

ВІЙСЬКОВІ НАУКИ

DOI: <https://doi.org/10.32839/2304-5809/2019-9-73-54>

УДК 621.397

Бондаренко Т.В., Міщенко А.О., Бондаренко О.Є.,
Новак А.О., Побережець Т.В.

Військовий інститут телекомунікацій та інформатизації

АНАЛІЗ МОЖЛИВОСТЕЙ ЗАСТОСУВАННЯ ТЕХНОЛОГІЇ P2P-MCU В ТЕЛЕКОМУНІКАЦІЙНИХ МЕРЕЖАХ СПЕЦІАЛЬНОГО ПРИЗНАЧЕННЯ ПРИ РЕАЛІЗАЦІЇ БАГАТОТОЧЕЧНОЇ ВІДЕОКОНФЕРЕНЦІЇ

Анотація. WebRTC дозволяє веб-браузерам взаємодіяти в реальному часі через API – інтерфейси JavaScript. Але коли кількість учасників збільшується, вимоги до пропускної здатності та процесорів стають серйозною проблемою в комірчастій мережі на основі push. У цій статті ми пропонуємо підхід P2P-MCU для багатоточечної відеоконференції, який ілюструє різні моделі зв'язку з використанням технології P2P-MCU. У нашому підході модуль MCU інтегрований в браузер для поєднання і перекодування відео та аудіо потоків в режимі реального часу. І коли браузер діє як вузол MCU, який залишає сеанс конференції без повідомлення, інший браузер-кандидат може негайно взяти на себе управління, і поточна конференція WebRTC може бути легко відновлена за допомогою нашого алгоритму вибору MCU. Наше рішення P2P-MCU знижує використання ЦП на 64% і споживання смуги пропускання на 35% для кожного учасника в порівнянні з рішенням з комірчастою мережею в досліджах на восьмисторонніх конференціях WebRTC. Хоча модуль P2P-MCU може вводити деяку затримку (<500 мс), затримка стабільна і її сприйманням практично нехтуються.

Ключові слова: технологія P2P-MCU, багатоточечна відеоконференція, WebRTC, JavaScript, NAT.

Bondarenko Tetiana, Mishchenko Anatoly, Bondarenko Oleg,
Novak Alina, Poberezhets Tetiana

Military Institute of Telecommunications and Informatization

ANALYSIS OF POSSIBILITIES OF APPLICATION OF TECHNOLOGY P2P-MCU OF IN TCNS OF THE SPECIAL SETTING DURING REALIZATION OF VIDEOCONFERENCE OF SUSPENSION POINTS

Summary. WebRTC (short for Web Real Time Communication) stands for Real-Time Web Communication. It is an open standard for real-time multimedia communications that works directly in a web browser. An open standard platform eliminates the need to download additional programs, add-ons, and extensions. At the API level, WebRTC is standardized by the W3C, and at the protocol level, by the IETF. WebRTC [1] (Web Real-Time Communications) is of great interest because the API is inherently supported by many newer versions of popular browsers, ie Google Chrome and Mozilla Firefox. However, because WebRTC was originally designed to communicate between browsers, even for a small group, a multi-point conferencing model can be complicated or expensive. WebRTC is an evolving communication standard with a deep implementation in the browser system Firefox and Chrome. So far, this standard is supported only by three browsers: Opera, Mozilla Firefox and Google Chrome. WebRTC was created with the aim of quickly transmitting and protecting confidential information when using local network streams (organizing network meetings, remote conferences, etc.). WebRTC allows to the web-browsers to co-operate real-time through API are interfaces of JavaScript. But when the amount of participants increases, requirements to the carrying capacity and processors become a serious problem in a cellular network on the basis of push. In the real article we offer approach P2P – MCU for a multidrop videoconference, that illustrates the different models of connection with the use of technology of P2P – MCU. In our approach the module of MCU is integrated in a browser for combination and перекодування of video and audio streams real-time. And when a browser operates as a knot of MCU, that abandons the session of conference without a report, other browser-candidate can immediately undertake management, and current conference of WebRTC it can be easily renewed by means of our algorithm of choice of MCU. Our decision of P2P – MCU reduces the use of ЦП on 64% and consumption of stripe of key-in on 35% for every participant as compared to a decision with a cellular network in experiments on the eightextraneous conferences of WebRTC. Although module of P2P – MCU can enter some delay (<500 мс), a delay is stable and by her perception practically.

