

Сконфигурировать медиа сервер RTP -> WebRTC (screen sharing), разработать скрипты и веб-страницы

Нужно разработать рабочий прототип системы для удаленного управления (screen sharing) через RTP поток.

Медиа сервер (Janus) нужно сконфигурировать так, чтобы он принимал RTP поток из тестового источника RTP и отправлял бы его на тестовую веб-страницу.

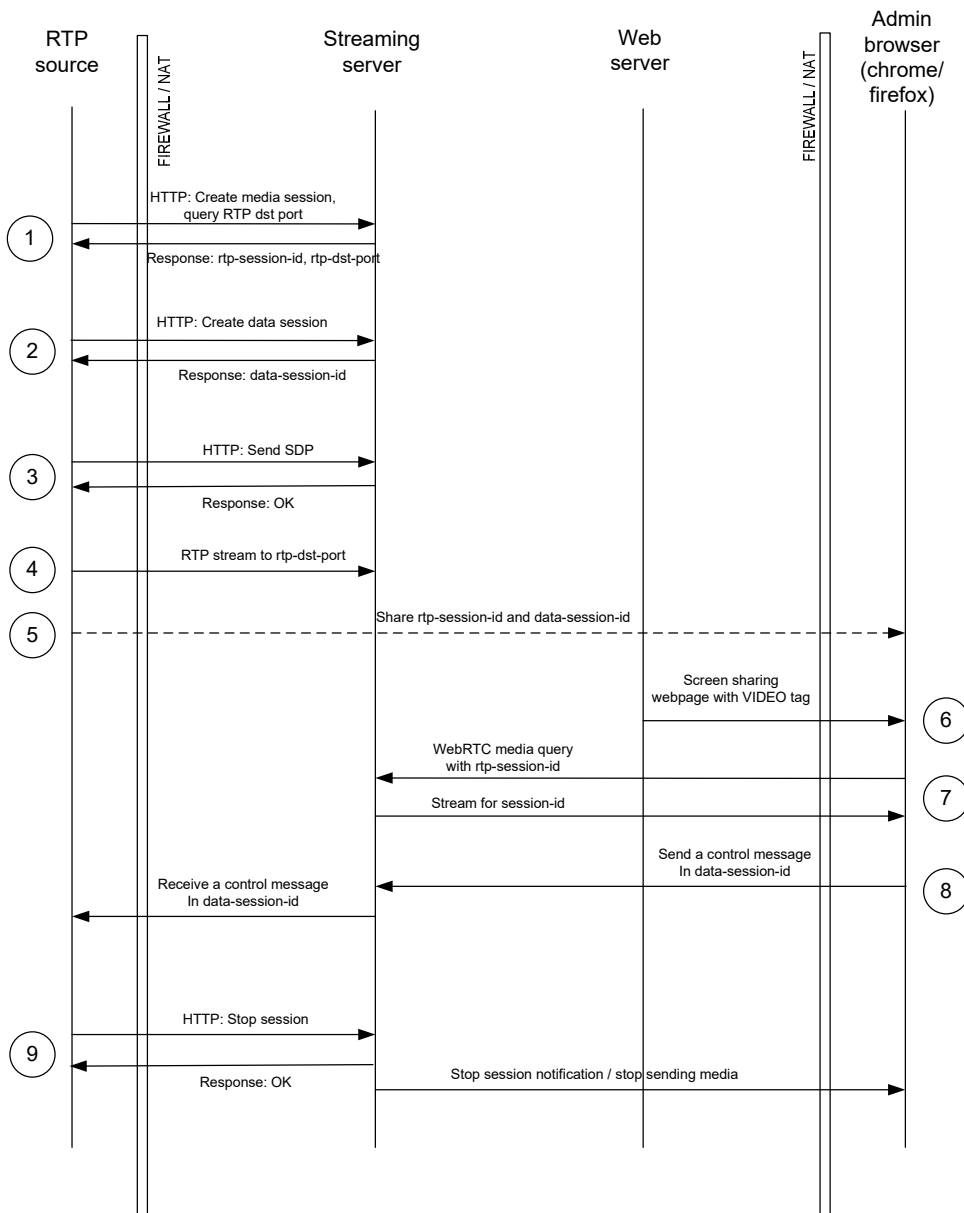
Тестовая страница должна отображать видео в браузере с помощью WebRTC. Также, она должна отправлять текстовые сообщения (управление) обратно в источник RTP, с минимальной задержкой (используя механизм Data Channels Janus).

