP на підприємствах у 2026 році: рішення, що встановлюються на основі браузера, не пропонують керування клієнтами, адаптивна підтримка кодеків, що забезпечує покращену якість голосового зв'язку, шифрування DTLS/SRTP, що забезпечує відповідність вимогам без проміжного програмного забезпечення, та відкритий код, що забезпечує корпоративну інфраструктуру за значно меншу ціну.

Заключне слово

WebRTC – це невидима інфраструктура, яка забезпечує комунікацію в режимі реального часу в Інтернеті. Вона знаходиться в Google Meet, коли ви приєднуєтеся до дзвінка з вкладки браузера. Вона знаходиться в голосових агентах зі штучним інтелектом, коли вони відповідають клієнту менш ніж за 200 мілісекунд. Вона знаходиться в платформах телемедицини, прямих трансляціях та корпоративних телефонних системах, які відмовилися від власного обладнання.

Для компаній, які будують на комунікації в режимі реального часу у 2026 році, питання не в тому, чи використовувати WebRTC, а в тому, як розгорнути його в правильному масштабі, з правильною архітектурою та правильною інфраструктурою навколо нього.

3.2 Розробка структурної схеми

Якщо ви створюєте, експлуатуєте або купуєте програмне забезпечення WebRTC – продукт для конференцій, кімнату телемедицини, онлайн-курс у реальному часі, відеоканал контакт-центру, торговий зал у реальному часі, інструмент для віддаленої співпраці або голосовий та відеосервіс зі штучним

інтелектом – користувацький досвід вашого продукту по суті визначається тим, наскільки добре працює оцінка пропускної здатності в найгірших мережах, у яких перебувають ваші глядачі. Коли оцінювач перевищує очікування, кадри зависають, звук переривається, а панелі контролю якості дзвінків світяться червоним. Коли очікування недооцінюється, зображення залишається розмитим, зображення на екрані розмивається, і кімната здається дешевою, навіть якщо пропускної здатності було достатньо. Інженери налаштовують цей цикл протягом року, а менеджери продуктів відправляють результат протягом десятиліття. Мета цієї статті полягає в тому, щоб дати немережевому інженеру ту саму ментальну модель, яку мають інженери WebRTC: що вимірює цикл, що він вирішує, де він консервативний, де він неправильний, і що ви можете змінити, щоб підштовхнути його.

Цикл керування в одному абзаці

Уявіть, що ви наливаєте воду у шланг і не бачите іншого кінця. Ви починаєте обережно. Через мить друг на дальньому кінці передзвонює і повідомляє вам, чи вода надходить вчасно, із запізненням чи взагалі ні. Ви прискорюєтеся, якщо відповіді залишаються чистими, і сповільнюєтеся, щойно вони прослизають. Відправник WebRTC робить саме це, від десяти до двадцяти разів на секунду, надсилаючи пакети замість води та повідомлення зворотного зв'язку RTCP замість друга. Відправник записує порядковий номер у кожен вихідний RTP-пакет, одержувач записує компактний RTCP-звіт зі списком цих порядкових номерів з позначками часу, що показують, коли кожен пакет прибув, а контролер перевантаження відправника порівнює цей час прибуття з часом відправлення пакетів. Якщо проміжок між відправкою та прибуттям збільшується, мережа заповнюється – зупиніться. Якщо проміжок залишається рівним і пакети не втрачаються, мережа в порядку – обережно підніміться і подивіться, чи змінилася відповідь. У цьому циклі вся гра. Все інше в цій статті – це деталі того, як ця гра правильно працює в умовах реальної втрати пакетів, джиттера мобільного радіо, подій переприв’язки NAT, раптового перехресного трафіку та десятитисячорядкової кодової бази, яка постачається в

кожній збірці Chromium.)

Чому медіадзвінку потрібен власний контролер перевантаження

Веб-сторінка, яка передає файл через HTTP, працює на TCP та успадковує контроль перевантаження TCP – зазвичай CUBIC, іноді BBR, іноді Reno. Медіа-виклик WebRTC навмисно цього не робить. Причина полягає в тому, що реакцією TCP на втрату пакетів є повторна передача втрачених байтів по порядку, що призводить до блокування початку рядка – новіші байти чекають після старіших, поки не надійдуть старі. Для завантаження файлу це правильний компроміс: пошкоджена електронна таблиця гірша за повільну. Для відео в реальному часі це неправильний компроміс: затримка в 200 мілісекунд, спричинена повторною передачею TCP, зробить звук чутно переривчастим, а відео помітно заїкатися, а затримка в одну секунду завершить розмову. Отже, медіафайли WebRTC працюють за протоколом UDP, упаковані як RTP із шифруванням SRTP (див. DTLS, SRTP, TLS, mTLS щодо медіафайлів), і механізм WebRTC має самостійно контролювати перевантаження, замість того, щоб дозволяти TCP робити це. Завдання контролера перевантаження полягає у визначенні швидкості, з якою програмі дозволено надсилати медіафайли, що програма потім забезпечує, знижуючи цільовий бітрейт свого кодера, відкидаючи шари одночасної трансляції або уповільнюючи темп передачі пакетів.

Компроміс, який робить контролер, також відрізняється від TCP. TCP оптимізує пропускну здатність: передача якомога більшої кількості байтів. Контролер WebRTC оптимізує низьку наскрізну затримку з прийнятною якістю, навіть якщо це означає залишення вільного простору. Ця різниця пояснює, чому стандартний стек WebRTC не буде так сильно тиснути на вузьке місце, як паралельне завантаження файлу; виклик має інші цілі, ніж завантаження, і контролер їх враховує.

Сигнал: що містить кожен пакет

Щоб цикл працював, відправнику потрібно, щоб одержувач міг визначити, який пакет і коли надійшов. Порядкові номери RTP існують (16 біт,

на кожен SSRC, визначені в RFC 3550), і їх достатньо для виявлення втрати, але вони не дозволяють одержувачу повідомляти про єдину часову шкалу для кількох медіапотоків в одному транспортному каналі. У типовому виклику відправник передає аудіопотік та відеопотік – іноді одночасне відео з трьома роздільними здатностями, іноді доріжку для спільного використання екрана, іноді шари Scalable Video Coding (SVC) – і контролер хоче обміркувати їх усі разом, оскільки вони мають одне й те саме вузьке місце в мережі. Це завдання розширення заголовка Transport-Wide Congestion Control.

Порядковий номер для всього транспорту

Відправник додає розширення заголовка RTP, яке викликається, transport-wide-cc-02до кожного вихідного пакета в з'єднанні. Розширення містить 16-бітове число, яке збільшується на одиницю для кожного пакета, надісланого цим транспортом, незалежно від того, до якого SSRC належить пакет. Формат задокументовано на власному сайті проекту WebRTC у розділі transport-wide-cc-02: два байти ідентифікатора розширення та довжини, два байти порядкового номера транспорту, загалом чотири байти. Оскільки число є для всього транспорту, а не для кожного потоку, одержувач може записати один ряд часу прибуття для аудіо, відео, шарів одночасної трансляції та будь-якого іншого SSRC, що спільно використовує транспорт, і відправник тоді може міркувати про один мережевий шлях замість одного шляху на SSRC.

Попереднє розширення під назвою abs-send-time (abs-send-time на сайті WebRTC) містить позначку часу передачі відправника з 24-бітною роздільною здатністю та є тим, що зчитує зворотний зв'язок REMB на стороні приймача (описаний нижче). Сучасні відправники майже завжди узгоджують обидва – abs-send-timeдля зворотної сумісності з приймачами REMB, transport-wide-cc-02для сучасного шляху зворотного зв'язку. Обидва узгоджуються за допомогою SDP; якщо одержувач не сигналізує про підтримку transport-wide-cc-02, відправник повертається до старішого шляху на основі REMB.

Зворотній зв'язок одержувача

Отримувач надсилає назад RTCP-пакет, у якому для групи порядкових

номерів транспортування зазначено, коли кожен пакет прибув (відносно базового часу) і чи прибув він взагалі. Формат – це повідомлення Transport Wide CC Feedback (Зворотний зв'язок щодо CC на рівні транспортування) – винахід Google, який ніколи не був стандартизований в IETF як RFC, але задокументований у draft-holmer-rmcat-transport-wide-cc-extensions-01 та постачається в кожній реалізації, похідній від libwebrtc. Частота зворотного зв'язку налаштовується, але зазвичай становить приблизно кожні 50 мс для відео, об'єднуючи порядкові номери транспортування, що спостерігалися з моменту попереднього зворотного зв'язку. Результатом є високочастотний часовий ряд, на який відправник може реагувати.

Друге, старіше повідомлення зворотного зв'язку – Очікуваний максимальний бітрейт приймача, REMB (визначений у draft-alvestrand-rmcat-remb-03) – досі використовується деякими приймачами. У світі REMB приймач самостійно запускає оцінювач на основі затримки та записує назад одне число: «надсилати не більше X біт за секунду в цьому потоці». Відправник підкоряється. У світі Transport-Wide CC приймач лише надсилає часові позначки, а відправник запускає оцінювач. Останнє є сучасним варіантом за замовчуванням, оскільки відправник може співвіднести часові позначки з інформацією, яку він має самостійно – що він намагався надіслати, що створив кодер, чи перебував зондовий пакет у польоті – та отримати кращу оцінку.

Стандартизована IETF альтернатива: RFC 8888 CCFB

У 2020 році IETF опублікувала RFC 8888, повідомлення зворотного зв'язку протоколу керування RTP (RTCP) для контролю перевантаження. RFC 8888, концептуально, є тим самим, що й зворотний зв'язок CC для всього транспорту: компактне повідомлення RTCP, яке повертає інформацію про надходження кожного пакета відправнику, щоб відправник міг запустити контролер перевантаження. Відмінності полягають у бітах: RFC 8888 повідомляє для кожного SSRC, а не для всього транспорту, містить явне поле позначення ECN-CE (явне сповіщення про перевантаження), яке дозволяє відправнику реагувати на сигнали ECN (важливо для історії L4S; докладніше нижче), та має

опублікований RFC. Вихідний код WebRTC чітко документує це: розширення CC для всього транспорту не може використовуватися одночасно з форматом зворотного зв'язку CCFB RFC 8888. Станом на 2026 рік підтримка RFC 8888 постачається в деяких реалізаціях, але ще не в libwebrtc за замовчуванням; міграція відбувається поступово.

Математика: як позначки часу прибуття стають бітрейтом

На цьому етапі відправник має, приблизно кожні 50 мс, список номерів транспортної послідовності, які він відправив, та час прибуття кожного з них. Два оцінювачі працюють над цим списком паралельно. Їхні виходи об'єднуються; кінцева бітрейт, що надсилається кодеру, є меншою з двох.

Оцінювач 1 – на основі затримки

Оцінювач на основі затримки є серцем Google Congestion Control. Поточним канонічним джерелом є (нині закінчений) Internet-Draft draft-ietf-rmcat-gcc-02(Holmer, Lundin, Carlucci, De Cicco, Mascolo, липень 2016 р.); алгоритм у вихідному коді libwebrtc з того часу змінився, але структура залишається такою ж, як і в проекті.

Інтуїція полягає в наступному. Якщо мережа не перевантажена, проміжок між двома пакетами у одержувача дорівнює проміжку у відправника – пакет A виходить о t, пакет B виходить о t + 10 мс, обидва проходять через порожню чергу, обидва прибувають з різницею в 10 мс. Якщо мережа починає перевантажуватися, пакет B прибуває пізніше, ніж очікуваний проміжок у 10 мс, оскільки йому довелося чекати в черзі. Оцінювач вимірює дельту часу між надходженнями – проміжок часу надходження мінус проміжок часу надсилання, який іноді називають варіацією односторонньої затримки – і подає його у фільтр Калмана, який оцінює повільний сигнал: наскільки змінюється тенденція затримки очікування в черзі? Адаптивний поріг (контролер сам коригує поріг у міру змін у мережі) класифікує тенденцію як один із трьох станів: недостатнє використання (проміжок зменшується – прискорюється), нормальний (проміжок рівний – утримується), надмірне використання (проміжок зростає – зменшується). Стан керує контролером швидкості, який мультиплікативно

збільшує бітрейт при "недостатньому використанні", утримує його при "нормальному" та мультиплікативно знижує його приблизно до поточної вхідної швидкості, виміряної приймачем) при "надмірному використанні".

Розроблений приклад у уповільненій зйомці. Скажімо, відправник надсилає зі швидкістю 1200 кбіт/с. Він надсилає відеопакет розміром 1250 байтів у момент часу відправника 100 000 мкс та пакет розміром 1250 байтів у момент часу 110 000 мкс (інтервал 10 мс). Приймач записує їх у момент часу 100 300 мкс та 110 330 мкс (інтервал між надходженнями на своєму тактовому сигналі 10,03 мс; базове зміщення 300 мкс не має значення для контролера). Дельта між надходженнями становить 10,03 мс − 10 мс = 30 мкс – фактично нічого. Відфільтрований тренд залишається близьким до нуля, поріг показує «нормальний», контролер утримується. Тепер запускається перехресний потік трафіку, і черга вузького місця починає заповнюватися. Наступна пара має інтервал на стороні відправника 10 мс, але інтервал надходження 13 мс – дельта 3 мс. Після кількох таких вибірок поспіль, оцінка тренду фільтром Калмана перетинає поріг, стан змінюється на «надмірне використання», і контролер множить цільову швидкість на коефіцієнт відхилення – зазвичай 0,85 від поточної швидкості вхідного отримання, близько 1000 кбіт/с у цьому прикладі – і повідомляє про це кодеру. Кодер скидає цільовий QP наступного кадру або пропускає кадр, швидкість надсилання падає до 1000 кбіт/с, черга виснажується, наступні вибірки повертаються до майже нульової дельти, стан повертається до «нормального», і через кілька секунд «нормального» контролер знову починає нарощувати швидкість – мультиплікативно, зазвичай з коефіцієнтом у діапазоні 1,05–1,08 на RTT – шукаючи нову швидкість вузького місця.

Весь цикл виконується один раз на пакет зворотного зв'язку (~50 мс). Протягом хвилинного виклику контролер приймає 1200 окремих рішень щодо цільового бітрейту. Кодер отримує новий цільовий бітрейт від контролера через Call::OnBitrateAllocationChangedінтерфейс libwebrtc (або еквівалент в інших стеках); кодер або перекодує наступний кадр з новим цільовим значенням, або, в режимі simulcast, перемикає, який шар він передає (див. Simulcast та SVC в SFU,

щоб дізнатися, як виглядає перемикач шарів з боку SFU).

Оцінювач 2 – на основі збитків

Оцінка на основі затримки хороша, але не звертає уваги на один режим відмови: з'єднання з втратами, коли пакети втрачаються ще до того, як вони потраплять у чергу. Радіоканал зі слабким сигналом, з'єднання Wi-Fi з колізіями, супутниковий зв'язок за негоди – вони часто мають низьку базову затримку (оскільки черги невеликі), але випадковим чином втрачають 2–10% пакетів. Оцінка на основі затримки сама по собі із задоволенням збільшить швидкість, оскільки її вибірки затримки виглядають чистими.

Оцінювач на основі втрат виявляє цей випадок. Він зчитує той самий зворотний зв'язок RTCP (які пакети надійшли, а які ні), обчислює коефіцієнт втрат протягом ковзного вікна та застосовує порогові значення: нижче 2% втрат оцінювач на основі втрат задоволений і видає високий "безпечний" бітрейт; вище 10% втрат він зменшує бітрейт приблизно до половини поточної швидкості надсилання; між 2 та 10% він лінійно інтерполює. Остаточна оцінка пропускної здатності, яку використовує кодер, дорівнює min (на основі затримки, на основі втрат). Компонент на основі втрат є захисною сіткою для випадку втрат, але без перевантаження.

Сама BWE на основі втрат розвивалася протягом трьох поколінь всередині libwebrtc, як це описано в зведених каталогах webrtcHacks Густаво Гарсії: оригінальна версія «пороги та ступінчасті зміни» з 2011 року, версія «згладжені пороги та тенденції» v1 приблизно з 2019 року та версія «максимальна ймовірність з кандидатських коефіцієнтів» v2, представлена у 2023 році. Зокрема, модель v2 – це те, що постачатимуться у виробничих стеках у 2026 році; вона ймовірнісно формулює залежність коефіцієнта втрат як функції швидкості відправлення та вибирає швидкість відправлення, яка максимізує спільну ймовірність спостережуваних втрат та спостережуваних підтверджених коефіцієнтів. Зміна має найбільше значення для нестабільних стільникових з'єднань, де старіші версії раніше різко перевищували або недовищували задані показники.