Keywords: technology of P2P-MCU, videoconference of suspension points, WebRTC, JavaScript, NAT.

Постановка проблеми. В наш час існує необхідність використовувати багатоточечну відеоконференцію в телекомунікаційних мережах спеціального призначення в реальному часі. Завдяки широко поширеним мережам фіксованого та мобільного широкосмугового зв'язку існує тенденція до проведення в будь-який час і в будь-якому місці багатоточечної відеокон-

ференцій в реальному часі. Щоб відповідати новим вимогам, WebRTC [1] (Web Real-Time Communications) викликаний великий інтерес, оскільки API за своєю природою підтримується багатьма новими версіями популярних браузерів, тобто Google Chrome і Mozilla Firefox. Однак, оскільки WebRTC спочатку був призначений для зв'язку між браузерами, навіть для невели-

кої групи, модель багатоточечної конференції може бути складною або дорогою. Зокрема, для підтримки N учасників конференції з чистою мобільною мережею буде $N * (N-1) / 2$ каналів. Вимоги до пропускної здатності / можливостей пристрою будуть збільшуватися в квадраті в порівнянні з кількістю учасників конференції. Відповідно, сервер MCU [2] (Multipoint Control Unit) введений для зменшення споживання смуги пропускання шляхом змішування мультимедіа, отриманого від користувачів в конференції, в один потік для кожного учасника. Однак MCU-сервер, зазвичай заснований на фіксованому та попередньо сконфігурованому обладнанні, часто досить дорогий і споживає значну частину смуги пропускання.

Аналіз останніх досліджень і публікацій.

Аналіз застосування та розвитку систем багатоточечного відеоконференцзв'язку здійснювався багатьма авторами. В [1] оцінюється час передачі відео пакетів та розраховується об'єм необхідної відеоінформації. В [14] представлені шляхи підвищення якості відеоконференцзв'язку в телекомунікаційних системах спеціального призначення. Недоліком проаналізованих матеріалів публікації за визначеною тематикою є відсутність врахування специфіки забезпечення багатоточечних відеоконференцій з використанням технології P2P – MCU. Розгорнута та функціонуюча на теперішній час у мережах спеціального зв'язку система ВКЗ обмежена застосуванням багатоточечних відео конференцій з використанням технології P2P – MCU.

Виділення не вирішених раніше частин загальної проблеми. Дослідженнями відомих авторів підтверджується актуальність досліджуваної проблеми. У цьому контексті актуальним залишається дослідження можливостей застосування технології P2P-MCU в телекомунікаційних мережах спеціального призначення за допомогою багатоточечної відеоконференції.

Мета статті. Головною метою цієї роботи є дослідження можливостей застосування технології P2P-MCU в телекомунікаційних мережах спеціального призначення за допомогою багатоточечної відеоконференції.

Виклад основного матеріалу. Внесок цієї статті: по-перше, ми розробляємо архітектуру P2P-MCU, що працює з поточними протоколами WebRTC; по-друге, ми пропонуємо стратегію визначення хоста MCU для динамічного і оптимального розміщення хоста MCU в веб-браузерах; нарешті, ми реалізуємо ефективний метод пробивання UDP [3] для мобільних користувачів за брандмауером / NAT для участі в відео конференції. Проводяться експерименти з комбінацією мобільних користувачів і користувачів ПК в різних конфігураціях. Результати показують, що наш підхід дійсний і ефективний для проведення багатосторонніх конференцій через WebRTC. Розглянемо принципи взаємодії користувачів в відеоконференції. Браузери WebRTC, які хочуть приєднатися до відеоконференції, можуть бути підключені різними способами. Для простоти ми використовуємо централізований веб-сервер для обробки сигналів між браузерами, що підтримують WebRTC, як показано на рис. 1. З'єднання між сервером сигналізації і браузером засновані на WebSocket [4]. Однак XHR / JSONP опитування може замінити з'єднання WebSocket в обмеженому середовищі на середовище 3G.

