# WebRTC - описание технологии и требования к оборудованию

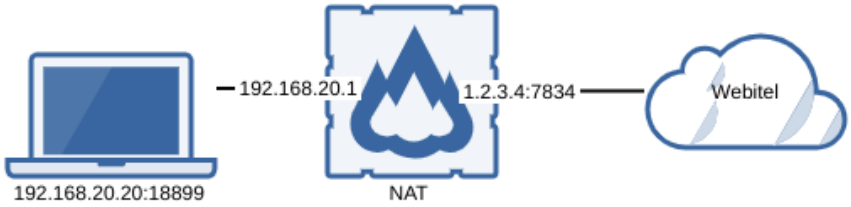
**WebRTC** - это технология, которая ориентирована на передачу видео и голоса между браузерами. Основные преимущества стандарта:

- Не требуется установка дополнительного ПО.
- Очень высокое качество связи, благодаря:
  - Использованию современных видео (VP8, H.264) и аудиокодеков (Opus).
  - Автоматическое подстраивание качества потока под условия соединения.
  - Встроенная система эхо- и шумоподавления.
  - Автоматическая регулировка уровня чувствительности микрофонов участников (АРУ).
- Высокий уровень безопасности: все соединения защищены и зашифрованы согласно протоколам TLS и SRTP.
- Есть встроенный механизм захвата контента, например, рабочего стола.
- Настоящая кросс-платформенность: одно и то же WebRTC приложение будет одинаково хорошо работать на любой операционной системе, при условии, что браузер поддерживает WebRTC.

Установить соединение – довольно трудная задача, так как компьютеры не всегда обладают публичными IP адресами, то есть адресами в интернете. WebRTC использует протокол ICE для решения проблемы работы в частной сети, который, правда, требует использования дополнительных серверов (STUN, TURN). При установке соединения в WebRTC не указывается адрес того узла, с которым нужно соединиться. Устанавливается сначала логическое соединение, а не физическое. Но это не покажется странным, если не забывать, что мы используем сторонний сигнальный механизм - ICE. Через некоторые callback'и WebRTC сообщает нам Ice candidate объекты. А почему кандидатов может быть много? Потому что расположение в сети может определяться не только своим внутренним IP адресом, но также и внешним адресом маршрутизатора, и не обязательно одного, а также адресами TURN серверов.

**STUN** сервер – это просто сервер в интернете, который возвращает обратный адрес, то есть адрес узла отправителя. Узел, находящийся за роутером, обращается к **STUN** серверу, чтобы пройти через **NAT**. Пакет, пришедший к **STUN** серверу, содержит адрес источника – адрес роутера, то есть внешний адрес нашего узла. Этот адрес **STUN** сервер и отправляет обратно. Таким образом, узел получает свой внешний **IP** адрес и порт, через который он доступен из сети. Далее, **WebRTC** с помощью этого адреса создает дополнительного кандидата (внешний адрес роутера и порт). Теперь в таблице **NAT** роутера есть запись, которая пропускает к нашему узлу пакеты, отправленные на роутер по нужному порту.

Рассмотрим этот процесс на примере:




Маршрутизатор обычно содержит таблицу NAT. Это специальных механизм, разработанный для того, чтобы узлы внутри приватной сети роутера смогли обращаться, например, к веб-сайтам. Веб-клиент шлет запрос на адрес сервера Webitel. Сначала данные попадают на роутер (NAT), а точнее на его внутренний интерфейс 192.168.20.1. После чего, роутер запоминает адрес источника (192.168.20.20) и порт запроса (18899), заносит его в специальную таблицу NAT, затем изменяет адрес источника на свой(192.168.20.20  1.2.3.4) и на новый свободный порт (7834). Далее, по своему внешнему интерфейсу роутер пересылает данные непосредственно на Webitel. Webitel обрабатывает данные, генерирует ответ и отправляет обратно. Отправляет роутеру 1.2.3.4 на порт 7834, так как именно он стоит в обратном адресе (роутер подменил адрес на свой). Роутер получает данные, смотрит в таблицу NAT и пересылает данные узлу 192.168.20.20 на порт 18899. Роутер выступает здесь как посредник.

Итак, в начале имеем пустую таблицу **NAT**.

| Internal IP | Internal PORT | External IP | External PORT |
|---|---|---|---|
| | | | |

После первого исходящего запроса, мы получаем такую таблицу **NAT**:

| Internal IP | Internal PORT | External IP | External PORT |
|---|---|---|---|
| 192.168.20.20 | 18899 | 1.2.3.4 | 7834 |

Если таблица NAT очищается очень часто (из-за большого количества абонентов в сети), тогда роутер не сможет обработать входящий запрос клиенту и все входящие звонки будут отбрасываться, пока клиент не сделает следующий исходящий запрос для создания новой записи в таблице **NAT**.

## Требования к оборудованию и сети

Требования к ПК пользователя зависят от используемого качества передачи голоса и видео. По умолчанию система пытается установить соединение в максимальном качестве связи. В таблице перечислены требования к процессорам (распространенные, но не все, модели процессоров), оперативной памяти, а так же, к интернет каналам.