Зондування: третій трюк

Контролер, який реагує лише на те, що кодер вже надсилає, має проблему: він не може розрізнити між «кодер надсилає 800 кбіт/с, і мережу це влаштовує» та «кодер надсилає 800 кбіт/с, і мережа може передавати 8000 кбіт/с». Сигнал швидкості обмежений зверху вихідним сигналом кодера. Щоб знайти стелю, контролер повинен час від часу надсилати більше трафіку, ніж генерує кодер, перевіряти, чи цей додатковий трафік також надходить вчасно, і збільшувати швидкість кодера, якщо це так.

Саме це робить зондування пропускної здатності. Контролер вводить короткий пакет доповнення – зазвичай пакет тривалістю 100 мс або менше з навмисно вищою швидкістю, ніж поточна ціль кодера – і спостерігає за зворотним зв'язком. Якщо пакети пакету надходять з такою ж майже нульовою варіацією односторонньої затримки, як і звичайний трафік, мережа має запас ходу; контролер підвищує свою ціль. Якщо пакети пакету накопичуються, мережа цього не робить; контролер залишає ціль у спокої. Корисне навантаження зонда подається з власних байтів доповнення SFU або, в деяких стеках, шляхом дублювання існуючого потоку пакетів RTP. Точна поведінка та історія задокументовані в статті webrtcHacks "Зондування пропускної здатності WebRTC" та у вихідному коді libwebrtc у розділі modules/pacing/bitrate_prober.cc.

Зондування – це те, що пришвидшує запуск. Без нього виклик мав би зростати від початкової бітрейтової швидкості (зазвичай 300 кбіт/с для відео) до максимальної, чекаючи, поки кодер виведе більше бітів, чого кодер не зробить, доки контролер не дасть йому вказівку. За допомогою зондування контролер може генерувати пакет даних, скажімо, зі швидкістю 2000 кбіт/с протягом кількох сотень мілісекунд після початку виклику, зворотний зв'язок підтверджує, що шлях чистий, і кодеру наказують перейти одразу на 2000 кбіт/с. Ефект, видимий кінцевим користувачем, полягає в тому, що зображення стає різкішим протягом першої секунди, а не через десять.

Альтернативи: SCReAM та NADA

Робоча група IETF з координації мультимедійних застосунків реального часу (RMCAT) опублікувала три алгоритми у період 2017–2019 років: Google Congestion Control як Internet-Draft, SCReAM як RFC 8298 та NADA як RFC 8698. Усі три спрямовані на вирішення однієї й тієї ж проблеми та використовують однакові сигнали зворотного зв'язку RTCP. Вони відрізняються формою алгоритму та тим, що вони оптимізують.

SCReAM (Self-Clocked Rate Adaptation for Multimedia) – це алгоритм Ericsson, розроблений з урахуванням мобільних радіоканалів. Він базується на вікнах (відстежує, скільки байтів «перебувають у польоті» відносно вікна перевантаження, як це робить TCP), а не виключно на швидкості, і є гібридним алгоритмом втрат і затримок. Завдяки віконній природі він має самосинхронізацію: швидкість надсилання неявно вказана у вікні та часі обміну даними, тому алгоритму не потрібна явна цільова швидкість передачі даних. SCReAM постачається в деяких виробничих стеках RTC (зокрема, у власному Ericsson; існує публічна реалізація на EricssonResearch/scream), а у 2026 році триває оновлення версії 2 за адресою draft-johansson-ccwg-rfc8298bis-screamv2-03.

NADA (Network-Assisted Dynamic Adaptation – Динамічна адаптація за допомогою мережі) – це алгоритм Cisco. Це контролер на основі швидкості, побудований для використання позначок явного сповіщення про перевантаження (ECN), якщо вони доступні, та повертається до затримки та втрат, коли їх немає. ECN дозволяє мережевому пристрою позначати біт IP-заголовка пакета замість того, щоб відкидати пакет, коли його черга починає заповнюватися; одержувач відсилає позначку назад, а відправник уповільнює роботу, не втрачаючи даних. NADA – це алгоритм, який найкраще підходить для історії L4S (низька затримка, низькі втрати, масштабована пропускна здатність) – сімейства стандартів, опублікованих як RFC 9330–9332 – оскільки L4S спирається на ECN. Там, де ECN недоступний, NADA повертається до затримки та втрат, як і два інших.

Порівняння, яке хвилює більшість інженерів: який з них постачається в браузерах? У 2026 році відповідь не змінилася з тих, що були десять років тому: Chrome, Edge, Safari та Firefox постачають Google Congestion Control, оскільки libwebrtc постачає його. Стандартизовані алгоритми (SCReAM, NADA) постачаються невеликою кількістю спеціалізованих стеків та в дослідницьких статтях. Це важливо, оскільки вся екосистема взаємодії між браузерами, між браузерами та SFU, а також між SFU та браузерами успадковує поведінку GCC. Якщо ваш продукт у будь-який момент працює через браузер, особливості GCC – це ваші особливості.

Якщо ви використовуєте продукт, який працює в браузерах, GCC – це ваш алгоритм, хочете ви цього чи ні. SCReAM та NADA стають цікавими лише тоді, коли вашими кінцевими точками є не браузери – вбудовані пристрої, користувацькі стеки RTC, обладнання для широкомовного внеску – і навіть тоді вам доводиться зважувати експлуатаційні витрати на запуск алгоритму, який не входить до решти трафіку WebRTC в Інтернеті.

Що насправді робить кодер з числом

Контролер перевантаження, який генерує цільові бітрейти, – це лише половина справи. Кодер, менеджер шарів одночасної трансляції або SVC та пейсер повинні перетворити це число у фактичну поведінку на дроті.

Кодер отримує інформацію про нову цільову швидкість передачі даних, і він реагує на наступний кадр. Для однопотокового відео WebRTC – один видавець, одна роздільна здатність, один бітрейт – кодер змінює швидкість передачі даних: кодер libvpx VP8 або libvpx-vp9 з rc_target_bitrateоновленою версією створюватиме наступний кадр з новою швидкістю передачі даних, зазвичай коригуючи параметр квантування (QP) для досягнення цільового значення. Якість плавно падає з бітрейтом до мінімуму, де кодер змінює поведінку та починає пропускати кадри замість подальшого збільшення QP – зазвичай близько 150–250 кбіт/с для відео 720p.

Для відео одночасної трансляції контролер не повідомляє кодеру жодного числа; він повідомляє менеджеру шарів одночасної трансляції загальний

бюджет, а менеджер шарів вибирає, яка комбінація шарів одночасної трансляції підходить. Типовий клієнт libwebrtc публікує три шари: 720p зі швидкістю 1500 кбіт/с, 360p зі швидкістю 500 кбіт/с та 180p зі швидкістю 150 кбіт/с. Якщо загальна цільова швидкість становить 1500 кбіт/с, менеджер публікує всі три. Якщо цільова швидкість падає до 700 кбіт/с, менеджер видаляє шар 720p і публікує лише 360p та 180p – разом 650 кбіт/с, що вписується в бюджет з запасом. Якщо цільова швидкість падає ще більше до 200 кбіт/с, публікується лише шар 180p. Падіння різке на кожному кроці, що є однією з причин, чому якість одночасної трансляції може виглядати «ступінчастою», а не плавною, коли мережа погіршується.

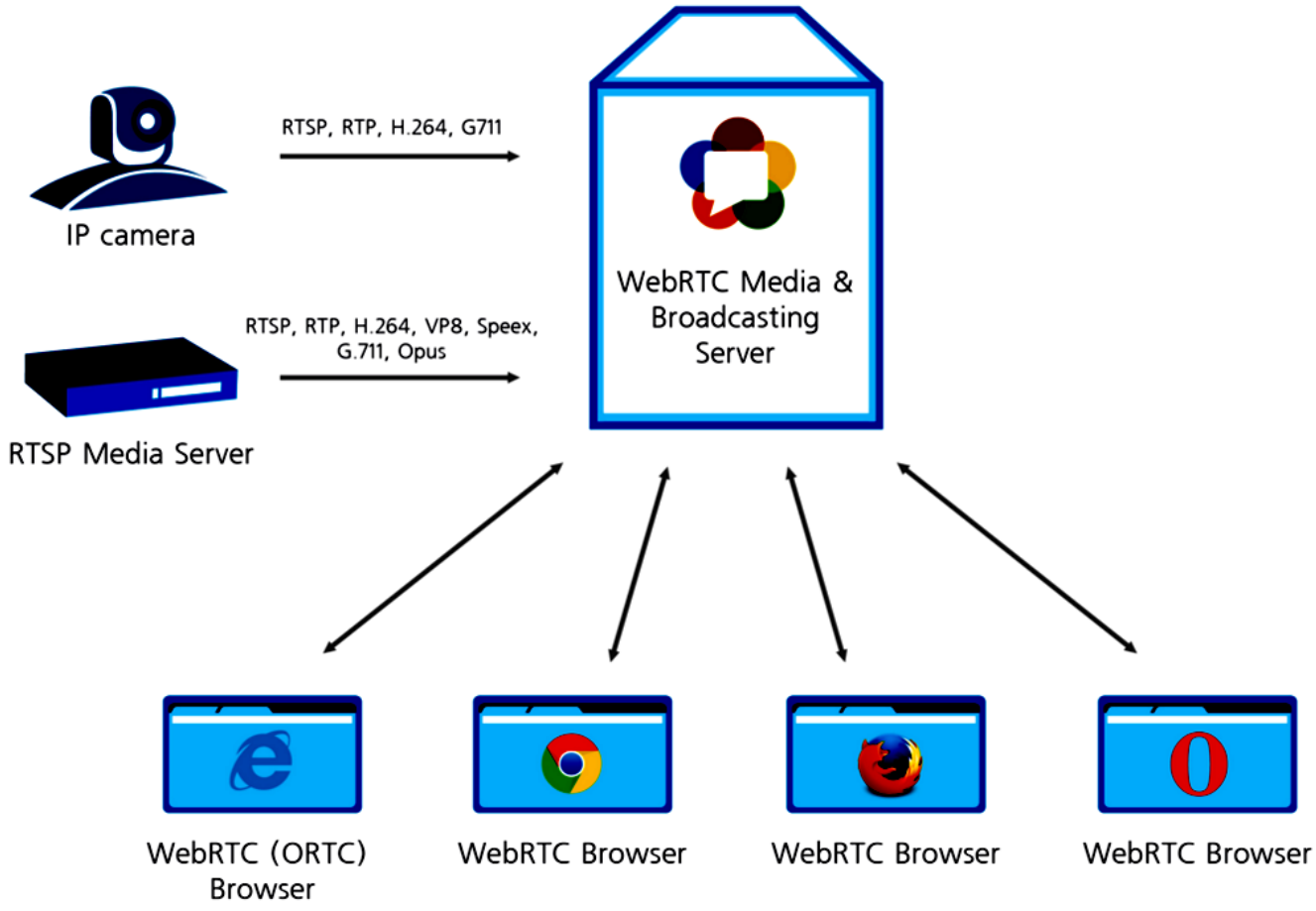


Рисунок 3.1 – Структурна схема системи

У розгортанні блоку вибіркової переадресації (SFU), SFU сам запускає оцінювач пропускної здатності для кожного абонента на низхідному каналі до кожного абонента та вибирає, який рівень одночасної трансляції пересилати цьому абоненту. Проект mediasoup безпосередньо документує це : оцінений вихідний бітрейт (від REMB або зворотного зв'язку Transport-Wide CC) розподіляється між споживачами, надаючи пріоритет доріжкам з вищим пріоритетом. Документація архітектури LiveKit описує ту саму схему: клієнтські SDK за замовчуванням публікують три рівні одночасної трансляції, і SFU вибирає рівень для кожного абонента на основі сигналів REMB та TWCC з кожного низхідного каналу (глибше занурення в архітектуру LiveKit). Один і той самий цикл виконується двічі – один раз між видавцем та SFU, один раз між SFU та кожним абонентом – тому неправильно налаштований шлях оцінки пропускної здатності на низхідному каналі SFU може зіпсувати враження, навіть якщо висхідний канал працює нормально.

Зрештою, пейсер – це те, що перетворює цільовий бітрейт у фактичні пакети в мережі, розподілені з часом. Без пейсера, кодер зі швидкістю 30 кадрів/с, що створює відео зі швидкістю 1 Мбіт/с, надсилав би всі байти кадру одним пакетом за один кадр, потім простоював би 33 мс, а потім знову надсилав би пакети. Така схема пакетної передачі погана для контролера: вона штучно збільшує затримку в черзі та заплутує оцінювач на основі затримки. Пейсер libwebrtc (в modules/pacing/) згладжує вихідну швидкість, утримуючи пакети в черзі та випускаючи їх з цільовою швидкістю, тому мережа бачить приблизно постійний потік замість пакетів. Пейсер також містить логіку зондування – коли контролер хоче провести зондування, пейсер фактично вводить додаткові пакети.

3.3 Розробка функціональної схеми

Функціональну схему розробленого клієнтського модуля системи відеоконференц-зв'язку (ВКЗ) наведено на рисунку 3.2.

Зі схеми вбачається, що архітектура клієнтської частини системи охоплює такі ключові функціональні блоки:

1. Параметри захоплення відеопотоку. Підтримувані роздільні здатності захоплення медіаданих з урахуванням можливостей апаратного кодування та пропускної здатності мережі:

– для локальних камер (веб-камер) – стандартні формати від низької роздільної здатності до високої (наприклад, 384x288, 640x480, 1280x720 та вище);

– для мережевих джерел та зовнішніх стрімів – широкий спектр дозволів (від 176x144 до 1600x1200) для забезпечення гнучкої адаптації під різні сценарії передачі даних через WebRTC.

2. Модуль аналізу активності та контролю стану медіапотоку. Цей компонент відповідає за відстеження стану каналу зв'язку та параметрів відеоряду в реальному часі. Він базується на аналізі метрик продуктивності та стабільності потоку, що дозволяє вчасно передавати сигнали до алгоритмів адаптивного управління якістю (ABR). Додатково реалізовано:

– фільтрацію шуму вхідного відеосигналу;

– виділення пріоритетних областей інтересу (ROI) для фокусування адаптивного кодування;

– групування подій активності у часові сеанси зв'язку.

3. Режими запису, локального буферування та архівації медіаданих. Система підтримує гнучкі сценарії збереження медіаконтенту (або сеансів ВКЗ):

– без збереження на диск (режим “живої” трансляції);

– запис відеопотоку за тригерами активності;

– розширені режими буферування (передзапис і післязапис фрагментів сеансу). Збереження медіафайлів може здійснюватися у форматах покадрового кодування, окремих кадрів (JPEG) або контейнерів (AVI / MP4) із використанням ефективних кодеків. Управління архівом реалізовано за кільцевим принципом (циклічне видалення застарілих даних при заповненні

дискового простору). Передбачено можливість накладення метаданих, міток часу та службового тексту на кадри.



Рисунок 3.2 – Функціональна схема системи

4. Режими медіаспостереження та трансляції. Перегляд відеопотоків у реальному часі забезпечується:

– за допомогою локального клієнтського застосунку з підтримкою мультиекранного виведення (макетів різної конфігурації);

– через сучасні веб-браузери на базі технології WebRTC без необхідності встановлення додаткових плагінів.

5. Авторизація, керування та налаштування. Доступ до розширених налаштувань клієнтського модуля та адміністрування здійснюється на основі автентифікації користувачів за паролем/токеном.

6. Доступ до архіву сеансів ВКЗ. Перегляд записаних матеріалів та логів подій реалізовано для авторизованих користувачів із відповідними правами доступу.

7. Система реакцій на мережеві та системні події. Реалізовано механізми оповіщення:

– звуковий супровід початку та завершення сеансу зв'язку;

– надсилання сповіщень через e-mail (або інші протоколи) із вкладеними даними;

– виконання попередньо налаштованих користувацьких скриптів і фонових завдань.

8. Багатокористувацький доступ і ролі. Архітектура передбачає розмежування прав для 4 основних груп користувачів: інсталятори, адміністратори, оператори архіву та спостерігачі. Програмні обмеження на кількість одночасних користувачів чи віддалених робочих місць відсутні.