У відеоконференції з повною сіткою однорангові з'єднання встановлюються між кожною парою учасників в відеоконференцзалі. У запропонованому нами рішенні тільки один учасник повинен встановити з'єднання з хостом MCU, який підтримує сеанс MCU. Отже, для інших браузерів-учасників немає необхідності запускати будь-які MCU. Максимальна кількість учасників залежить від обчислювальної потужності хоста MCU. Розглянемо функціональність і характеристики сервера сигналізації, сеанс відеоконференцзв'язку використовуваного в розрахованому відеочаті на багато користувачів. Сервер сигналізації забезпечує адміністрування окремого відеоконференцзалу. Відеоконференцзал має список учасників. Учасник може створити нове середовище, приєднатися до неї і покинути її. Кожен раз, коли змінюється кількість учасників, то всі учасники в середовищі будуть повідомлені про це повідомленням. Оскільки обидва браузера можуть перебувати за NAT,



Рис. 1. З'єднання між сервером сигналізації і браузером засновані на WebSocket

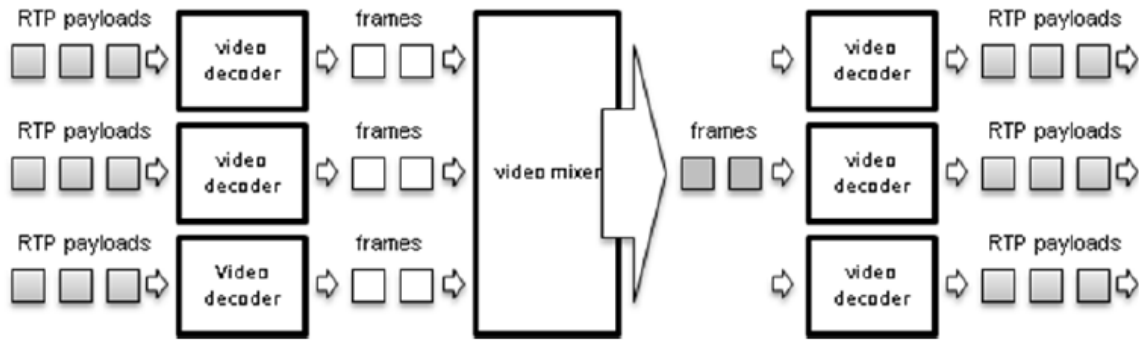


Рис. 2. Принцип об'єднання відеокадрів

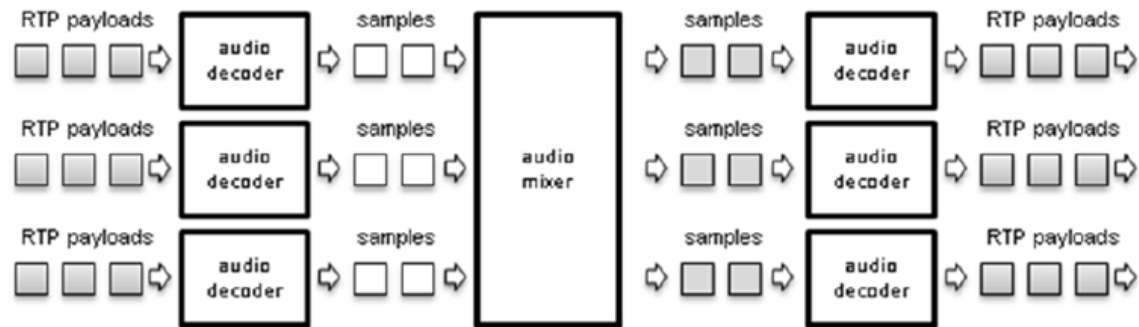


Рис. 3. Принцип об'єднання аудіокадрів

вони можуть не мати можливості спілкуватися безпосередньо. Сервер сигналізації працює шляхом маршрутизації повідомлень між учасниками в відеоконференцзалі. Прикладами таких повідомлень є можливості браузера, типу NAT, який він має, переваги частоти кадрів відео, повідомлення SDP [5] (протокол опису сеансу), кандидати WebRTCICE [6] і т.д. Кожне повідомлення містить у собі унікальну особистість учасника. Учасники можуть відправляти особисті повідомлення один одному або групі повідомлення всім учасникам в одному середовищі.