И источник RTP, и веб-страница, будут за файрволами. Поэтому, скорее всего в системе также должен быть STUN/TURN/ICE сервер, для управления адресами и портами UDP.

Система должна работать на Linux. Компоненты системы должны быть упакованы в Docker контейнер (у меня уже есть Docker контейнер для Janus сервера, правда не сконфигурированного).

Детали диаграммы взаимодействия и описание – во вложении.

Диаграмма взаимодействия



1. Источник RTP создает сессию на медиа-сервере через Janus API. Медиа-сервер возвращает ID сессии, а также адрес и порт, куда следует отправлять RTP поток.
2. Источник RTP создает Data Channel на медиа-сервере. Медиа-сервер возвращает ID сессии для сессии данных, в которой будут отправляться управляющие сообщения.
3. Источник RTP отправляет SDP на медиа сервер. На этом шаге может понадобиться STUN сервер, так как источник RTP находится за NAT-ом.
4. Источник RTP отправляет RTP поток на указанный медиа-сервером адрес и порт.
5. Идентификаторы rtp-session-id и data-session-id (вручную) сообщаются пользователю, который будет смотреть медиа-поток в браузере.

6. Пользователь открывает веб-страницу, отображающую медиа-поток, в браузере, и вводит ID сессии. Здесь тоже может понадобиться STUN сервер для отправки управляющих сообщений.
7. Страница соединяется с медиа-сервером, принимает поток и отображает видео в окошке.
8. Пользователь отправляет тестовое сообщение путем ввода в поле на странице. Сообщение доставляется в источник RTP.
9. Источник RTP уведомляет медиа-сервер об окончании трансляции. Веб-страница прекращает показывать видео.

Стек технологий

В качестве тестового источника RTP, можно использовать `ffmpeg`.

Для отправки и получения управляющих сообщений, SDP и т.д., можно либо разработать простое приложение командной строки, либо использовать `bash` скрипты и утилиту `netcat`.

В качестве медиа сервера нужно использовать `Janus`.

STUN сервер можно выбрать любой на ваш вкус (какой проще в установке и настройке).

Также, можете использовать любой веб-сервер, например `Apache`.

Веб-страница должна быть реализована на HTML+JavaScript, либо без использования JavaScript-фреймворка (`jQuery` использовать можно), либо с использованием `AngularJS`.

Что вам нужно сделать

1. Разработать конфигурацию медиа сервера `Janus`, которая выполняет задачу.
2. При необходимости включить ICE/STUN/TURN сервер в решение.
3. Разработать тестовое приложение – источник RTP, который устанавливает соединение с медиа-сервером, отправляет SDP, и получает тестовые сообщения от веб-страницы. RTP поток может отправляться с помощью `ffmpeg`.
4. Разработать веб-страницу, выполняющую свою задачу.
5. Протестировать и продемонстрировать работу на моем сервере (Ubuntu 18.04).
6. Разработать Docker контейнер, который позволит запускать решение на любом Linux. В качестве исходного образа рекомендую использовать Ubuntu 18.04.
7. Написать инструкцию по развертыванию.

Финальное решение должно работать «из коробки», и будет тестироваться путем развертывания по вашей инструкции на двух разных ОС (Ubuntu 16.04 и Ubuntu 18.04).