9. Довідкова система та документація.

Після детального розгляду всіх структурних блоків функціональної схеми перейдемо до аналізу діаграми взаємодії процесів, які відбуваються в розробленій системі клієнтського модуля ВКЗ.

3.4 Розробка діаграми процесів

Діаграму процесів розробленої системи наведено на рисунку 3.3. Її детальний розгляд дозволяє простежити логіку взаємодії компонентів клієнтського модуля відеоконференц-зв'язку (ВКЗ). У межах структурного

проєктування використовується модель візуалізації потоків даних (Data Flow Diagram, DFD), яка описує рух інформаційних масивів у системі.

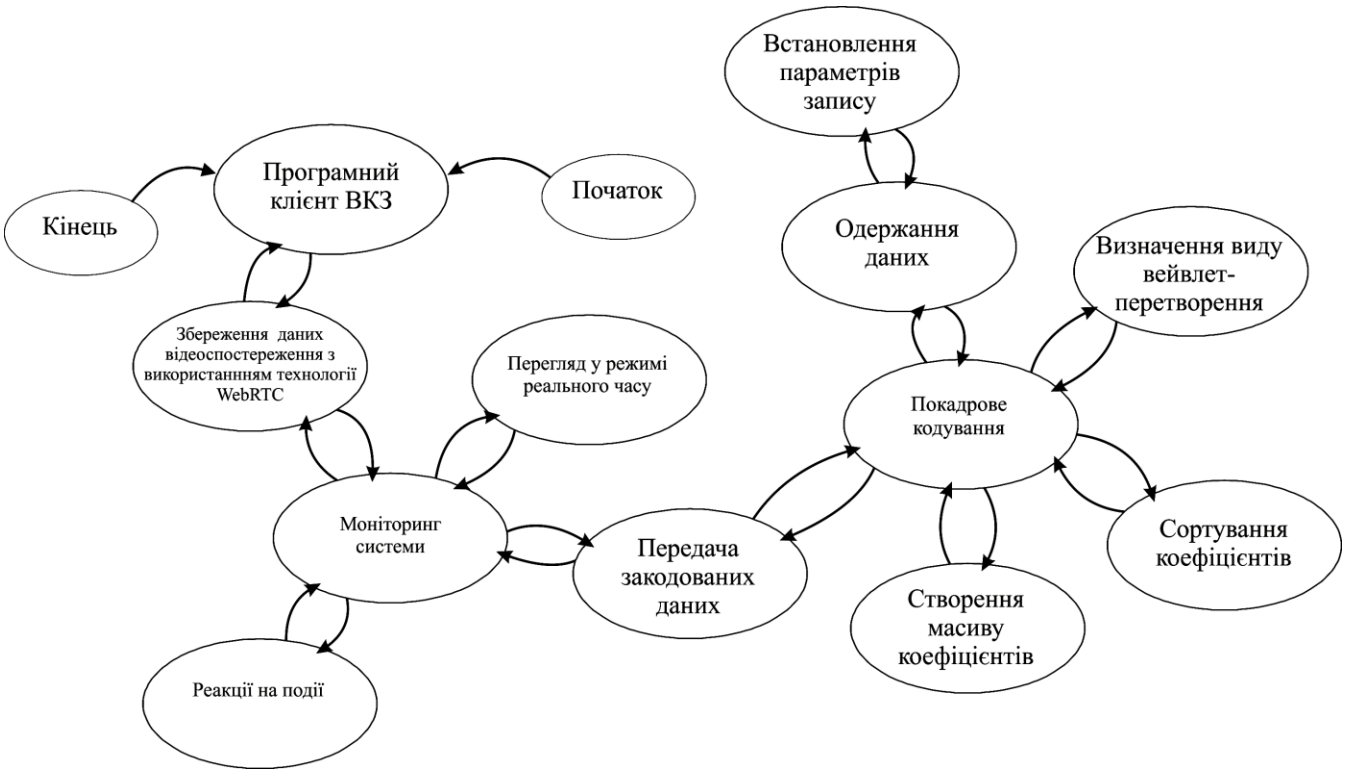


Рисунок 3.3 – Діаграма взаємодії процесів

Традиційно процес проєктування розпочинається з побудови контекстної діаграми потоків даних, яка визначає межі системи та її взаємодію із зовнішнім середовищем (користувачами, сигнальними серверами, мережевими інфраструктурами). У подальшому ця модель деталізується шляхом поетапної декомпозиції процесів і потоків даних, що дозволяє повноцінно відобразити архітектуру розроблюваного клієнтського модуля ВКЗ, включно з механізмами захоплення медіаданих, роботою сигналінгу WebRTC та адаптивним керуванням бітрейтом.

Таким чином, розглянувши опис системи, структурну, функціональну схеми системи, та діаграму взаємодії процесів перейдемо до опису блок-схем основної програми, та підпрограм, які використовуються, для реалізації системи.

4 РЕАЛІЗАЦІЯ ПРОЕКТУ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ПРАВИЛЬНІСТЬ ПРОЕКТНИХ РІШЕНЬ

4.1 Блок-схеми та опис алгоритмів функціонування системи

Алгоритмічне забезпечення клієнтського модуля відеоконференцзв’язку (ВКЗ) призначене для організації двостороннього передавання аудіо- та відеоданих у реальному часі, керування мультимедійними пристроями та підтримання прийнятної якості зв'язку за змінних характеристик мережі. Основою розроблюваної системи є технологія Web Real-Time Communication (WebRTC), яка забезпечує обмін мультимедійними потоками між учасниками конференції. Особливе значення має адаптивне управління параметрами потокового відео. Воно передбачає автоматичне коригування роздільної здатності, частоти кадрів та інтенсивності передавання даних відповідно до стану мережевого з'єднання і можливостей клієнтського пристрою.

Загальну послідовність роботи клієнтського модуля представлено блок-схемою на рисунку 4.1. Алгоритм функціонування охоплює ініціалізацію застосунку, перевірку доступності мультимедійних пристроїв, встановлення WebRTC-з'єднання, оброблення потоків та контроль їхньої якості. Після успішного підключення активується періодичний збір статистичних показників. На їхній основі визначається необхідність зміни параметрів відеопередавання. Робота алгоритму завершується після виходу користувача з конференції або припинення сеансу з технічних причин.

Архітектура клієнтського модуля передбачає логічне розділення функцій керування конференцією, оброблення медіапотоків та аналізу якості з'єднання. Керувальна частина відповідає за взаємодію з користувачем і сигнальним сервером. Мультимедійний компонент забезпечує захоплення, передавання та відтворення аудіо- і відеоданих.

Аналітичний компонент отримує статистику WebRTC та оцінює стан передавання. Такий розподіл дозволяє змінювати алгоритми адаптації без необхідності перебудови всього механізму організації відеозв'язку.

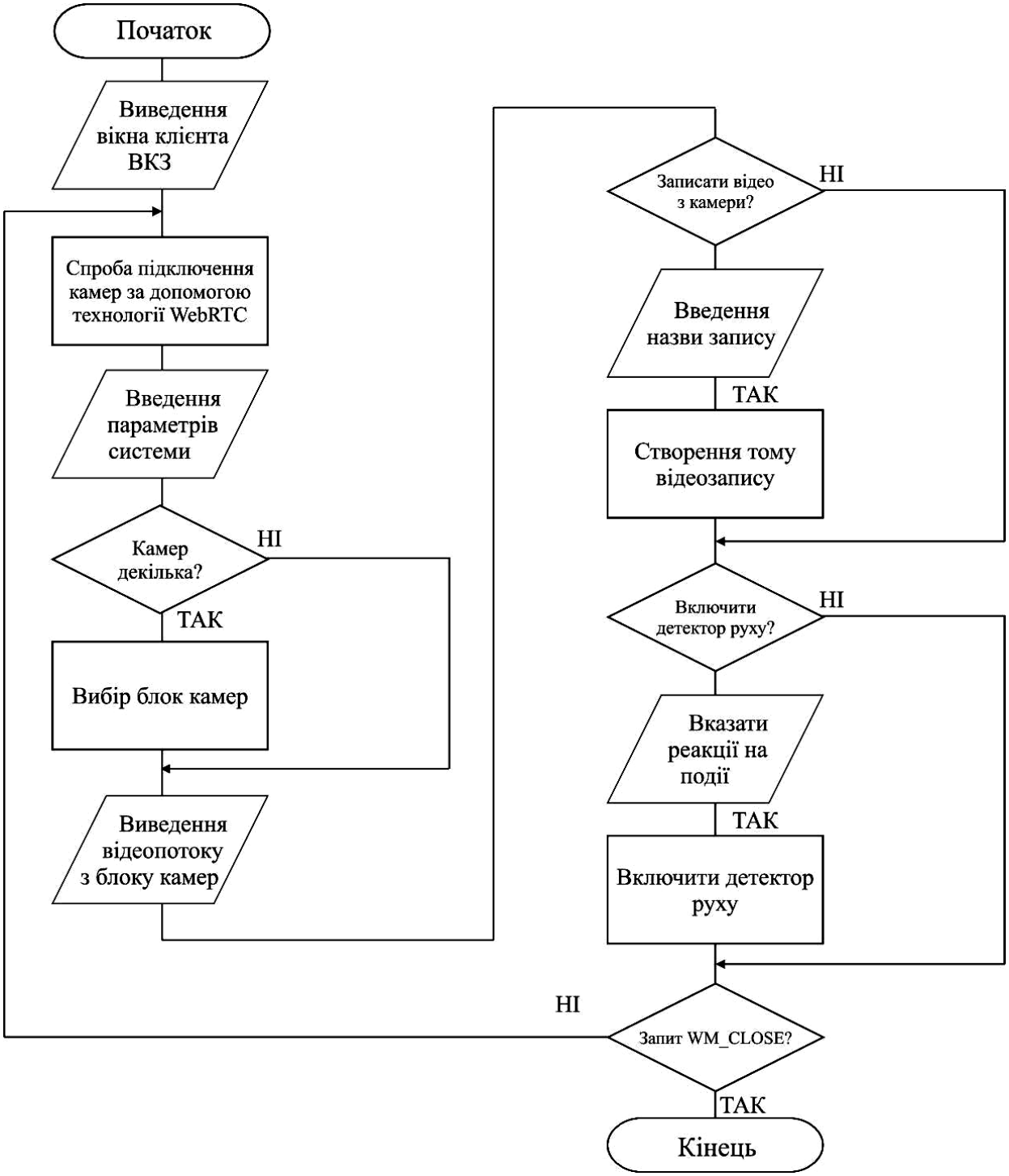


Рисунок 4.1 – Блок-схема основної програми

Встановлення медіаз'єднання здійснюється через програмний інтерфейс RTCPeerConnection. Для узгодження параметрів передавання використовуються повідомлення Session Description Protocol (SDP), а для визначення можливих мережевих маршрутів застосовується Interactive Connectivity Establishment (ICE). Обмін службовими повідомленнями потребує окремого сигнального механізму. Як можливий варіант його реалізації розглядається використання WebSocket. При цьому WebRTC не визначає обов'язкового протоколу сигналізації, тому конкретний спосіб обміну залежить від архітектури системи ВКЗ.

Окремим функціональним блоком є підпрограма адаптивного керування якістю вихідного відеопотоку. Її алгоритм відображено на рисунку 4.2. Підпрограма використовує відомості про пропускну здатність з'єднання, втрати пакетів, затримки та фактичну частоту кадрів. За результатами аналізу визначається поточний режим передавання і можливість його зміни. Якщо спостерігається погіршення умов, параметри відеопотоку можуть знижуватися. Повернення до вищої якості здійснюється після підтвердження стабілізації показників.

Алгоритм ініціалізації клієнтського модуля

Функціонування клієнтського застосунку починається з перевірки доступності програмних засобів, необхідних для організації відеозв'язку. На початковому етапі визначається підтримка браузером інтерфейсів WebRTC та можливість отримання аудіо- і відеоданих від локальних пристроїв. Додатково перевіряється наявність параметрів підключення до сервісу конференцій.

Вхідними даними алгоритму є конфігурація клієнтського модуля, параметри користувацького сеансу та відомості про доступні функції браузера. Конфігурація може включати адресу сигнального сервера, параметри ICE-серверів, початковий режим якості відео та правила оброблення помилок. Ці відомості повинні надходити з передбаченого джерела конфігурації, а не формуватися безпосередньо під час кожного встановлення з'єднання.

Після перевірки програмного середовища виконується ініціалізація внутрішнього стану застосунку. Створюються логічні об'єкти, необхідні для

зберігання інформації про учасників конференції, локальні мультимедійні потоки та активні з'єднання. Окремо готуються структури для накопичення статистики якості передавання.

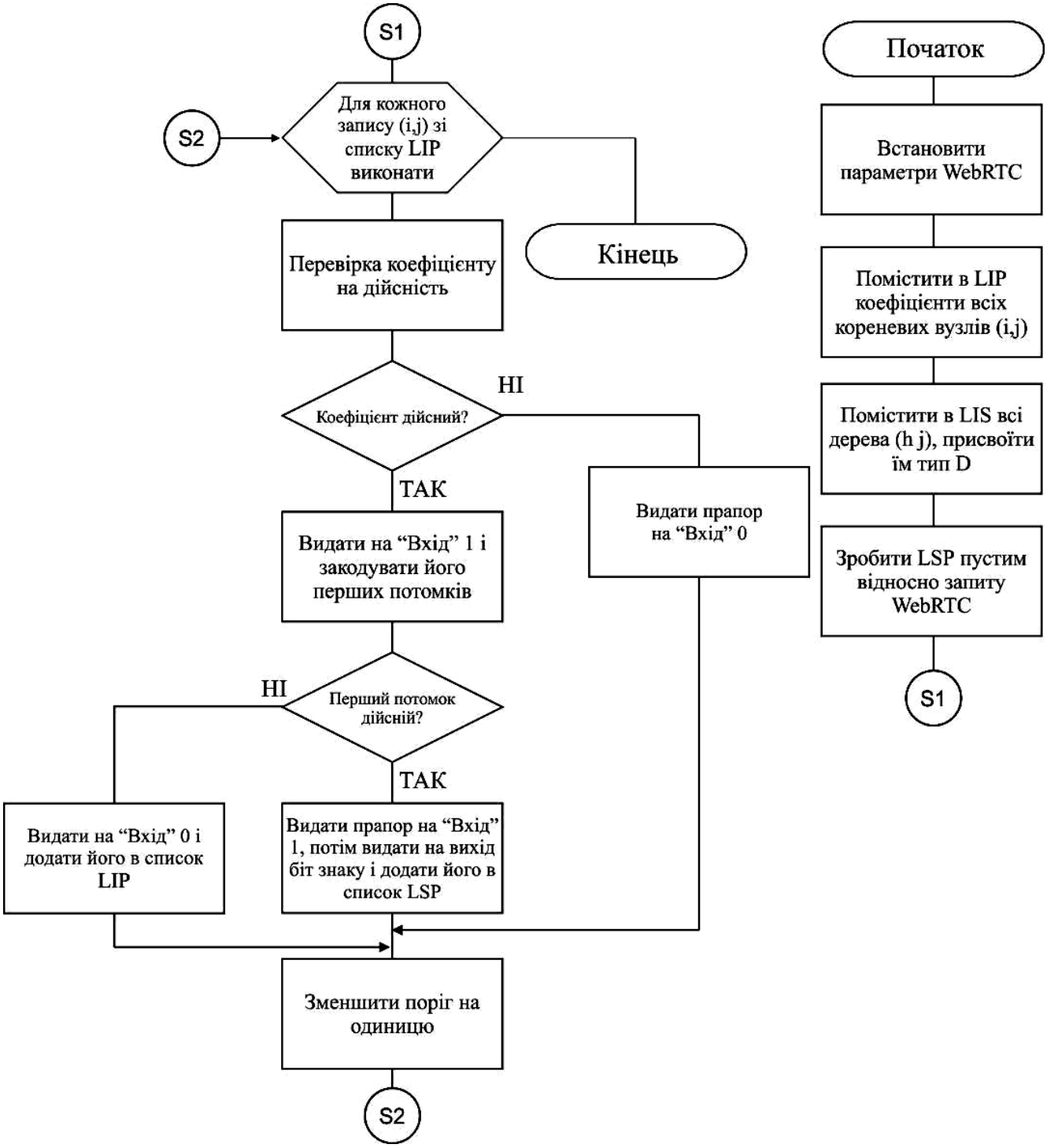


Рисунок 4.2 – Блок-схема роботи підпрограми

Доцільно використовувати модель керування станами клієнтського модуля. Початковий стан відповідає відсутності активної конференції. Після ініціювання підключення застосунок переходить до підготовки мультимедійних пристроїв. Наступні стани характеризують сигнальний обмін, встановлення медіаз'єднання та активне передавання потоків. У разі порушення зв'язку передбачається перехід до стану відновлення.