У нашому запропонованому підході ми представляємо P2P-MCU. P2P-MCU – це реалізація розподіленого MCU з використанням тимчасової (P2P) архітектури. P2P виключає централізовані медіа-сервери і складні інвестиції в інфраструктурі. На відміну від традиційного сервера відеоконференцій, MCU-хост працює в настільному браузері, який зазвичай знаходиться за NAT. Щоб подолати обмеження підключення NAT, модуль P2P-MCU використовує утиліти обходу сеансу для NAT (STUN) [7], щоб визначити правильну публічну адресу, яку NAT виділяє для трафіку UDP між локальним і зовнішнім хостами. MCU забезпечує кодування / декодування окремих потоків RTP [8] і змішування потоків RTP. MCU призначає аудіо / відео кодер і декодер, які здатні декодувати потоки RTP від учасника. У поєднанні відео, всі учасники будуть бачити одне й те саме комбіноване відео. Процес починається з декодування відео, яке декодує корисні дані RTP в відеокадри. Потім відео комбіноване всі відеокадри від різних учасників в один потік відеокадрів, як показано на рис. 2. Нарешті, окремий відекодер кодує комбіновані відеокадри назад в корисне навантаження RTP для кожного учасника.

Комбінування всіх аудіозаписів учасників в один потік може привести до відлуння голосу у відповідь. Щоб уникнути цього, у кожного учасника буде свій власний комбінований аудіопотік, який не містить його власного аудіо. Спочатку аудіодекодер декодує корисне навантаження RTP в аудіозразки. Потім аудіокомбіноване змішує аудіозразки в окремі потоки аудіозразків, як показано на рис. 3. Нарешті, окремий аудіокодер кодує відповідні аудіозразки назад в корисні дані RTP.

Інтеграція з браузером.

У цьому пункті ми коротко описуємо інтеграцію MCU в архітектуру WebRTC, щоб забезпечити швидку розробку і розгортання додатків для відеоконференцій через JavaScript API. У запропонованому нами рішенні ми покажемо можливість використання клієнтської бібліотеки P2P-MCU і STUN для JavaScript для зручного управління веб-додатками (рис. 4). Це контролер сеансу MCU і детектор NAT.

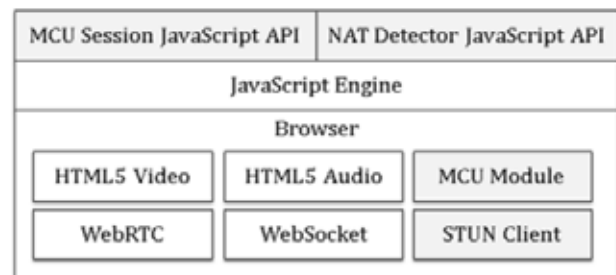


Рис. 4. Архітектура модифікованого браузера

Контролер сеансу MCU представлений у вигляді бібліотеки JavaScript API для полегшення розширення і модифікації логіки додатка. Він керує модулем P2P-MCU і включає в себе функ-