**Важно**: Загруженность процессора желательна не более 50%.

| | Стандартное качество (SD) | Высокое качество (HQ) | Улучшенное качество (ED) | Высокой четкости качество (HD) | HD (при 60 fps) или FullHD |
|---|---|---|---|---|---|
| Операционная система | Microsoft Windows 10, macOS 10.12+, Linux | | | | |
| Процессор | Intel Celeron G3xxx (от 2,7 ГГц);<br><br>Intel Celeron 3xxx**U** (от 1,5 ГГц);<br><br>AMD Ryzen 3 2xxx**U** (от 2,5 ГГц) | Intel Celeron G3xxx (от 2,7 ГГц);<br><br>Intel Celeron 3xxx**U** (от 1,7 ГГц);<br><br>AMD Ryzen 3 2xxx**U** (от 2,5 ГГц) | AMD Athlon 64 X2 (от 2,4 ГГц);<br><br>Intel Celeron G3xxx (от 2,7 ГГц);<br><br>Intel Pentium 4xxx**Y** (от 1,5 ГГц);<br><br>AMD Ryzen 3 2xxx**U** (от 2,5 ГГц) | Intel Core i5-4xx**M** (от 2,4 ГГц);<br><br>Intel Core i5-2xxx (от 2,3 ГГц);<br><br>Intel Pentium 4xxx**U** (от 2,1 ГГц);<br><br>AMD Ryzen 3 2xxx**U** (от 2,5 ГГц) | Intel Core i7-2xxx (от 3,0 ГГц);<br><br>Intel Pentium G4xxx (от 2,9 ГГц);<br><br>Intel Core i5-7xxx**U** (от 2,6 ГГц);<br><br>AMD Ryzen 3 1xxx (от 3,1 ГГц) |
| Оперативная память | 1 ГБ | 2 ГБ | | 4 ГБ | |
| Сеть | 256 кбит/с в обоих направлениях | от 512 кбит/с в обоих направлениях | от 1 Мбит/с в обоих направлениях | от 2 Мбит/с в обоих направлениях | от 4 Мбит/с в обоих направлениях |

Для WebRTC критична низкая потеря и минимальные задержки пакетов. Технологии беспроводной связи, такие как 3G, 4G, EDGE, спутниковая связь архитектурно предполагают большие (по сравнению с проводным доступом) задержки и потери. Кроме того, качество связи сильно зависит от уверенности приема в данном конкретном месте и загруженности сети поставщика сервиса (провайдера). По своему опыту, мы не рекомендуем использование беспроводных технологий доступа в Интернет, т.к. они часто не могут предоставить необходимое качество связи.

Рекомендуем использовать браузер Google Chrome или Mozilla Firefox.

Технология WebRTC не поддерживает работу клиентов за HTTP Proxy, для корректной передачи голоса должно быть разрешено подключение клиентов за NAT на порты **TCP 443, UDP 1024-65535** сервера телефонии.

Поскольку весь голосовой и сигнальный трафик между клиентом и сервером шифруется с помощью **Datagram Transport Layer Security (DTLS)**, не допускается перехват трафика с подменой сертификатов либо использование антивирусного ПО с данным функционалом.

## Краткая сводка

Здесь приведены некоторые утверждения о сущностях **WebRTC**:

- Медиа поток
  - Видео и аудио данные упаковываются в медиа потоки
  - Медиа потоки синхронизируют медиа дорожки, из которых состоят
  - Различные медиа потоки не синхронизированы между собой
  - Медиа потоки могут быть локальными и удаленными, к локальному обычно подключена камера и микрофон, удаленные получают данные из сети в кодированном виде

    - Медиа дорожки бывают двух типов – для видео и для аудио
    - Медиа дорожки имеют возможность включения/выключения
    - Медиа дорожки состоят из медиа каналов
    - Медиа дорожки синхронизируют медиа каналы, из которых состоят
    - Медиа потоки и медиа дорожки имеют метки, по которым их можно различать
- Дескриптор сессии
    - Дескриптор сессии используется для логического соединения двух узлов сети
    - Дескриптор сессии хранит информацию о доступных способах кодирования видео и аудио данных
    - **WebRTC** использует внешний сигнальный механизм – задача пересылки дескрипторов сессии (**sdp**) ложится на приложение
    - Механизм логического соединения состоит из двух этапов – предложения (**offer**) и ответа (**answer**)
    - Генерация дескриптора сессии невозможна без использования локального медиа потока в случае предложения (**offer**) и невозможна без использования удаленного дескриптора сессии в случае ответа (**answer**)
- Кандидаты
    - Кандидат (**Ice candidate**) – это адрес узла в сети
    - Адрес узла может быть своим, а может быть адресом роутера или **TURN** сервера
    - Кандидатов всегда много
    - Кандидат состоит из **IP** адреса, порта и типа транспорта (**TCP** или **UDP**)
    - Кандидаты используются для установления физического соединения двух узлов в сети
    - Кандидатов также нужно пересылать через сигнальный механизм
- **STUN/TURN/ICE/NAT**
    - **NAT** – механизм обеспечения доступа к внешней сети
    - Сетевые роутеры поддерживают специальную таблицу **NAT**
    - Роутер подменяет адреса в пакетах – адрес источника на свой, в случае, если пакет идет во внешнюю сеть, и адрес приемника на адрес узла во внутренней сети, если пакет пришел из внешней сети
    - Для обеспечения многоканального доступа к внешней сети **NAT** использует порты
    - **ICE** – механизм обхода **NAT**
    - **STUN** и **TURN** сервера – сервера-помощники для обхода **NAT**
    - **STUN** сервер позволяет создавать необходимые записи в таблице **NAT**, а также возвращает внешний адрес узла