Переходи між станами повинні виконуватися після підтвердження завершення відповідної асинхронної операції. Наприклад, спроба передавання відео не повинна залежати від припущення, що користувач уже надав доступ до камери. Аналогічно створення повторного з'єднання необхідно узгоджувати зі станом попереднього підключення.

Якщо браузер не підтримує необхідні функції або відсутні обов'язкові параметри конфігурації, ініціалізація завершується з повідомленням про причину. При цьому застосунок повинен залишатися в узгодженому стані та не створювати незавершених фонових операцій.

Результатом алгоритму є підготовлений клієнтський модуль, готовий до захоплення мультимедійних даних та встановлення зв'язку з іншими учасниками.

Алгоритм захоплення та підготовки мультимедійних потоків

Перед передаванням аудіо- та відеоданих необхідно отримати доступ до відповідних локальних пристроїв. У браузерному середовищі для цього використовується інтерфейс MediaDevices, який надає можливість запитувати мультимедійні потоки через метод getUserMedia().

Вхідними параметрами алгоритму є бажані характеристики відео, необхідність використання мікрофона та поточні дозволи користувача. Для відеокамери можуть задаватися початкова роздільна здатність, частота кадрів та ідентифікатор пристрою. Для аудіопотоку можуть визначатися підтримувані налаштування оброблення звуку.

Після формування запиту браузер перевіряє можливість використання відповідного пристрою. Якщо доступ потребує дозволу користувача,

відображається стандартний механізм його надання. Подальша робота залежить від результату цього запиту.

У разі успішного отримання медіапотоку застосунок перевіряє наявність аудіо- та відеотреків. Для локального попереднього перегляду відеопотік може передаватися до відповідного елемента інтерфейсу. Такий перегляд доцільно організовувати без відтворення локального звуку, щоб уникнути небажаного акустичного зворотного зв'язку.

Якщо користувач заборонив доступ до камери, система повинна відобразити відповідне повідомлення. Відсутність відеопристрою не обов'язково повинна унеможливлювати участь у конференції. За підтримки відповідного режиму користувач може продовжити роботу з одним аудіоканалом.

Окремо обробляється ситуація, коли пристрій існує, але не підтримує запитаних характеристик. У такому випадку можливе повторне отримання потоку з менш жорсткими обмеженнями. Наприклад, замість обов'язкової високої роздільної здатності можуть використовуватися рекомендовані значення, які браузер узгоджує з можливостями камери.

Під час активного сеансу користувач може вимкнути камеру або мікрофон. Для тимчасового вимкнення локального медіатреку допускається зміна властивості enabled. Така операція відрізняється від повного завершення роботи пристрою через stop(). Після зупинки треку його повторне використання без нового захоплення неможливе.

Для зміни камери без створення нового з'єднання може застосовуватися RTCRtpSender.replaceTrack(). Перед заміною необхідно отримати новий трек і переконатися в його придатності. Якщо заміна не підтримується в межах поточного узгодження, може знадобитися повторне погодження параметрів з'єднання.

Результатом алгоритму є локальні мультимедійні треки, підготовлені для передавання через WebRTC.

Алгоритм сигнальної взаємодії та узгодження параметрів зв'язку

Для встановлення WebRTC-з'єднання учасники повинні обмінятися інформацією про підтримувані мультимедійні параметри та мережеві можливості. Цей процес організовується за допомогою сигнального каналу, реалізація якого визначається серверною архітектурою ВКЗ.

На початковому етапі клієнт встановлює з'єднання із сигнальним сервером. Після проходження передбаченої перевірки доступу він надсилає запит на приєднання до конференції. У відповідь може надходити інформація про поточний сеанс і доступних учасників.

Наступним етапом є створення об'єкта RTCPeerConnection. У його конфігурації задаються параметри ICE-серверів та інші підтримувані налаштування. Після цього до з'єднання додаються локальні мультимедійні треки або створюються відповідні трансівери.

Для початкового узгодження клієнт може сформувати SDP-пропозицію за допомогою createOffer(). Отриманий опис встановлюється як локальний через setLocalDescription() та передається іншій стороні сигнальним каналом.

Сторона, яка отримала пропозицію, обробляє її через setRemoteDescription(). Після перевірки сумісності формується SDP-відповідь за допомогою createAnswer(). Вона встановлюється локально та передається ініціатору. Після отримання відповіді ініціатор завершує відповідний етап узгодження.

SDP містить інформацію, необхідну для встановлення медіасеансу. Зокрема, в ньому узгоджуються медіанапрями, підтримувані кодеки та параметри транспорту. Під час оброблення повідомлень необхідно контролювати стан сигналізації, оскільки не кожна операція допустима на довільному етапі з'єднання.

Особливу увагу слід приділити одночасному ініціюванню переговорів двома сторонами. Така ситуація може спричинити конфлікт SDP-пропозицій. Для її оброблення може використовуватися підхід узгоджених переговорів, відомий як perfect negotiation. Він передбачає визначення правил оброблення конфліктних пропозицій та використання допустимих механізмів відкочування стану.

Повідомлення сигналізації повинні бути пов'язані з конкретним сеансом. Після завершення конференції застарілі відповіді не повинні повторно активувати закрите з'єднання. Для цього доцільно контролювати ідентифікатори сеансів і поточний стан клієнтського модуля.

Результатом алгоритму є узгоджені параметри WebRTC-з'єднання, необхідні для подальшого встановлення транспортного каналу.

Алгоритм встановлення транспортного з'єднання

Після початку узгодження мультимедійних параметрів необхідно визначити мережевий маршрут між учасниками. Для цього WebRTC використовує механізм ICE, який забезпечує пошук і перевірку можливих способів з'єднання.

На початковому етапі клієнт формує локальні ICE-кандидати. Вони можуть відповідати локальним мережевим інтерфейсам, адресам, отриманим через STUN, або ретрансляційним адресам TURN-сервера. Кожний кандидат характеризує можливий спосіб організації мережевого обміну.

Застосування STUN дозволяє визначати зовнішні параметри мережевого доступу в середовищах із трансляцією адрес. Однак наявність STUN не гарантує можливості встановлення прямого з'єднання. У випадках, коли безпосередній обмін неможливий, може використовуватися TURN-сервер для ретрансляції трафіку.

Для зменшення часу встановлення зв'язку доцільно використовувати поступове передавання кандидатів. Цей підхід відомий як Trickle ICE. Нові кандидати передаються іншій стороні в міру їх отримання без необхідності очікувати завершення всього процесу пошуку.

Після обміну кандидатами механізм ICE перевіряє можливість використання відповідних пар мережевих адрес. У результаті визначається придатний маршрут для передавання мультимедійних потоків. Вибір здійснюється відповідно до правил ICE та характеристик доступних кандидатів.

Клієнтський модуль контролює зміну стану ICE-з'єднання. Перехід до стану успішного встановлення дозволяє визначити готовність транспортного

рівня. Якщо перевірка всіх доступних маршрутів завершилася невдало, формується повідомлення про неможливість встановлення зв'язку.

Після визначення транспортного маршруту WebRTC виконує необхідні процедури захищеного встановлення медіасеансу. Передавання аудіо- та відеоданих здійснюється через механізми захищеного RTP. Відповідні криптографічні параметри узгоджуються під час встановлення з'єднання.

Результатом алгоритму є транспортне з'єднання, придатне для обміну мультимедійними потоками. Надалі його стан контролюється протягом усього сеансу конференції.

Алгоритм передавання та відтворення мультимедійних потоків

Після встановлення з'єднання локальні мультимедійні треки передаються через відповідні об'єкти RTCRtpSender. Браузер виконує кодування медіаданих, формування транспортних пакетів і передавання їх мережею. На приймальній стороні здійснюються отримання, декодування та відтворення потоків.

Для передавання відео можуть використовуватися кодеки VP8, VP9, H.264 або AV1 залежно від підтримки конкретними учасниками та результатів узгодження. Вибір кодека впливає на ступінь стиснення, потребу в обчислювальних ресурсах і можливості адаптивного керування. Наявність кодека в браузері не означає, що він обов'язково буде використаний для кожного з'єднання.

Для приймання віддалених потоків застосовується механізм оброблення події track. Після отримання нового треку клієнт визначає його тип і пов'язує з відповідним учасником конференції. Відеопотік передається до елемента відображення, а аудіопотік спрямовується на доступний пристрій відтворення.

Під час приймання потокового відео важливим є збереження рівномірності відтворення. Пакети можуть надходити з неоднаковими інтервалами через зміни мережевої затримки. Для компенсації такого явища WebRTC використовує внутрішні механізми буферизації та синхронізації відтворення.

Збільшення буфера може зменшувати кількість переривань відтворення, але водночас підвищувати затримку. Тому клієнтський модуль не повинен

безпідставно збільшувати буферизацію. У системах відеоконференцзв'язку оперативність взаємодії між учасниками має важливе значення.

Якщо віддалений потік припинив надходити, застосунок перевіряє стан відповідного треку та з'єднання. Тимчасове припинення отримання кадрів не завжди означає завершення конференції. Причиною може бути перевантаження мережі, вимкнення камери учасником або зміна конфігурації передавання.

Для забезпечення коректної роботи інтерфейсу необхідно узгоджувати стан мультимедійних треків зі станом учасників. Після відключення користувача пов'язані елементи відтворення повинні звільняти ресурси та припиняти використання неактуальних потоків.

Алгоритм отримання статистичних показників WebRTC

Для реалізації адаптивного керування якістю необхідно періодично оцінювати фактичні характеристики передавання. Технологія WebRTC надає механізм отримання статистики через метод getStats(). Він дозволяє одержувати інформацію про транспортні з'єднання, вихідні та вхідні RTP-потоки.

Вхідними даними алгоритму є активне WebRTC-з'єднання та встановлений інтервал збору статистики. Після отримання звіту клієнт визначає записи, які належать до контрольованого відеопотоку. Важливо правильно зіставляти показники з конкретним відправником або одержувачем, оскільки одна конференція може містити декілька мультимедійних потоків.

Для вихідного відеопотоку аналізуються кількість переданих байтів, кількість відправлених пакетів, число закодованих кадрів та доступні відомості про параметри кодування. Для вхідного потоку можуть використовуватися кількість отриманих пакетів, втрати, міжпакетні коливання затримки та показники декодування.

Окрему інформацію надають статистичні записи транспортного рівня. Вони можуть містити дані про вибрану пару ICE-кандидатів та час проходження мережевого обміну в обидва боки. Значення Round-Trip Time використовується як один із показників поточного стану мережі.

Для оцінювання фактичної інтенсивності передавання необхідно порівнювати послідовні значення накопичувальних лічильників. Наприклад, різниця між кількістю переданих байтів за два вимірювання дозволяє визначити обсяг даних, переданий протягом відповідного інтервалу. Для правильної інтерпретації враховується фактична тривалість між вимірюваннями.

Показники втрат пакетів також потребують коректного вибору джерела. Дані про втрати вхідного потоку характеризують приймання на поточному клієнті. Для оцінювання доставки вихідного потоку можуть використовуватися доступні віддалені статистичні повідомлення. Якщо такі відомості відсутні, їх не слід замінювати довільними оцінками.

Деякі браузери надають додаткову інформацію про причини обмеження якості кодування. Зокрема, поле qualityLimitationReason може вказувати на обмеження через пропускну здатність або обчислювальні ресурси. Однак доступність цього показника залежить від реалізації браузера. Тому алгоритм не повинен вимагати його обов'язкової присутності.

Під час отримання статистики перевіряється коректність часових позначок та актуальність даних. Після перезапуску з'єднання накопичувальні лічильники можуть починатися з нових значень. У такому випадку результати нового сеансу не повинні механічно порівнюватися зі статистикою попереднього підключення.

Підготовлені показники зберігаються у короткочасній історії спостережень. Вона використовується для аналізу тенденцій і визначення необхідності зміни параметрів відео. Принципи отримання й інтерпретації статистики визначаються специфікацією WebRTC та відповідними інтерфейсами статистичних звітів.

Алгоритм оцінювання якості мережевого з'єднання

Після отримання статистичних показників необхідно визначити, чи відповідають поточні характеристики мережі вимогам передавання відео. Оцінювання повинно спиратися на сукупність параметрів, оскільки окремий показник не завжди дозволяє достовірно встановити причину погіршення якості.

Вхідними даними алгоритму є історія вимірювань пропускної здатності, втрат пакетів, затримки та фактичних характеристик відеопотоку. На початковому етапі перевіряється достатність накопиченої інформації. Якщо кількість вимірювань недостатня або дані містять значні пропуски, алгоритм зберігає поточний режим якості до отримання достовірніших показників.

Наступним етапом є згладжування короткочасних коливань. У реальних мережах значення затримки та швидкості передавання можуть змінюватися навіть за стабільного загального навантаження. Якщо реагувати на кожне окреме відхилення, система буде надто часто змінювати параметри відео. Це може негативно впливати на сприйняття конференції користувачем.

Для згладжування може застосовуватися ковзне середнє або експоненційне згладжування. Вибір способу визначається необхідною чутливістю. Коротке вікно дозволяє швидше виявляти зміни, але підвищує вплив випадкових коливань. Довше вікно забезпечує стабільнішу оцінку, проте затримує реакцію на перевантаження.

Після попереднього оброблення аналізуються основні ознаки погіршення з'єднання. Зростання втрат пакетів може свідчити про перевантаження мережі або порушення доставки. Збільшення затримки може вказувати на накопичення пакетів у мережевих чергах. Зменшення фактичної інтенсивності передавання за одночасного погіршення інших показників може бути ознакою недостатньої пропускної здатності.

Під час аналізу необхідно розрізняти мережеві обмеження та проблеми кодування. Наприклад, зменшення частоти кадрів може бути спричинене перевантаженням центрального процесора навіть за достатньої пропускної здатності мережі. Тому доцільно враховувати доступні показники роботи відеокодера та фактичну частоту формування кадрів.

Результатом алгоритму є оцінка поточного стану передавання. Для керування якістю можуть використовуватися стани нормальної роботи, початкового погіршення та підтвердженого перевантаження. Окремо визначається ситуація недостатньої кількості статистичних даних.

Алгоритм адаптивного управління якістю потокового відео

Адаптивне управління якістю є центральним алгоритмом клієнтського модуля. Його призначення полягає у виборі параметрів відеопотоку відповідно до умов передавання. Основним завданням є збереження безперервності відеоконференції за зміни пропускної здатності мережі та обчислювальних можливостей пристрою.

Вхідними даними алгоритму є поточна оцінка якості з'єднання, статистика відеопередавання, характеристики локального пристрою та встановлені обмеження. Результатом є рішення щодо збереження, підвищення або зниження параметрів вихідного відеопотоку.

Для організації керування доцільно передбачити декілька профілів якості. Початковий профіль може передбачати невисоку роздільну здатність і помірну частоту кадрів. Наступний профіль забезпечує вищу деталізацію. Максимальний профіль використовується за достатніх мережевих і обчислювальних ресурсів. Конкретні числові характеристики визначаються під час налаштування системи та перевірки можливостей браузера.

Після встановлення з'єднання обирається початковий профіль. Його доцільно визначати з урахуванням параметрів камери та доступних відомостей про середовище. Використання максимальних характеристик без перевірки умов може спричинити додаткове навантаження і затримки.

Під час роботи алгоритм періодично отримує оновлену оцінку якості. Якщо показники перебувають у допустимих межах, поточний профіль зберігається. Якщо протягом декількох послідовних вимірювань спостерігається стійке погіршення, формується рішення про зниження параметрів.

Зниження якості може виконуватися шляхом обмеження максимальної інтенсивності передавання, зменшення частоти кадрів або роздільної здатності. Вибір параметра залежить від характеру проблеми. Наприклад, у разі обмеженої пропускної здатності доцільно зменшити обсяг відеоданих. За нестачі обчислювальних ресурсів важливим може бути зниження складності кодування.