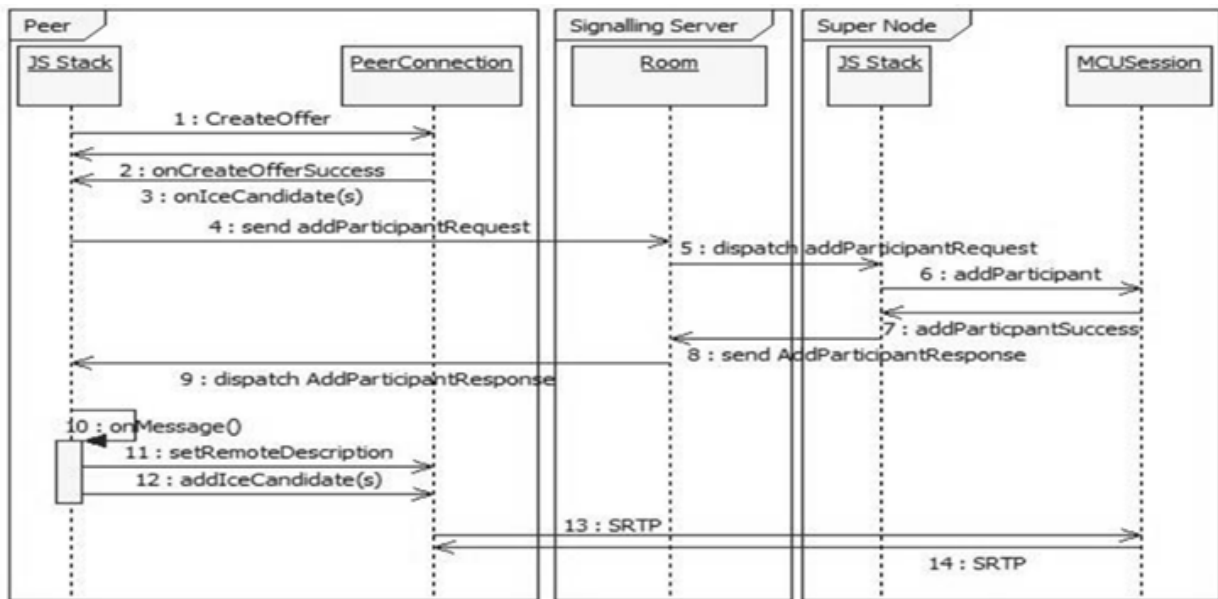


Рис. 5. Приклад встановлення зв'язку між MCU та учасником за допомогою сервера сигналізації

ціональність відеоконференцзалу MCU, включаючи створення відеоконференцзалу, додавання учасника в середовище і видалення групи учасників розмови з середовища.

Він працює шляхом опрацювання запитів SDP і відповідей на них, що генеруються і передаються між браузерами WebRTC через сервер сигналізації, як показано на рис. 5.

Детектор NAT інкапсулює функціональність клієнта STUN і представлення у вигляді бібліотеки JavaScript API. Клієнт STUN перевіряє наявність NAT і визначає тип NAT, за яким розміщується браузер.

Об'єкт WebRTC Peer Connection використовує механізми ICE для проходження багаторівневих NAT-пристроїв між веб-браузерами. Якщо пряме з'єднання неможливе, так як браузери знаходяться за симетричним NAT, трафік прямує через сервер ретрансляції TURN [9] в якості запасного варіанту, що може збільшити затримку. Визначивши тип NAT, ми можемо гарантувати, що сеанс MCU не буде розміщений за симетричним NAT, і виключаємо участь ретранслятора TURN.

Зв'язок.

Відео чат один на один.

В нашій запропонованій моделі ми припускаємо, що в конференцзалі є хоча б один браузер, інтегрований з MCU і не підтримує симетричний NAT. Ми розробляємо логіку, що відповідає за вибір найбільш відповідного учасника для роз-

міщення сеансу MCU. Хост MCU запрошує всіх учасників в середовищі встановити однорангові з'єднання з сеансом MCU. Після встановлення тимчасових з'єднань відео і аудіопотоків і до обох тимчасових вузлів направляються в сеанс MCU. Кілька відео та аудіо змішуються в відеопотік та окремі аудіопотоки і направляються всім учасникам (рис. 6).

