

ПОРІВНЯЛЬНИЙ АНАЛІЗ МОЖЛИВОСТЕЙ ТОПОЛОГІЙ МЕРЕЖІ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЇ WEBRTC

Основна мета вибору оптимальної топології при використанні технології WebRTC – це збільшення користувачів, які зможуть перебувати в аудіо та відео конференціях. Саме тому розробникам необхідно зробити вибір, яку топологію краще використовувати в залежності від різних вимог. Проблема технології WebRTC в обмеженій масштабованості. Щоб по-справжньому масштабувати програми WebRTC, необхідно використовувати топологію, яка найкраще підходить за поставленими розробниками вимогами (Рис.1).

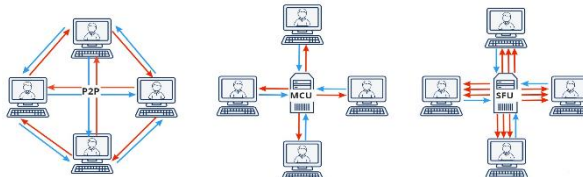


Рис. 1. Різновиди топологій мереж з P2P, MCU, SFU

Початкова реалізація WebRTC передбачалася лише для підтримки однорангового зв'язку peer-to-peer (P2P). Цей зв'язок є найпростішим у налаштуванні та найбільш економічно ефективним для використання в додатку. У P2P учасник безпосередньо з'єднаний з кожним іншим учасником через активне з'єднання/канал даних. Одноранговий зв'язок працює таким чином, що учасник надсилає і одночасно завантажує відео та аудіо від кожного однорангового користувача.

Основною перевагою P2P є його простота реалізації та низька вартість. Друга перевага полягає в тому, що між учасниками гарантується наскрізна безпека, оскільки кінцеві точки можуть покладатися на безпеку транспортного рівня і не повинні шифрувати медіа, адже на мережевому шляху немає посередників.

Проблеми зі змішаною P2P топологією для WebRTC починаються при дзвінках з великою кількістю учасників. У сценарії багатостороннього дзвінка кожен учасник повинен надіслати свій медіа-контент усім іншим учасникам. Крім того, існують також значні обчислювальні витрати для кожного клієнтського пристрою, оскільки він повинен кодувати один і той же потік кілька разів. На практиці безпосередній взаємозв'язок працює добре, якщо кількість учасників дзвінка невелика. Multipoint Control Unit (MCU) протягом багатьох років був основною системою конференцій для великих груп. Це не дивно, враховуючи його здатність забезпечувати стабільне потокове аудіо/відео з низькою пропускну здатністю і переносити більшу частину обробки потоку з клієнта на централізований сервер. У зіркоподібній топології з MCU кожен клієнт підключений до централізованого сервера MCU, який декодує, масштабує та змішує всі вхідні потоки в один новий потік, а потім кодує та надсилає його всім учасникам.

Переваги використання MCU полягають у тому, що він особливо добре працює в середовищах з низькою пропускну здатністю і відмінно масштабується зі збільшенням кількості учасників. Недоліком MCU є велике навантаження процесора під час декодування та перекодування для складання єдиного потоку зі всіма учасниками, що несе за собою також збільшення вартості. Другим недоліком є затримка відображення відео, яке може бути ускладнене втратою пакетів на одному з посилянь, в результаті чого потрібно чекати, поки весь кадр перекодується. В зіркоподібній топології з Selective Forwarding Unit (SFU) кожен учасник надсилає свій медіа-потік на централізований сервер SFU і отримує потоки від усіх інших учасників через той самий центральний сервер. Топологія з SFU дозволяє учаснику дзвінка надсилати кілька медіа-потоків до SFU, де сервер може вирішити, який із медіа-потоків слід переслати іншим учасникам виклику. На відміну від топології з MCU, SFU не потребує декодування та перекодування отриманих потоків, а просто діє як маршрутизатор потоків між учасниками виклику. Кінцеві точки SFU повинні мати більшу обчислювальну потужність, ніж в MCU.

Топології з SFU дуже добре масштабуються, зберігаючи при цьому навантаження на сервер до мінімуму. Зменшення ролі сервера до «маршрутизатора» потоків також тримає витрати досить низькими.

В результаті порівняльного аналізу можна зробити такі висновки:

1. Змішана топологія з P2P і її проста реалізація підходить для менш ніж 3 одночасних учасників.
2. Зіркоподібна топологія з SFU підтримує від 5 до 10 учасників і є компромісним варіантом між P2P та MCU.
3. Зіркоподібна топологія з MCU підтримує більше 10 учасників, але значно дорожча і складніша в реалізації порівняно з P2P та SFU.