У браузерному середовищі зміна параметрів вихідного RTP-потоку може виконуватися через RTCRtpSender.getParameters() та RTCRtpSender.setParameters().

Перший метод повертає поточну конфігурацію відправника. Після зміни допустимих параметрів конфігурація передається другому методу. Зокрема, можуть використовуватися maxBitrate, maxFramerate та scaleResolutionDownBy, якщо відповідні можливості підтримуються браузером.

Параметр maxBitrate задає верхнє обмеження інтенсивності передавання, але не гарантує досягнення зазначеної швидкості. Фактичне значення залежить від стану мережі та внутрішніх алгоритмів WebRTC. Параметр maxFramerate використовується для обмеження частоти кадрів, а scaleResolutionDownBy дозволяє зменшувати роздільну здатність під час кодування.

Для зміни характеристик вихідного відеотреку також може застосовуватися MediaStreamTrack.applyConstraints(). При цьому необхідно розмежовувати зміну параметрів захоплення та зміну параметрів RTP-кодування. Перший механізм впливає на отримання відео від пристрою, а другий на обмеження його передавання.

Важливо враховувати, що WebRTC уже містить внутрішні механізми оцінювання доступної пропускної здатності та керування перевантаженням. Запропонований прикладний алгоритм не повинен конкурувати з ними через часті або суперечливі зміни параметрів. Його функцією є вибір допустимих меж і профілю якості з урахуванням більш тривалих тенденцій.

Після зниження параметрів алгоритм продовжує аналіз показників. Якщо втрати пакетів і затримка зменшуються, обраний режим може зберігатися. Якщо погіршення триває, допускається перехід до наступного, менш ресурсомісткого профілю.

Відновлення високої якості повинно виконуватися обережніше, ніж її зниження. Для підвищення профілю необхідно підтвердити стабільність умов протягом визначеного часу. Це зменшує ризик повторного перевантаження після короткочасного покращення мережі.

Після переходу до вищого профілю виконується повторне оцінювання результату. Якщо параметри з'єднання швидко погіршуються, система повертається до попереднього режиму. Для запобігання частим перемиканням доцільно використовувати різні умови підвищення та зниження якості.

Такий механізм відповідає принципу гістерезису. Він дозволяє уникати ситуації, коли невеликі коливання поблизу порогового значення постійно спричиняють зміну роздільної здатності або частоти кадрів.

Конкретні параметри адаптації повинні визначатися експериментально. Для цього необхідно дослідити поведінку WebRTC за різних обмежень пропускної здатності, рівнів втрат і характеристик клієнтського обладнання. Без відповідних вимірювань неможливо обґрунтовано встановити універсальні пороги для всіх мережевих середовищ.

Алгоритм розподілення ресурсів між мультимедійними потоками

Під час відеоконференції одночасно передаються аудіо- та відеодані. За обмеженої пропускної здатності необхідно враховувати різну чутливість цих потоків до погіршення мережевих характеристик. Для мовленнєвого зв'язку важливими є безперервність відтворення та невелика затримка. Відеопотік допускає ширший діапазон зміни роздільної здатності й частоти кадрів.

У клієнтському модулі доцільно передбачити політику пріоритетного збереження працездатності аудіоканалу. У разі зниження доступної пропускної здатності алгоритм спочатку оцінює можливість обмеження відеопередавання. Це дозволяє зменшувати загальне навантаження без негайного припинення голосового зв'язку.

Вхідними даними є статистика обох типів потоків та поточні характеристики транспортного з'єднання. Після виявлення перевантаження аналізується можливість зниження відеобітрейту. Якщо доступних ресурсів недостатньо навіть після переходу до мінімального профілю відео, можливим рішенням є тимчасове припинення його передавання відповідно до політики конференції.

У разі відновлення мережевих умов відеоканал може бути активований повторно. Перед цим перевіряється стан локального треку та відповідного RTP-відправника. Для мінімізації навантаження зміна режиму повинна виконуватися без зайвого повторного створення всього WebRTC-з'єднання.

Важливо враховувати, що безпосередній розподіл транспортної пропускної здатності виконується механізмами реалізації WebRTC. Прикладний алгоритм може визначати допустимі обмеження для потоків, але не повинен вважатися повною заміною внутрішнього контролю перевантаження.

Алгоритм адаптації до обчислювальних можливостей пристрою

Якість відеоконференції залежить не лише від мережі, а й від характеристик клієнтського обладнання. Кодування відео потребує процесорних ресурсів, а його відтворення створює навантаження на підсистему декодування та графічний інтерфейс.

За високого завантаження пристрою фактична частота кадрів може зменшуватися навіть за стабільного мережевого з'єднання. Тому алгоритм адаптації повинен розрізняти ознаки мережевого перевантаження та недостатньої продуктивності локального середовища.

Вхідними даними можуть бути доступні статистичні показники кодування, кількість фактично сформованих кадрів і значення причин обмеження якості. За наявності відповідних даних аналізується тривалість кодування кадрів та співвідношення між заданою і фактичною частотою передавання.

Якщо зменшення частоти кадрів супроводжується ознаками обмеження через процесорні ресурси, доцільно знизити складність формування відеопотоку. Для цього можуть використовуватися нижча роздільна здатність або менша частота кадрів.

За стабілізації показників допускається поступове повернення до попереднього профілю. При цьому необхідно враховувати, що браузер не завжди надає повну інформацію про фактичне завантаження процесора або графічного прискорювача. Тому рішення повинні ґрунтуватися на доступних показниках і не припускати наявності відсутніх даних.

Розглянемо частину вихідного коду.

```
namespace AD_q
{
using System;
using System.Drawing;
using System.IO;
using System.Threading;
using System.Net;
using videosource;
public class SPIHTSource : IVideoSource
{
private stringsource;
  private string login = null;
  private string password = null;
  private object userData = null;
  private int  framesReceived;
  private int  bytesReceived;
  private bool useSeparateConnectionGroup = false;
  private bool preventCaching = false;
  private int  frameInterval = 0;
// інтервал подання фреймів у мілісекундах
  private const int bufSize = 512 * 1024;
// розмір буферу
  private const int readSize = 1024;
// розмір блоку для читання
  private Thread thread = null;
  private ManualResetEvent stopEvent = null;
  // нова подія у фреймі
  public event CameraEventHandler NewFrame;
  // SeparateConnectioGroup властивості
  // індикатор відкриття WebRequest у групі
  public bool SeparateConnectionGroup
  {
   get { return useSeparateConnectionGroup; }
   set { useSeparateConnectionGroup = value; }
  }
  // PreventCaching властивості
  // Якщо властивості є правильними, то керуємо
  // параметриу URL. Цей клієнт повинен бути встановлений на проксі-сервері.
  public bool PreventCaching
  {
   get { return preventCaching; }
```

```
    set { preventCaching = value; }
   }
   // FrameInterval властивості – інтервал між фреймами
   //Якщо властивості встановлені в 100, тоді джерело формує 10 фреймів
   // в секунду
   public int FrameInterval
   {
    get { return frameInterval; }
    set { frameInterval = value; }
   }
   // VideoSource властивості
   public virtual string VideoSource
   {
    get { return source; }
    set { source = value; }
   }
   // Властивості підключення
   public string Login
   {
    get { return login; }
    set { login = value; }
   }
   // Точка входу події
   public void WorkerThread()
   {
    byte[]   buffer = new byte[bufSize];
// буфер читання потоку
    HttpWebRequest req = null;
    WebResponse  resp = null;
    Stream   stream = null;
    Random   rnd = new Random((int) DateTime.Now.Ticks);
    DateTime  start;
    TimeSpan  span;
    while (true)
    {
     int read, total = 0;
     try
     {
      start = DateTime.Now;
      // створюємо запрос
      if (!preventCaching)
      {
       req = (HttpWebRequest) WebRequest.Create(source);
```

```
      }
      else
      {
req = (HttpWebRequest) WebRequest.Create(source +
((source.IndexOf('?') == -1) ? '?' : '&') + "fake=" + rnd.Next().ToString());
      }
      // встановлюємо логін та пароль
      if ((login != null) && (password != null) && (login != ""))
       req.Credentials = new NetworkCredential(login, password);
      // встановлюємо найменування групи підключення
      if (useSeparateConnectionGroup)
       req.ConnectionGroupName = GetHashCode().ToString();
      // отримуємо відповідь
      resp = req.GetResponse();
      // отримуємо відповідь потоку
      stream = resp.GetResponseStream();
      // цикл
      while (!stopEvent.WaitOne(0, true))
      {
// перевіряємо загальне читання
       if (total > bufSize - readSize)
       {
        total = 0;
       }
// Читаємо наступний блок у потоці
       if ((read = stream.Read(buffer, total, readSize)) == 0)
        break;
       total += read;
// Додаємо лічильник зчитаних байт
       bytesReceived += read;
      }
      if (!stopEvent.WaitOne(0, true))
      {
       // додаємо лічильник фреймів
       framesReceived++;
// остановка читання зображення
       if (NewFrame != null) {
        Bitmap bmp = (Bitmap) Bitmap.FromStream
        (new MemoryStream(buffer, 0, total));
// Клієнт увідомлення
        NewFrame(this, new CameraEventArgs(bmp));
// Будуємо картинку
        bmp.Dispose();
```

```
       bmp = null; } }
// Чекаємо в циклі
     if (frameInterval > 0)
     {
// діапазон часу
      span = DateTime.Now.Subtract(start);
// засипання у мілісекундах
      int msec = frameInterval - (int) span.TotalMilliseconds;
      while ((msec > 0) && (stopEvent.WaitOne(0, true) == false))
      {
// засипання...
       Thread.Sleep((msec < 100) ? msec : 100);
       msec -= 100;
      }  } }
    catch (WebException ex)
    {
  System.Diagnostics.Debug.WriteLine("=====: " + ex.Message);
// чекаємо у циклі наступну спробу
     Thread.Sleep(250);    }
    catch (Exception ex)
    {    System.Diagnostics.Debug.WriteLine("=========: " + ex.Message);
    }    finally  {
// помилка запиту
     if (req != null) {
      req.Abort();
      req = null;     }
// закриваємо потоку
     if (stream != null)     {
      stream.Close();
      stream = null;     }
// close response
     if (resp != null)     {
      resp.Close();
      resp = null;
     }}
// need to stop ?
    if (stopEvent.WaitOne(0, true))
     break;    }   }  } }
```

Алгоритм керування багатопотоковим відео

У конференціях із декількома учасниками однакові параметри відео не завжди потрібні для всіх одержувачів. Наприклад, потік активного доповідача

може відображатися у великому вікні, тоді як відео інших користувачів займає невеликі області екрана. Це створює можливість зменшення обсягу переданих даних шляхом використання декількох рівнів якості.

Для таких сценаріїв може застосовуватися Simulcast, що передбачає передавання декількох закодованих варіантів одного відеоджерела. Кожний варіант характеризується власними параметрами роздільної здатності та інтенсивності передавання.

Під час початкового створення відповідного RTP-відправника визначаються параметри доступних кодувань. Якщо використовується addTransceiver(), конфігурація може містити декілька записів sendEncodings. Для їх ідентифікації застосовуються відповідні ідентифікатори RID.

Після початку конференції передбачені кодування можуть активуватися або деактивуватися відповідно до підтримуваних можливостей. При цьому змінювати кількість і порядок уже узгоджених кодувань через звичайний виклик setParameters() не допускається.

У конференції з медіасервером типу Selective Forwarding Unit (SFU) сервер може вибирати відповідний варіант потоку для кожного одержувача. Клієнтський модуль у такому випадку повинен взаємодіяти з механізмом підписки на потоки або керування їхньою якістю.

Альтернативним підходом є масштабоване відеокодування Scalable Video Coding (SVC). Воно передбачає формування шарів відеоданих, які можуть використовуватися для отримання різних рівнів якості. Можливість застосування SVC залежить від підтримки кодека, браузера та відповідного серверного компонента.

Для конференцій із прямими з'єднаннями між усіма учасниками можливості централізованого вибору шарів обмежені. За збільшення кількості користувачів зростає навантаження на вихідний канал кожного клієнта. Тому для багатосторонніх конференцій доцільно розглядати архітектуру з SFU. При цьому конкретний спосіб серверної організації не може вважатися встановленим без відомостей про фактичну реалізацію ВКЗ.

Алгоритм керування демонстрацією екрана

Додатковою функцією клієнтського модуля може бути передавання вмісту екрана або окремого вікна застосунку. Цей режим відрізняється від звичайного передавання відео з камери, оскільки для демонстрації тексту й графічних матеріалів важливішою може бути деталізація зображення, ніж висока частота кадрів.

Для захоплення екрана використовується getDisplayMedia(). Перед початком демонстрації браузер отримує відповідний дозвіл користувача. Застосунок не повинен автоматично вибирати приховане джерело без передбаченої браузером процедури надання доступу.

Після отримання мультимедійного треку алгоритм перевіряє його стан та визначає спосіб передавання. Якщо конференція передбачає заміну відео з камери демонстрацією екрана, може застосовуватися replaceTrack(). За необхідності одночасного передавання обох потоків створюється окремий відправник або трансівер із відповідним узгодженням.

Для демонстрації екрана доцільно використовувати окремий профіль якості. Він може передбачати збереження достатньої роздільної здатності для читання тексту та обмеження частоти кадрів. Однак для демонстрації динамічного відео можуть знадобитися інші параметри.

Під час завершення демонстрації алгоритм повертає попередню конфігурацію. Якщо камера була тимчасово замінена екранним потоком, відновлюється відповідний відеотрек. Окремо обробляється ситуація, коли користувач припинив демонстрацію через стандартний інтерфейс браузера.

Алгоритм відновлення з'єднання після порушень зв'язку

Під час відеоконференції можливі тимчасова втрата доступу до мережі, зміна IP-адреси або недоступність окремого маршруту. Такі події не завжди потребують завершення конференції. Тому клієнтський модуль повинен передбачати алгоритм контрольованого відновлення зв'язку.

Вхідними даними є поточний стан RTCPeerConnection, зміни стану ICE та відомості про доступність сигнального каналу. На початковому етапі

визначається характер порушення. Тимчасовий стан disconnected відрізняється від підтвердженого failed, оскільки короткочасна перерва не завжди означає остаточну втрату транспортного маршруту.

У разі короткочасного порушення доцільно передбачити інтервал очікування відновлення. Якщо з'єднання повертається до працездатного стану, додаткові дії не виконуються. Це дозволяє уникати непотрібного повторного узгодження під час незначних мережевих коливань.

Якщо транспортне з'єднання не відновлюється, може виконуватися ICE restart. У браузерному середовищі відповідний механізм доступний через RTCPeerConnection.restartIce(). Після його виклику необхідно забезпечити повторний сигнальний обмін для погодження нових ICE-параметрів.

Під час відновлення важливо запобігати одночасному виконанню декількох несумісних процедур узгодження. Для цього контролюється стан сигналізації та використовується послідовне виконання відповідних операцій.

Якщо сигнальний канал також недоступний, необхідно спочатку відновити можливість обміну службовими повідомленнями. Повторні спроби доцільно виконувати з обмеженою частотою та поступовим збільшенням інтервалу між ними.

Після успішного відновлення WebRTC-з'єднання перевіряється стан мультимедійних треків. Якщо вони залишилися активними, можливе продовження передавання без повторного отримання доступу до камери. Якщо треки завершили роботу, необхідно виконати їх відновлення відповідно до дозволів користувача.

У випадку остаточної помилки застосунок повідомляє про неможливість продовження сеансу та переводить клієнтський модуль до визначеного стану. При цьому завершуються непотрібні спроби встановлення з'єднання і звільняються відповідні ресурси.

Алгоритм завершення конференції та звільнення ресурсів

Завершення роботи клієнтського модуля повинно забезпечувати коректне припинення мультимедійних операцій. Це необхідно для звільнення пристроїв захоплення та запобігання збереженню неактивних з'єднань.

Після отримання команди виходу з конференції система припиняє формування нових керувальних операцій. Далі зупиняється періодичне отримання статистики та робота алгоритму адаптивного керування якістю.

Наступним етапом є завершення активного WebRTC-з'єднання. Для цього використовується RTCPeerConnection.close(). Після закриття відповідний об'єкт не повинен повторно застосовуватися для встановлення нового сеансу.