Трьохсторонній відео чат.

Коли учасник в будь-який час приєднується до середовища, браузер, в якому розміщується сеанс MCU, пропонує новому учаснику встановити однорангове з'єднання (рисунок 7 зліва). Коли хост MCU в будь-який час залишає середовище, наступний відповідний учасник буде обраний для проведення нового сеансу MCU і повторного запрошення всіх учасників в середовище (рис. 7 праворуч). Якщо немає відповідного учасника для розміщення сеансу, відеочат знаходиться в режимі очікування. Кілька відео та аудіо сеансів завжди змішуються в сеансі MCU.

Восьмисторонній відео чат.

Насправді, різні пристрої будуть мати різні можливості. Мобільні пристрої мають меншу обчислювальну потужність, процесорну здатність, обсяг пам'яті і розмір екрану, ніж настільні комп'ютери. Відео стандартної чіткості, що підходить для настільних комп'ютерів, занадто вимогливі для мобільних пристроїв. Для підтримки гетерогенних пристроїв, різні пристрої повинні

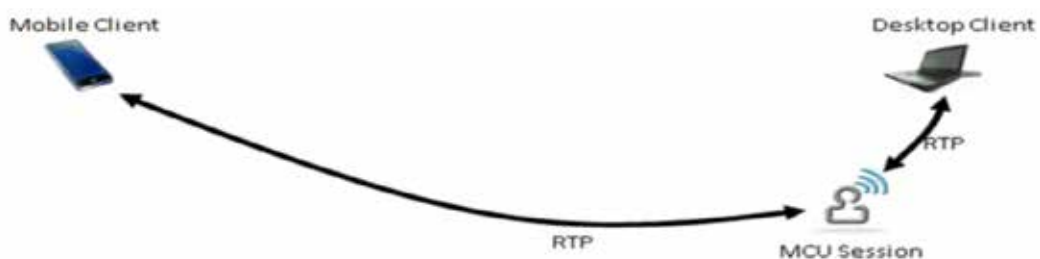


Рис. 6. Сценарій відеоконференцзв'язку за участю одного стаціонарного учасника з мобільним клієнтом

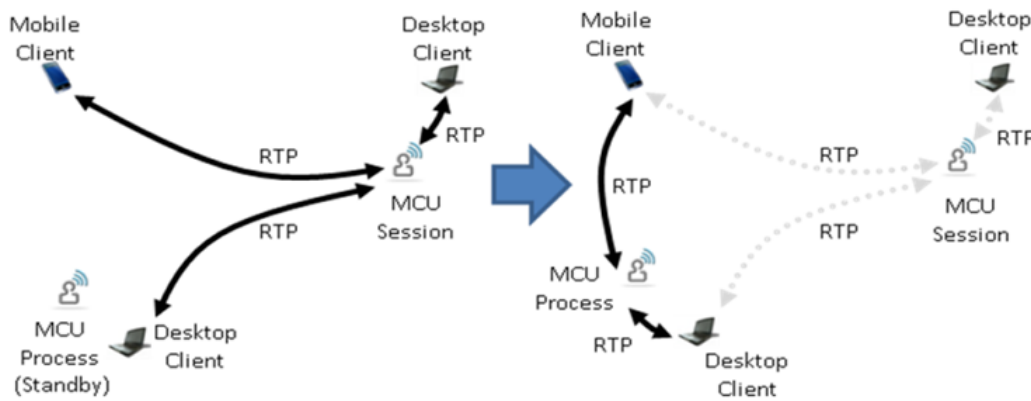


Рис. 7. Сценарій за участю двох стаціонарних учасників та мобільних клієнтів

отримувати відео і аудіо потоки в їх найбільш зручному профілі.

В WebRTC основним методом підтримки декількох учасників є використання декількох об'єктів Peer Connection, по одному для кожного учасника. Але при цьому методи складно змішати потоки в одне і закодувати їх для різних медіа-профілів. У нашому підході хост MCU володіє достатньою обчислювальною потужністю і пропускною спроможністю мережі, щоб змішувати і доставляти медіапотоки для кожного учасника безпосередньо в більш зручному профілі.

MCU Host визначення.

Після того, як MCU інтегрований в браузер, як описано вище, важливе питання полягає в тому, як вибрати відповідний хост MCU в системі для ефективної координації відеоконференції. Теоретично, будь-який браузер в системі може бути обраний для виконання MCU, якщо він задовольняє вимогам хоста MCU. Але на практиці, щоб зробити вибір ефективно, ми припускаємо, що MCU в цій статті належить одному браузеру на відеоконференції. Крім того, для керування сигналізацією та інформацією під час процедури визначення хоста MCU необхідний блок керування мережею. У нашій практиці сервер сигналізації є одним з найкращих кандидатів на пристрій управління.

Відповідно, є три кроки для визначення хоста MCU:

Крок 1: сервер сигналізації збирає інформацію для визначення MCU;

Крок 2: сервер сигналізації визначає, який браузер в конференції повинен бути хостом MCU, на основі інформації, зібраної на кроці 1.

Крок 3: сервер сигналізації повідомляє всі браузери про результат вибору.

Детально, для кроків 1 і 2, інформація включає в себе інформацію про мережі та інформацію про користувача. Інформація про мережу може

включати в себе інформацію про завантаження мережі і інформацію про топології, якщо необхідно, в той час як інформація про користувача може включати тип терміналу, незалежно від того, чи знаходиться користувач за брандмауером чи ні, і можливості терміналу, якщо необхідно. Крім того, на кроці 2 сервер сигналізації кількісно оцінює метрики, зазначені в інформації, оцінює метрики з попередньо визначеними вагами і вибирає клієнта з найкращою оцінкою як MCU. Нарешті, сервер сигналізації повинен повідомити всіх клієнтів, які з них є хостом MCU, і конференція може початися.

Висновки і пропозиції. Таким чином тенденція розвитку технології P2P-MCU відповідає підтримки багатоточечної відеоконференції в телекомунікаційних мережах спеціального призначення. Ми описали підхід P2P-MCU для підтримки багатоточечної відеоконференції навіть зі звичайним мобільним пристроєм Android. Наше рішення P2P-MCU можливо використовувати для мобільних користувачів, оскільки вони можуть просто встановити офіційний Chrome для Android на свої смартфони. Хоча новий доданий модуль P2P-MCU може вводити деяку затримку (<500 мс), затримка стабільна і сприймається учасниками майже непомітно. Наша стратегія визначення хоста MCU гарантує, що відеоконференція буде встановлена і відновлена без проблем. Відповідно, продуктивність P2P-MCU досить стабільна, і показник успішності встановлення з'єднання в мережах 3G становить майже 90%, що набагато вище, ніж в звичайному WebRTC. Результати восьмистороннього експерименту по відеоконференціям показали, що наше рішення може скоротити використання ЦП на 64% і споживання смуги пропускання на 35% для кожного учасника в порівнянні з чистою стільниковою мережею WebRTC.

Список літератури:

1. A. Bergkvist, D.C. Burnett, C. Jennings, and A. Narayanan. WebRTC 1.0: Real-time Communication between Browsers.
2. M.H.W. LeMair, D.D. Kandlur, and Z.Y. Shae. "On multipoint control units for videoconferencing," in Proc. 19th Conference on Local Computer Networks, 1994, pp. 356–364.
3. P. Srisuresh, B. Ford, and D. Kegel. State of Peer-to-Peer (P2P) Communication Across Network Address Translators (NATs).[Online]. Available: <http://www.rfc-editor.org/rfc/rfc5128.txt>
4. I. Fette and A. Melnikov. The Web Socket Protocol [Online]. Available: <http://www.rfc-editor.org/rfc/rfc6455.txt>
5. M. Handley, V. Jacobson, and C. Perkins. SDP: Session Description Protocol. [Online]. Available: <http://www.rfc-editor.org/rfc/rfc4566.txt>