Локальні мультимедійні треки зупиняються, якщо вони більше не використовуються іншими функціями застосунку. Це дозволяє звільнити камеру, мікрофон та інші джерела захоплення.

Після цього очищуються відомості про активних учасників і пов'язані елементи відтворення. Сигнальний сервер отримує повідомлення про вихід користувача відповідно до прийнятого протоколу взаємодії. Після завершення процедури клієнт переходить до початкового стану або відображає екран завершеного сеансу.

Особливості програмної реалізації та інформаційної безпеки

Програмна реалізація клієнтського модуля може здійснюватися з використанням браузерних інтерфейсів JavaScript та WebRTC. Конкретний фреймворк побудови інтерфейсу, спосіб організації сигнального сервера та структура програмних компонентів визначаються обраною архітектурою застосунку. За відсутності програмного коду ці технологічні рішення слід розглядати як можливий варіант реалізації.

Для зменшення взаємозалежності компонентів доцільно відокремити модуль керування конференцією від модуля оброблення статистики. Компонент адаптації повинен отримувати підготовлені показники через визначений програмний інтерфейс. Це дозволяє перевіряти його алгоритмічну логіку незалежно від фактичного мультимедійного сеансу.

Важливим аспектом є асинхронність більшості браузерних операцій. Отримання доступу до камери, створення SDP-опису та встановлення параметрів RTP можуть завершуватися із затримкою або помилкою. Тому необхідно обробляти результати асинхронних викликів і перевіряти актуальність стану перед виконанням наступних дій.

Засоби WebRTC передбачають захист передавання медіаданих за допомогою DTLS-SRTP. Водночас захист сигнального каналу та перевірка прав користувачів залишаються окремими завданнями системи ВКЗ. Для сигнального обміну доцільно використовувати захищені з'єднання та механізми перевірки доступу до конференції.

Використання медіасервера також потребує врахування моделі довіри. Застосування WebRTC не означає автоматичного забезпечення наскрізного шифрування між усіма користувачами конференції за наявності проміжної серверної обробки. Якщо така властивість є обов'язковою, необхідно передбачити відповідну додаткову архітектуру захисту.

Окремо необхідно контролювати доступ до локальних пристроїв. Камера, мікрофон і демонстрація екрана повинні активуватися лише за відповідного дозволу користувача. Після завершення конференції застосунок повинен припиняти захоплення даних.

Перевірка коректності алгоритмів та оцінювання якості

Для дослідження ефективності запропонованих алгоритмів необхідно передбачити перевірку роботи клієнтського модуля за різних характеристик мережі. На початковому етапі контролюється правильність встановлення WebRTC-з'єднання та передавання мультимедійних потоків за стабільних умов.

Далі перевіряється функціонування механізму отримання статистики. Отримані показники повинні відповідати фактичному стану мультимедійних потоків. Особливу увагу необхідно приділити коректності обчислення інтенсивності передавання та обробленню випадків відсутності окремих статистичних полів.

Для перевірки алгоритму адаптації можуть використовуватися контрольовані обмеження пропускної здатності. Поступове зменшення доступного мережевого ресурсу дозволяє дослідити реакцію клієнта на погіршення умов. Під час випробувань необхідно фіксувати момент зміни профілю, фактичні характеристики відео та стан аудіоканалу.

Окремо доцільно досліджувати вплив втрат пакетів і збільшення затримки. Такі сценарії дозволяють визначити, чи здатний алгоритм відрізняти короткочасні коливання від стійкого перевантаження. При цьому важливо враховувати особливості внутрішніх механізмів WebRTC, які також реагують на зміну мережевих умов.

Для оцінювання стабільності адаптації необхідно контролювати частоту перемикань між профілями. Надмірна кількість змін може погіршувати сприйняття відео навіть за прийнятної середньої якості. Тому важливими показниками є не лише час реакції, а й тривалість перебування в кожному режимі.

Випробування відновлення зв'язку можуть включати тимчасове відключення мережі, зміну доступного маршруту та перезапуск сигнального з'єднання. Для кожного випадку необхідно перевірити можливість повторного встановлення медіасеансу та коректність оброблення станів клієнтського модуля.

Під час перевірки на різних пристроях необхідно враховувати відмінності у підтримці браузерних функцій. Зокрема, параметри RTP-кодування та доступність окремих статистичних полів можуть відрізнятися. Тому результати випробувань бажано фіксувати із зазначенням браузера, операційної системи та характеристик клієнтського обладнання.

Практичну оцінку якості можна здійснювати за показниками частоти відтворення кадрів, затримки, стабільності передавання та кількості переривань. Для порівняння алгоритмів доцільно використовувати однакові мережеві сценарії. Це дозволяє визначати вплив саме механізму адаптації, а не випадкових змін умов експерименту.

Запропоноване алгоритмічне забезпечення охоплює основні процеси роботи клієнтського модуля ВКЗ від підготовки мультимедійних пристроїв до

завершення конференції. Центральне місце займає адаптивне керування якістю відео на основі статистичних показників WebRTC.

Поєднання контролю мережевого стану, зміни параметрів кодування та механізмів відновлення з'єднання створює основу для підтримання працездатності відеозв'язку в умовах змінного навантаження.

Остаточне визначення ефективності запропонованих рішень потребує експериментальної перевірки з урахуванням характеристик мережі, браузерного середовища та клієнтських пристроїв.

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Клієнтський модуль системи відеоконференцзв’язку (ВКЗ) на базі технології WebRTC призначений для організації аудіовізуальної взаємодії користувачів у реальному часі. Його функціональність охоплює підключення до конференції, захоплення та передавання мультимедійних потоків, відтворення відео інших учасників і адаптивне керування якістю передавання. Особливістю програмного забезпечення є можливість коригування параметрів відеопотоку залежно від характеристик мережевого з'єднання. Головне вікно програми показане на рисунку 5.1.

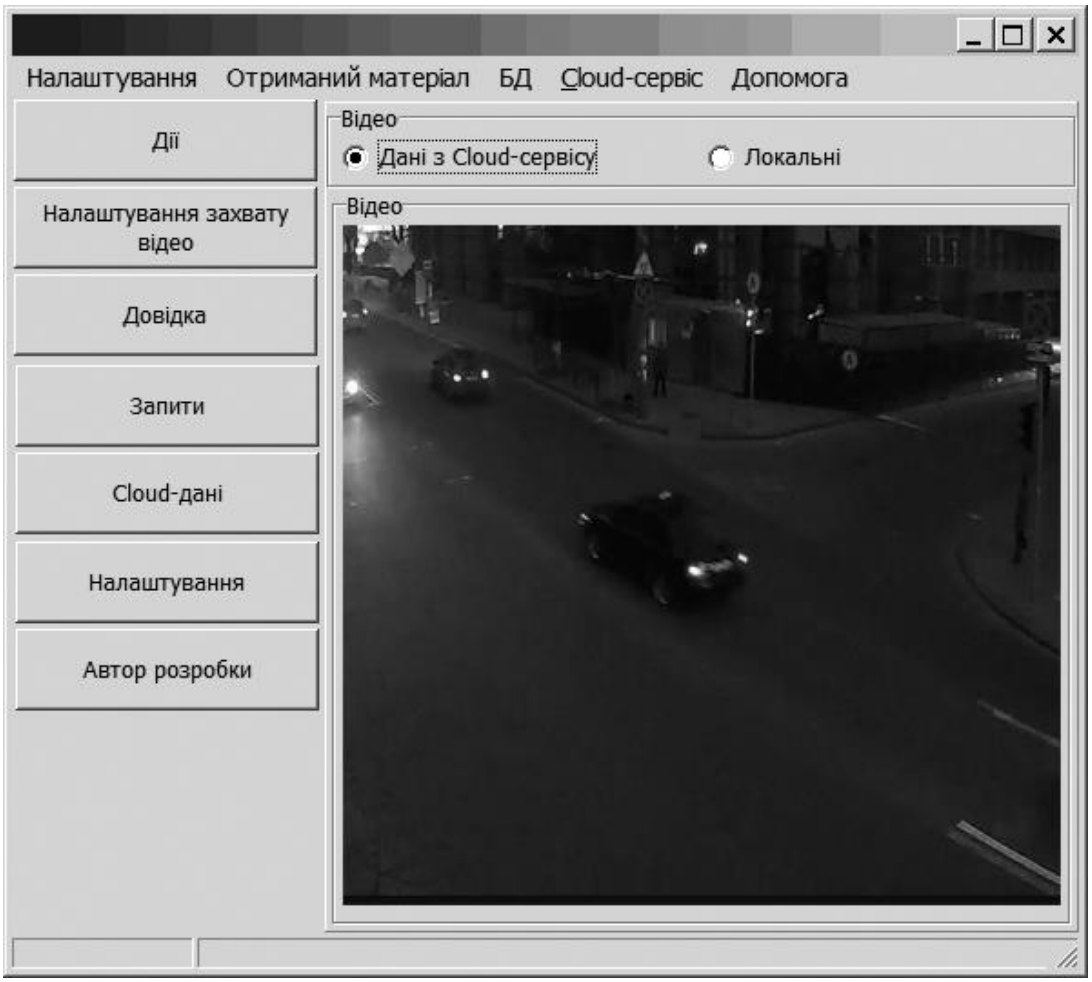


Рисунок 5.1 – Головне вікно ПЗ

Використання клієнтського модуля передбачає роботу у веббраузері з підтримкою WebRTC. Такий підхід дозволяє організувати мультимедійний обмін без встановлення окремого застосунку на комп'ютер користувача. Для роботи необхідні камера, мікрофон, пристрій відтворення звуку та доступ до мережі. Окремі функції можуть використовуватися без відеокамери, якщо передбачено участь у конференції лише з аудіозв'язком.

Програмну основу клієнтського модуля становлять браузерні інтерфейси WebRTC, які можуть використовуватися з JavaScript. Інтерфейс MediaDevices забезпечує отримання мультимедійних потоків від локальних пристроїв. RTCPeerConnection призначений для встановлення та підтримання з'єднання між учасниками. Для контролю характеристик передавання застосовується getStats(), а зміна допустимих параметрів кодування може здійснюватися через RTCRtpSender. Конкретні бібліотеки та фреймворки користувацького інтерфейсу визначаються фактичною програмною реалізацією.

Перед впровадженням необхідно підготувати серверну інфраструктуру, яка забезпечуватиме сигнальну взаємодію користувачів. Сигнальний сервер використовується для обміну повідомленнями встановлення з'єднання та керування сеансами конференцій. Одним із можливих способів його реалізації є використання WebSocket. Сервер повинен підтримувати обмін SDP-повідомленнями та ICE-кандидатами між відповідними учасниками. Для багатосторонніх конференцій може додатково використовуватися медіасервер типу Selective Forwarding Unit (SFU), якщо його передбачено архітектурою.

Підготовка середовища починається з перевірки сумісності браузерів із необхідними функціями WebRTC. Особливу увагу потрібно приділити підтримці відеокодеків, механізмів керування RTP-потоками та отримання статистики. Оскільки доступ до мультимедійних пристроїв у браузерах залежить від захищеного контексту, робоче розгортання вебзастосунку доцільно організовувати через HTTPS. Локальне тестування може виконуватися з урахуванням передбачених браузером винятків для локального середовища.

Наступним етапом є налаштування адрес сервісів. У конфігурації клієнтського модуля визначаються адреса сигнального сервера та параметри ICE-серверів. Для визначення можливих мережевих маршрутів використовується STUN, а для ретрансляції мультимедійного трафіку за відсутності прямого з'єднання може застосовуватися TURN. Параметри доступу до TURN-сервера повинні передаватися захищеним способом. Їх не слід включати у відкритий вихідний код у вигляді постійних секретних облікових даних.

Після підготовки серверної частини виконується налаштування параметрів мультимедійного захоплення. Для камери визначаються початкові обмеження роздільної здатності та частоти кадрів. Вони повинні враховувати можливості пристрою і не вимагати обов'язкової підтримки максимальних характеристик. Для мікрофона можуть використовуватися доступні браузерні засоби оброблення звуку. Якщо запитані параметри не підтримуються, застосунок повинен передбачати вибір допустимого режиму.

Окремого налаштування потребує компонент адаптивного керування якістю відеопотоку. Для нього визначаються доступні профілі якості, інтервал отримання статистики та правила переходу між профілями. Як контрольовані показники розглядаються інтенсивність передавання, втрати пакетів, мережева затримка та фактична частота кадрів. Порогові значення необхідно уточнювати за результатами тестування, оскільки вони залежать від характеристик мережі та використовуваних відеокодеків.

Для налаштування вихідного потоку можуть використовуватися параметри maxBitrate, maxFramerate та scaleResolutionDownBy. Вони дозволяють задавати обмеження швидкості передавання, частоти кадрів і масштабу роздільної здатності. Перед використанням цих можливостей перевіряється їх підтримка браузером. Зміна параметрів не повинна спричиняти постійного повторного встановлення WebRTC-з'єднання. Для запобігання частим перемиканням між профілями доцільно використовувати окремі умови зниження та підвищення якості.

Рекомендована послідовність введення програмного комплексу в роботу передбачає початковий запуск сигнального сервера та перевірку його мережевої доступності. Після цього перевіряється працездатність STUN- і TURN-сервісів, необхідних для встановлення транспортних з'єднань. Якщо архітектура передбачає SFU, його необхідно налаштувати до проведення багатосторонніх сеансів. Завершальним етапом є розгортання клієнтського вебзастосунку та відкриття його через браузер.

Для оцінювання роботи адаптивного механізму необхідно перевірити отримання статистичних показників через getStats(). У контрольованому середовищі можуть створюватися обмеження пропускної здатності та додаткові мережеві затримки. За результатами спостережень перевіряється реакція системи на погіршення умов передавання. Очікуваною поведінкою є збереження працездатності конференції з відповідним зниженням параметрів відео, якщо це допускають поточні мережеві умови.

Окремо перевіряється відновлення якості після усунення мережевих обмежень. Підвищення параметрів повинно відбуватися після підтвердження стабільності з'єднання. Необхідно також оцінити відсутність надмірно частих перемикань профілів. Для цього аналізуються зміни фактичної частоти кадрів, інтенсивності передавання та тривалості перебування в кожному режимі.

Перед переходом до регулярної експлуатації необхідно перевірити роботу застосунку в підтримуваних браузерах, правильність оброблення дозволів на доступ до пристроїв та коректне звільнення ресурсів після завершення конференції. Додатково контролюються захист сигнального обміну, перевірка повноважень користувачів і відсутність неконтрольованого доступу до мультимедійних потоків. Результати перевірок використовуються для оцінювання готовності клієнтського модуля до практичного застосування.

Для ідентифікації програмного продукту передбачається представлення відомостей про його розробника. Такі відомості дозволяють встановити авторство програмного забезпечення та використовуються під час його

подальшого супроводження. Авторські дані клієнтського модуля системи відеоконференцзв'язку представлені на рисунку 5.2.

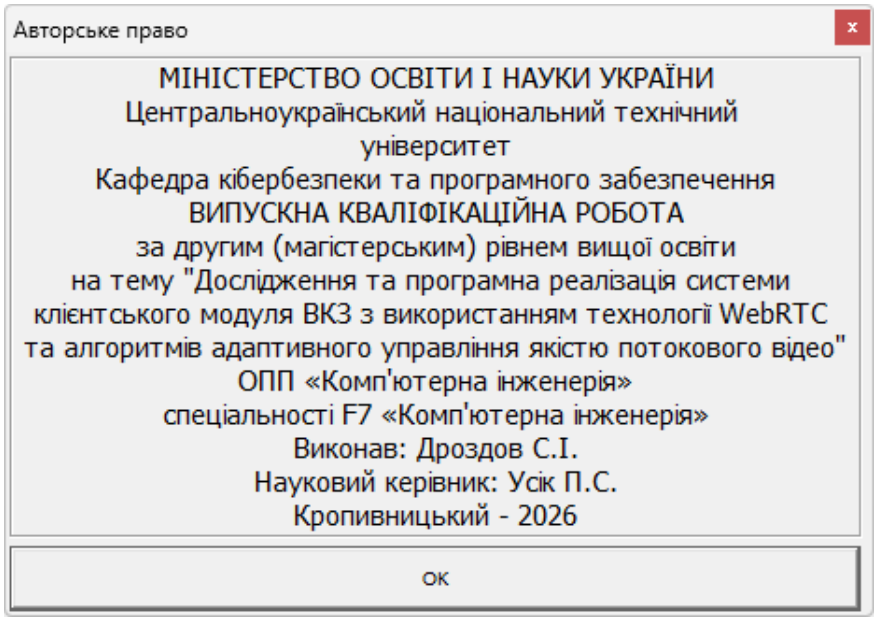


Рисунок 5.2 – Форма авторського права

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи клієнтського модуля ВКЗ з використанням технології WebRTC та алгоритмів адаптивного управління якістю потокового відео.

Метою розробки є дослідження та програмна реалізація системи клієнтського модуля ВКЗ з використанням технології WebRTC та алгоритмів адаптивного управління якістю потокового відео.

Об'єктом дослідження є процес клієнтського модуля ВКЗ з використанням технології WebRTC та алгоритмів адаптивного управління якістю потокового відео.

Предметом дослідження є методи клієнтського модуля ВКЗ з використанням технології WebRTC та алгоритмів адаптивного управління якістю потокового відео.

Методи дослідження базуються на методах теорії кодування, методах комп'ютерних мереж, методах математичної статистики, методах розробки програмного забезпечення.

Наукова новизна отриманих результатів. У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

– Удосконалено метод клієнтського модуля ВКЗ з використанням технології WebRTC та алгоритмів адаптивного управління якістю потокового відео.

– Розроблено вітчизняний продукт клієнтського модуля ВКЗ з використанням технології WebRTC та алгоритмів адаптивного управління якістю потокового відео, який має більш широкі можливості, на відміну від існуючих аналогів.

7. ЗАХОДИ ЩОДО ОХОРОНИ ПРАЦІ І ТЕХНІКИ БЕЗПЕКИ

7.1 Вступ

Законом України “Про охорону праці” [3] регламентуються загальні положення державної політики в галузі охорони праці, а реалізуються ці положення нормативно-правовими актами про охорону праці, зокрема Наказом Міністерства соціальної політики України № 207 від 25.04.2018 р. та №508/31960 «Про затвердження Вимог щодо безпеки та захисту здоров’я працівників під час роботи з екранними пристроями» [5], яким затверджено нормативно-правовий акт з охорони праці НПАОП 0.00-7.15-18, «Правила охорони праці під час експлуатації електронно-обчислювальних машин», та «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-97.

Програмісти у процесі роботи мають негативний вплив на органи зору, а також мають значну розумову напругою і нервово-емоційне навантаження. Руки (суглоби пальців та м'язи рук) при роботі з клавіатурою мають теж істотне навантаженням. До шкідливих факторів, які впливають на робітників галузі інформаційних технологій (ІТ) спеціалісти відносять високочастотні електромагнітні коливання (випромінювання) роботи апаратної частини ЕОМ та виділення шкідливих газів.

Ці шкідливі фактори можуть привести до професійних захворювань.

Керуючись нормативно-правовими актами «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98 [5], та «Вимоги щодо безпеки та захисту здоров’я працівників під час роботи з екранними пристроями» НПАОП 0.00-7.15-18 розглянемо шкідливі чинники роботи персоналу.

Щоб запропонувати заходи щодо зменшення негативного впливу комп’ютера на організм людини визначимо фактори, які можуть викликати професійне захворювання і впливають на працездатність програміста.

7.2 Шкідливі і небезпечні фактори при роботі з комп’ютером

Працівники, задіяні на роботах, пов'язаних з періодичною або постійною роботою за комп'ютером, піддаються впливу факторів виробничої небезпеки, основними з яких є наступні:

1. Фізичні:

– Підвищений рівень напруги в електричному ланцюзі, замикання якої може пройти через тіло працюючого.

– Підвищений рівень рентгенівського випромінювання.

– Підвищений рівень ультрафіолетового випромінювання.

– Підвищений рівень інфрачервоного випромінювання.

– Можливість ураження статичною електрикою.

– Запиленість повітря робочого приміщення.

– Підвищений вміст важких (+) аероіонів.

– Нерівномірний розподіл яскравості в полі зору.

– Підвищений рівень пульсації світлового потоку.

2. Хімічні: підвищений вміст у повітрі вуглекислого газу, озону, аміаку, фенолу, формальдегіду та інше.

3. Психофізіологічні:

– Напруга зору.

– Напруга пам'яті.

– Напруга уваги..

– Тривале статичне напруження.

– Підвищений обсяг інформації, що обробляється в одиницю часу.

– Монотонність праці в окремих випадках.

– Нераціональна організація робочого місця.

До основних шкідливих факторів при роботі з комп'ютером відносять: тривале сидяче положення, електромагнітне випромінювання, навантаження на зір, перевантаження кистьових суглобів, можливість захворювань органів дихання, алергії, порушення нормального перебігу вагітності та інше.

7.3 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста

Розглянемо умови праці у приміщенні, в якому працюють програмісти. Геометричні розміри приміщення наведено у таблиці 7.1.

Таблиця 7.1 – Розміри приміщення

Найменування	Значення, м
Ширина	4
Довжина	5
Висота	3

Таблиця 7.2 – Площа та обсяг приміщення, на одного працюючого*

Геометрична характеристика	Одиниця виміру	Нормативне значення*	Фактичне значення
Площа, S	$м^2$	не менше 6.0	6,6
Об'єм, V	$м^3$	не менше 20.0	20

* Згідно ДСанПіН 3.3.2.007-98 (Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин).

У зазначеному приміщенні працюють троє людей. За даними, які наведено у таблиці 7.1, та таблиці 7.2, можна зробити висновок, що площа та об’єм приміщення у розрахунку на одно робоче місце програміста не відповідають нормативним вимогам ДСанПіН 3.3.2-007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» [5], але відповідають нормативним вимогам Наказу Міністерства соціальної політики України № 207, від 14.02.2018 «Про затвердження Вимог щодо безпеки та захисту здоров’я працівників під час роботи з екранними пристроями» [5] та НПАОП 0.00-1.28-10 «Правила охорони

праці під час експлуатації електронно-обчислювальних машин»). Таким чином можна зробити висновок, що санітарно-гігієнічні умови праці на робочому місці програміста відповідають вимогам.

Температура повітря в приміщенні визначається впливом температури зовнішнього повітря і тепловою енергією, яка виділяється всередині приміщення. Джерелами виділення теплоти в даному приміщенні є електроустаткування, освітлювальні прилади, а також люди. У світлий час доби джерелом надлишкового тепла є сонячна радіація. Згідно Постанови № 42 від 01.12.1999 Головного державного санітарного лікаря України, робота, виконувана в даному приміщенні, відноситься до категорії Іа. В цьому випадку людина витрачає енергії до 120 ккал у годину. Вологість повітря в приміщенні визначається впливом багатьох факторів, серед яких: вологість атмосферного повітря, виділення вологи людьми (при диханні та випарами з поверхні шкіри).

Мікроклімат повітряного середовища в приміщенні характеризується запиленістю та загазованістю повітря. Мікроклімат приміщення визначається діючим на організм людини поєднанням, вологості, температури, швидкості руху повітря та інтенсивності теплового випромінювання. Аналіз мікроклімату складається з визначення зазначених вище факторів і порівняння результатів із встановленими нормами.

У таблиці 7.3 наведено оптимальні та фактичні значення параметрів мікроклімату як для категорії ваги робіт Іа, так і розглянутого приміщення. У приміщеннях, де встановлено ЕОМ, рекомендується застосування тільки оптимальних значень показників мікроклімату.

Проведений аналіз показує, що показники мікроклімату в приміщенні відповідають установленим нормам. Штучне опалення застосовується у холодний період року.

В літню пору застосовується кондиціонер.

Для боротьби з пилом робляться регулярні провітрювання та вологі прибирання приміщенні.

Таблиця 7.3 – Оптимальні і фактичні значення параметрів мікроклімату

Пора року	Оптимальні для Іа			Фактичні		
	Температура, °С	Вологість, %	Швидкість повітря, м/с	Температура, °С	Вологість, %	Швидкість повітря, м/с
Холодна	22-24	40-60	0,1	21-22	46-54	0,1
Тепла	23-25	50-70	0,1	24-25	50-65	0,11

У приміщенні знаходяться наступні джерела шуму: принтер *XEROX PHASER 3020BI*, електродвигуни вентиляторів ЕОМ.

Одним з найважливіших факторів, які впливають на ефективність трудової діяльності людини, та попереджають травматизм і професійні захворювання програмістів є освітлення на робочому місці.

З 2019 року діють Державні будівельні норми України “Природне і штучне освітлення” – ДБН В.2.5-28:2018 [1], у яких прописані вимоги до використання всіх освітлювальних приладів, у тому числі світлодіодних.

Працю працівника, який постійно працює за комп’ютером, згідно ДБН В.2.5-28:2018 [1], можна віднести до роботи з малою точністю (найменший розмір об'єкта має розрізнення від 1 до 5 мм) V-го розряду зорової роботи, з великою контрастністю об'єкта розрізнення (символів на екрані дисплея), з темним тлом (підрозряд зорової роботи В).

Приміщення можна віднести до 1-ої групи приміщень, у яких проводиться розрізнення об'єктів зорової роботи при фіксованому напрямку лінії зору того, що працює на робочу поверхню.

Для такого типу приміщень і розряду зорової роботи нормоване значення коефіцієнта природної освітленості (КПО) робочої поверхні (при поєднаному, спільному освітленні), повинен становити не більше 1,5%, освітленість при штучному висвітленні повинна становити 300 Лк. [1], Крім того все поле зору повинне бути освітлено достатньо рівномірно – ця основна гігієнічна вимога.

Оскільки яскраве світло на ділянці периферійного зору значно збільшує напруженість очей і, як наслідок, призводить до їх швидкої стомлюваності, ступінь освітлення приміщення і яскравість екрану комп'ютера повинні бути приблизно однаковими.

7.4 Розробка заходів з умов поліпшення охорони праці

Згідно аналізу умов праці в розглянутому приміщенні, ми одержали наступні результати:

– розміри приміщення, у розрахунку на одному працюючого, відповідають нормативам;

– мікроклімат відповідає нормативному значенню;

– акустичні умови роботи не перевищують нормативних значеннь;

Таким чином можна припустити, що основною причиною можнивого зниження працездатності програміста є психофізіологічний фактор, тому основна пропозиція буде така: дотримання позитивної психологічної атмосфери в колективі та регламентованого режиму праці та відпочинку, організація робочого місця з урахуванням ергономічних вимог.

Рекомендовані заходи: регулярні періодичні наочні огляди персоналом шляхів для евакуації людей із приміщення, відповідно до плану евакуації (який повинен розташовуватись на видному місці у приміщенні), включення до колективного договору мінімально можливого вмісту аптечок з обов’язковою наявністю масок-клапанів, або іншого спорядження для штучного дихання. Регулярна періодична перевірка параметрів заземлення та занулення (вимірювання опору ланцюга).

Регулярна наочне знайомство персоналу із шляхами для евакуації людей із приміщення відповідно до плану евакуації, забезпечення розподільних щитів спеціальними розетками з заземлюючими контактами; організація заземлення всіх приладів і пристроїв, які працюють при напрузі вище 36 В.

7.5 Розрахункова частина

Проведемо розрахунок штучного освітлення за методом коефіцієнту використання світлового потоку для приміщення ширина якого складає 4 м, довжина – 5 м, висота – 3 м.

Для того, щоб визначити потрібну кількість світильників, які повинні забезпечити нормований рівень освітленості, необхідно враховувати, що, з одного боку середня освітленість робочих місць з постійним перебуванням людей повинна бути не менш як 200 люкс, а з іншого – штучне освітлення при системі комбінованого освітлення для зорової роботи найвищої точності повинна складати 300 лк та вимогу ДБН В.2.5-28:2018, що створювати освітленість більше ніж 300 лк при світлодіодних світильникх дозволяється тільки за наявності обґрунтування.

Таким чином для розрахунку приймаэмо середню освітленість робочих місць 300 Люкс [1].

Визначимо світловий потік, що падає на робочу поверхню за формулою [6]:

$$F = E \cdot S \cdot K \cdot Z / n,$$

де:

F – світловий потік, що розраховується, Лм;

E – нормована мінімальна освітленість, Лк; E = 300 Лк. [1];

S – площа поверхні, на яку падає світловий потік, м^2 [1];

Z – відношення середньої освітленості до мінімальної (зазвичай приймається рівним 1.1... 1.2, у нашому випадку Z = 1,1);

K – коефіцієнт запасу, що враховує зменшення світлового потоку світильників в результаті їх забруднення в процесі експлуатації (його значення залежить від типу приміщення і характеру робіт, що проводяться в ньому, в нашому випадку K = 1,5) [1];

n – коефіцієнт використання світлового потоку, (відношення світлового потоку, що падає на розрахункову поверхню, до сумарного потоку всіх ламп і

обчислюється в долях одиниці; залежить від характеристик світильника, розмірів приміщення, забарвлення стін і стелі, що характеризуються коефіцієнтами відбиття від стін ($\rho_{стін.}$) і стелі ($\rho_{стелі}$), значення коефіцієнтів дорівнюють $\rho_{стін}$ = 50% і $\rho_{стелі}$ = 50% [6].

Обчислимо індекс приміщення за формулою:

$$i = S/(h\cdot (A+B)),$$

де:

S – площа поверхні, на яку падає світловий потік, м^2 [1];

h – розрахункова висота підвісу, h = 3 м;

A – ширина приміщення, A = 4 м;

B – довжина приміщення, B = 5 м.

Підставимо всі значення у формулу та визначимо індекс приміщення: i=0,74.

Знаючи індекс приміщення, знаходимо n = 0.29 (з табличних даних коефіцієнтів використання світлового потоку (n) світильників відповідного типу [6]). Підставимо всі значення у формулу, визначимо світловий потік: F=43043 Лм.

Будемо використовувати світильники (світлодіодні панелі) Matt White Армстронг 36W 6000К, світловий потік яких $F_л$ = 3000 Лм.

Число ламп визначається по формулі:

$$N = F/F_л$$

де:

F – світловий потік,

$F_л$ – світловий потік одного світильника.

Підставимо всі значення у формулу та визначемо індекса приміщення:

$$N = 43043/ 3000=14,34 \text{ шт.}$$

Приймаємо необхідну кількість світильників 15 шт.

Висновки до розділу

Дотримання всіх необхідних умов праці не лише сприяє збереженню здоров’я працівників, а також підвищує ефективність виробництва в цілому.

З цих міркувань було здійснено аналіз приміщення, призначеного для праці програмістів, проведено розгляд небезпечних та шкідливих факторів, що негативно впливають на програмістів під час роботи. Виконано розрахунок штучного освітлення, як одного з ключових факторів впливу на працездатність та здоров’я програміста. Розроблено заходи з охорони праці.

8 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи клієнтського модуля ВКЗ з використанням технології WebRTC та алгоритмів адаптивного управління якістю потокового відео.

В межах України в недостатній мірі представлені вітчизняні розробки в цій області.

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти наведені теоретичне узагальнення й рішення наукового завдання дослідження методів клієнтського модуля ВКЗ з використанням технології WebRTC та алгоритмів адаптивного управління якістю потокового відео.

Рішення даного завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем клієнтського модуля ВКЗ з використанням технології WebRTC та алгоритмів адаптивного управління якістю потокового відео.

– Досліджена система клієнтського модуля ВКЗ з використанням технології WebRTC та алгоритмів адаптивного управління якістю потокового відео.

– На основі отриманих результатів досліджень створена програмна реалізація системи клієнтського модуля ВКЗ з використанням технології WebRTC та алгоритмів адаптивного управління якістю потокового відео.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання клієнтського модуля ВКЗ з використанням технології WebRTC та алгоритмів адаптивного управління якістю потокового відео.

Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності

предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Розроблене програмне забезпечення має простий, дружній та зручний інтерфейс користувача, що забезпечує легкість у освоєнні роботи програмного продукту, зручність у використанні, і не потребує особливих спеціальних знань.

При створені програмного забезпечення було використано об'єктно-орієнтований підхід, що відповідає сучасним тенденціям у галузі розробки комерційних програмних систем.