6. J. Rosenberg. Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols.
7. J. Rosenberg, J. Weinberger, C. Huitema, and R. Mahy. STUN Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs). [Online]. Available: <http://www.rfc-editor.org/rfc/rfc3489.txt>
8. H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A Transport Protocol for Real-Time Applications. [Online]. Available: <http://www.rfc-editor.org/rfc/rfc3550.txt>
9. R. Mahy, P. Matthews, and J. Rosenberg. Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN). [Online]. Available: <http://www.rfc-editor.org/rfc/rfc5766.txt>
10. MCU Open Source Version.
11. Chromium. [Online]. Available: <http://www.chromium.org>
12. Андреев А., Аржанов В., Семенов К. Применение видеоконференцсвязи в Вооружённых силах иностранных государств. *Зарубежное военное обозрение*. 2008. № 7. С. 19–25; № 8. С. 16–22.
13. Ватолін В., Рагушняк А., Смирнов М., Юкин В. Методи сжатия данных. Устройство архиваторов, сжатие изображений и видео. Москва : ДИАЛОГ – МИФИ, 2002. 384 с.
14. Пантась С.О., Цатурян О.Г., Беляков Р.О., Радченко М.М. Шляхи підвищення якості відеоконференцзв'язку в телекомунікаційних системах спеціального призначення. *Збірник наукових праць ВІТІ*. 2017. № 3. 113 с.

References:

1. A. Bergkvist, D.C. Burnett, C. Jennings, and A. Narayanan. WebRTC 1.0: Real-time Communication between Browsers.
2. M.H.W. LeMair, D.D. Kandlur, and Z.Y. Shae, “On multipoint control units for videoconferencing,” in Proc. 19th Conference on Local Computer Networks, 1994, pp. 356–364.
3. P. Srisuresh, B. Ford, and D. Kegel. State of Peer-to-Peer (P2P) Communication Across Network Address Translators (NATs). [Online]. Available: <http://www.rfc-editor.org/rfc/rfc5128.txt>
4. I. Fette and A. Melnikov. The Web Socket Protocol. [Online]. Available: <http://www.rfc-editor.org/rfc/rfc6455.txt>
5. M. Handley, V. Jacobson, and C. Perkins. SDP: Session Description Protocol. [Online]. Available: <http://www.rfc-editor.org/rfc/rfc4566.txt>
6. J. Rosenberg. Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols.
7. J. Rosenberg, J. Weinberger, C. Huitema, and R. Mahy. STUN Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs). [Online]. Available: <http://www.rfc-editor.org/rfc/rfc3489.txt>
8. H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A Transport Protocol for Real-Time Applications. [Online]. Available: <http://www.rfc-editor.org/rfc/rfc3550.txt>
9. R. Mahy, P. Matthews, and J. Rosenberg. Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN). [Online]. Available: <http://www.rfc-editor.org/rfc/rfc5766.txt>
10. MCU Open Source Version.
11. Chromium. [Online]. Available: <http://www.chromium.org>
12. Andreev A., Arzhanov B., Semenov K. Application of videokonferencsvyazi is in Military powers of the foreign states. *Foreign military review*. 2008. № 7. P. 19–25; № 8. P. 16–22.
13. Vatin V., Ratushnyak A., Smirnov M., Yukin V. Methods of compression of data. Device of self-extractors, compression of images and video. Moskva : DIALOG – MIFI, 2002. 384 p.
14. Pantas S.O., Caturyan O.G., Belyakov R.O., Radchenko M.M. Shlyakhi upgrading videokonferenczv'yazku in the telecommunication systems of the special setting. *Collection of scientific labours of VITI*. 2017. № 3. 113 p.