Програма реалізована на мові високого рівня Visual C#. Дана мова програмування дозволяє найбільш ефективно обробляти дані. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на загальне програмне забезпечення, що поставляється із засобами обчислювальної техніки й спеціальне програмне забезпечення, що спеціально розроблене для даної конкретної системи й включає програми, що реалізують її функції.

Програма призначена для виконання під управлінням багатозадачної операційної системи Windows 10/11.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

В цілому створене програмне забезпечення підтверджує правильність використаних проектних рішень та повністю відповідає вимогам технічного завдання. Створене програмне забезпечення має потенційну можливість для подальшого вдосконалення і застосування у різних галузях.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Дроздов С.І. Дослідження та програмна реалізація системи клієнтського модуля ВКЗ з використанням технології WebRTC та алгоритмів адаптивного управління якістю потокового відео // Збірник праць молодих науковців ЦНТУ. – Вип. 17. – Кропивницький: ЦНТУ, 2027.

2. Ranjan Parekh. Fundamentals of Image, Audio, and Video Processing Using MATLAB®With Applications to Pattern Recognition. CRC Press. 2021. 406 p.

3. Alasdair McAndrew. A Computational Introduction to Digital Image Processing. Chapman & Hall. 2021. 560 p.

4. Михайло Пічугін, Іван Канкін, Володимир Воротніков Комп'ютерна графіка. Навчальний посібник / Центр навчальної літератури 346 с. 2019р.

5. Маценко В.Г. Комп’ютерна графіка: Навчальний посібник. – Чернівці: Рута, 2009 – 343 с.

6. Інженерна комп’ютерна графіка: підручник / В.В. Проців [та ін.] / М-во освіти і науки України, Нац. гірн. унт-т. – Дніпро: НГУ, 2017. – 247 с.

7. Проців В.В. Прикладна комп'ютерна графіка [Текст]: Навч. посібник / В.В. Проців, К.А. Зіборов, К.М. Бас, Г.К. Ванжа; М-во освіти і наук, Нац. гірн. ун-т. – Д.: НГУ, 2016. – 187 с.

8. Kopf, Johannes and Lischinski, Dani. Depixelizing Pixel Art (англ.) // ACM Trans. Graph. – 2011. – Vol. 30, no. 4. – P. 99:1--99:8.

9. Giachetti, Andrea and Asuni, Nicola. Real-Time Artifact-Free Image Upscaling (англ.) // Trans. Img. Proc.. – 2011. – Vol. 20, no. 10. – P. 2760—2768.

10. Богаченко, С., Смірнов, О., Буравченко, К., Смірнова, Т., Конопліцька-Слободенюк, О., Якименко, Н., Смірнов, С. «Обмеження ресурсів мобільних та edge-платформ при розробці програмного забезпечення для реалізації та оптимізації моделей машинного навчання». *Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка»*, 2026, т. 2 № 34, С. 677-686.

11. Смірнов, О., Ткачук, Р., Козірова, Н., Константинова, Л., Конопліцька-Слободенюк, О., Якименко, Н., Смірнов, С.. «Дослідження застосування векторних баз даних у генеративному штучному інтелекті». *Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка»*, 2026, Том 1 № 33, 667-682.

12. Смірнов О., Заріцкий В., Буравченко К., Конопліцька-Слободенюк О., Константинова Л., Якименко Н., Смірнов С. «Оптимізація розпізнавання облич за допомогою CUDA прискореної бібліотеки dlib». *Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка»*, 2026, Том 4, № 32, С. 573-582.

13. Kuznetsov, O., Frontoni, E., Kryvinska, N., Chevardin, V., Smirnov, O. «Wireless Network Encryption Stream Ciphers, Computational Modeling, and Security Analysis». *Computational Modeling and Simulation of Advanced Wireless Communication Systems*, 2024, pp. 379–402.

14. Kuznetsov, O., Frontoni, E., Kryvinska, N., Smirnov, O., Imoize, G.L. «Computational Modeling of Enhanced Spread Spectrum Codes for Asynchronous Wireless Communication». *Computational Modeling and Simulation of Advanced Wireless Communication Systems*, 2024, pp. 403–447

15. Смірнова Т.В., Конопліцька-Слободенюк О.К., Буравченко К.О., Смірнов С.А., Кравчук О.В., Козірова Н.Л., Смірнов О.А. «Дослідження технологій забезпечення кібербезпеки хмарних сервісів IaaS, PaaS та SaaS». *Кібербезпека: освіта, наука, техніка.* 2024. №4(24), С. 6-27.

16. Батрак О., Смірнова Т., Гнатюк В., Одарченко Р., Смірнов О. «Дослідження показників ефективності функціонування та перспектив розвитку систем IP-телефонії». *Підводні технології,* 2024, № 13, с. 28-35.

17. Al-Mudhafar Aqeel, A.M., Smirnova, T., Buravchenko, K., Smirnov, O. «The method of assessing and improving the user experience of subscribers in software-configured networks based on the use of machine learning». *Advanced Information Systems*, 2023, 7(2), pp. 49-56.

18. Smirnov, O., Sydorenko, V., Aleksander, M., Zhyharevych, O., Yenchev,

S. «Simulation of the cloud IoT-based monitoring system for critical infrastructures». *CEUR Workshop Proceedings,* Volume 3530, 2023, pp. 256-265.

19. Smirnov, O., Odarchenko, R., Smirnova, T., Bondar, S., Volosheniuk, D. «Optimal Structure Construction of Private 5G Network for the Needs of Enterprises». Lecture Notes on Data Engineering and Communications Technologies, 2023, 178, pp. 208–223.

20. Аль-Мудхафар Акіл Абдулхуссейн М., Смірнова Т.В., Буравченко К.О., Смірнов О.А. «Метод оцінки та підвищення користувальницького досвіду абонентів в програмно-конфігурованих мережах на основі використання машинного навчання». *Сучасні інформаційні системи,* 2023, том 7, № 2, С. 49-56.

21. Smirnov, O., Neskorodieva, T., Fedorov, E., Rudakov, K., Neskorodieva, A. «Method Detection Audit Data Anomalies on Basis Restricted Cauchy Machine» CEUR Workshop Proceedings, Volume 3187, 2022,

22. Smirnov O., Smirnova T., Anas M. Al-Oraiqat, Drieiev O., Polishchuk L., Sheroz Khan, Yassin M. Y. Hasan, Aladdein M. Amro, Hazim S. AlRawashdeh «Method for Determining Treated Metal Surface Quality Using Computer Vision Technology». Sensors (Basel, Switzerland) Volume 22, Issue 16, 6223, 2022.

23. Smirnov O., Kuznetsov A., Kryvinska N., Kiian A., Kuznetsova K. «Full Non-Binary Constant-Weight Codes». SN Computer Science, Vol 2, 337, 2021. https://doi.org/10.1007/s42979-021-00739-w

24. Smirnov O., Kuznetsov A., Zhora V., Onikiychuk A., Pieshkova O. «Hiding Messages in Audio Files Using Direct Spread Spectrum». 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2021, Cracow, Poland, 22-25 September 2021. P. 414-418.

25. Smirnov O., Kuznetsov A., Lokotkova I., Kuznetsova T., Florov S., Lebid O. «Using Orthogonal Signals to Hide Information in Images». 4 IEEE International Conference on Advanced Information and Communication Technologies (AICT) – 2021, Lviv, Ukraine, September 21-25, 2021. P. 255-260.

26. Smirnov, O., Kuznetsov, A., Potii, O., Poluyanenko, N., Stelnyk, I., Mialkovsky, D. «Combining and filtering functions in the framework of nonlinear-feedback shift register». International Journal of Computing; 2020, Volume 19, Issue 2 – Research Institute for Intelligent Computer Systems – 2020. – P. 247-256.

27. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». CEUR Workshop Proceedings. Volume 2740, 2020, Pages 102-114.

28. Smirnov O., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and cybersecurity of virtual cloud resources». Journal of theoretical and applied information technology Vol.98. No 21, 2020, P. 3334-3346.

29. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New technique for data hiding in cover images using adaptively generated pseudorandom sequences». CEUR Workshop Proceedings Volume 2654, 2020, Pages 1-14.

30. Smirnov O., Kuznetsov A., Onikiychuk A., Makushenko T., Anisimova O., Arischenko A. «Adaptive pseudo-random sequence generation for spread spectrum image steganography». 2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT), Ukraine, Kyiv, May 14-18. 2020. P. 161-165.

31. Smirnov O., Kuznetsov A., Kiian A., Cherep A., Kanabekova M., Chepurko I. «Testing of code-based pseudorandom number generators for post-quantum application». 2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT), Ukraine, Kyiv, May 14-18. 2020. P. 172-177.

32. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhiienko R., Babenko V., Kuznetsova T., «Representation of Cascade Codes in the Frequency Domain». In: Radivilova T., Ageyev D., Kryvinska N. (eds) Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies, vol 48. Springer, Cham. 2021. pp 557-587.

33. Smirnov, O., Drieieva, H., Drieiev, O., Polishchuk, Y., Brzhanov, R.,

Aleksander, M. «Method of fractal traffic generation by a model of generator on the graph». CEUR Workshop Proceedings Volume 2616, 2020, Pages 366-379.

34. Smirnov, O., Drieieva, H., Drieiev, O., Simakhin, V., Bondar, S., Odarchenko, R. «Managing multifractal properties of the binary sequence generated with the Markov chains», CEUR Workshop Proceedings Volume 2608, 2020, Pages 633-645.

35. Smirnov, O., Kuznetsov, A., Gorbacheva, L., Babenko, V., «Hiding data in images using a pseudo-random sequence», CEUR Workshop Proceedings Volume 2608, 2020, Pages 646-660.

36. Zhurakovskyi, B., Tsopa, N., Batrak, Y., Odarchenko, R., Smirnova, T «Comparative analysis of modern formats of lossy audio compression». Workshop Proceedings, 2020, 2654, стр. 315-327.

37. Smirnov O. Kuznetsov A., Zaicenko Yu., Pastukhov M., Oleshko O., Kuznetsova K., «Formation of Discrete Signals with Special Correlation Properties». International Conference on Information and Telecommunication Technologies and Radio Electronics, UkrMiCo 2019; Odessa; Ukraine; 9-13 September 2019. P.22-28.

38. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». International Journal of Computing; 2019, Volume 18, Issue 4 – Research Institute for Intelligent Computer Systems – 2019. – P. 393-407.

39. Smirnov, O., Kuznetsov, A., Reshetniak, O., Ivko, N., Katkova, T., Kuznetsova, T., «Generators of Pseudorandom Sequence with Multilevel Function of Correlation». 2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T), Kyiv, Ukraine, 8 – 11 October 2019 . P.517-522.

40. Smirnov, O., Krasnobayev, V., Yanko, A., Kuznetsova, T. «Methods of nulling numbers in the system of residual classes». CEUR Workshop Proceedings, Vol 2588, P. 90-106, 2019.

41. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Averchev, A., Pastukhov, M., Kuznetsova, K., «Formation of Pseudorandom Sequences with Special Correlation

Properties», 2019 3rd International Conference on Advanced Information and Communications Technologies, AICT -2019/ Lviv, Ukraine, 2-6 July, 2019, P. 395-399.

42. Smirnov, O., Kuznetsov, A., Kavun, S., Babenko, B., Nakisko, O., Kuznetsova, K., «Malware Correlation Monitoring in Computer Networks of Promising Smart Grids», 2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS), Kyiv, Ukraine April 17-19, 2019 P. 347-352.

43. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Pastukhov, M., Kuznetsova, K., Prokopovych-Tkachenko, D., «Discrete Signals with Special Correlation Properties», CEUR Workshop Proceedings Volume 2353, CEUR Workshop Proceedings 2019, Pages 618-629.

44. Smirnov A.A., Kuznetsov A.A., Danilenko D.A., Berezovsky A., «The statistical analysis of a network traffic for the intrusion detection and prevention systems», Telecommunications and Radio Engineering. – Volume 74, Issue 1. – Begel House Inc. – 2015. – P. 61-78.

45. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New Technique for Hiding Data in Cover Images Using Adaptively Generated Pseudorandom Sequences». CEUR Workshop Proceedings Volume 2732, 2020, Pages 214-227.

46. Т.В. Смірнова, О.М. Дрєєв, О.А. Смірнов «Хмарна інформаційна система оцінювання шорсткості з використанням дискретного частотного аналізу макрофотографій». IV міжнародна науково-практична конференція “Інформаційна безпека та комп’ютерні технології”, м. Кропивницький. 15-16 квітня 2021р. – Кропивницький: ЦНТУ. – 2021. – С. 30.

47. О.А. Смірнов, П.С. Усік, «дослідження перспектив використання технологічних рішень в мережах 5g» у Кібербезпека та інформаційні технології: монографія. – Х. : ТОВ «ДІСА ПЛЮС», 2020.С. 122-135.

48. О.А.Смірнов, Т.В.Смірнова, Л.І. Поліщук, К.О. Буравченко, А.О.Макевнін, «Дослідження хмарних технологій як сервісів», Кібербезпека: освіта, наука, техніка. № 3(7). С. 43-62. 2020.

49. Смірнов О.А., Дрєєва Г.М., Дрєєв О.М., Смірнова Т.В. «Фрактальний аналіз генератора самоподібного трафіку на основі ланцюга Маркова». Центральноукраїнський науковий вісник. Технічні науки. № 2(33). с. 161-172, 2019.

50. О. Смірнов, Є. Деменко, О. Онікійчук, А. Арищенко, Л. Горбачова, «Формування псевдовипадкових послідовностей для приховування даних в зображеннях» Комп'ютерні науки та кібербезпека. № 4. С. 30-37. 2019.

51. Смірнов О.А., Конопліцька-Слободенюк О.К., Смірнов С.А., Буравченко К.О., Смірнова Т.В. Поліщук Л.І. Проектування комп'ютерних систем та мереж. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2019. – 264 с.

52. Smirnov, O., Kuznetsov, A., Kuznetsova., K. Synthesis of Discrete Signals with Improved Correlation Properties. Монографія: In.: ISCI'2019: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan D. Gorbenko and Alexandr A. Kuznetsov, ASC Academic Publishing, USA, 2019, pp. 281-299. – ISBN: 978-0-9989826-8-7 (Hardback), ISBN: 978-0-9989826-9-4 (Ebook).

53. Смірнов О.А., Дрєєва Г.М. Метод генерування фрактального трафіку за допомогою моделі генератора на графі. Монографія: Інформаційна безпека та інформаційні технології : монографія / за заг. ред. В. С. Пономаренка. – Х. : Вид. Рожко С.Г. 2019. С. 123-139

54. Дрєєва Г.М., Смірнов О.А., Дрєєв О.М. Метод генерування фрактальноподібної числової послідовності на основі скінченного автомату для моделювання трафіку у мережі. Центральноукраїнський науковий вісник. Технічні науки. № 1(32). с. 173-183, 2019.

55. Смірнов О.А., Кавун С.В., Коваленко О.В., Дрєєв О.М. Мережні інформаційні технології. Навчальний посібник – Кіровоград: РВЛ КНТУ, 2016. – 159 с.

56. Смірнов О.А., Смірнов С.А. Дідик А.К., Дрєєв О.М. Моделі системи нейромережевих експертів безпечної маршрутизації у хмарних антивірусних

системах. Збірник наукових праць "Системи обробки інформації". – Випуск 3 (140). – Х.: ХУПС – 2016. – С. 36-39.

57. Смірнов О.А., Кавун С.В., Коваленко О.В., Доренський О.П., Дрєєв О.М., Вялкова В.І. Комп'ютерні мережі. Навчальний посібник – Кіровоград: РВЛ КНТУ, 2016. – 233 с.

58. Смірнов О.А., Дрєєв О.М. Порівняння бітових щільностей при використанні різних методів кодування інформації. Збірник наукових праць "Системи обробки інформації". – Випуск 2 (118). т.2. – Х.: ХУПС – 2014. – С. 64-67

59. Смірнов О.А., Дрєєв О.М. Порівняння бітових щільностей при використанні різних методів кодування інформації. Збірник тез VI міжнародної науково-практичної конференції "Проблеми та перспективи розвитку ІТ-індустрії". м. Харків. 17-18 квітня 2014р. – Харків: ХНЄУ. – 2014. – С. 240.

60. Смірнов О.А., Коваленко О.В., Кожанова А.С., Левошко О.Л., Константинова Л.В. Основи системного програмування. Навчальний посібник. – Кіровоград: КНТУ 2013. – 257с.