



WebRTC IP addresses and port ranges

Connecting a WebRTC session takes multiple network connections and messages taking place over different types of transport protocols.

WebRTC use: TCP connections and UDP.

WebRTC signaling connections

In order to send and receive the SDP blobs to make the connection.

WebRTC use: HTTP, WebSocket. They are implemented on top of TCP.

WebRTC media connections

Media in WebRTC gets connected via [SRTP](#). Most of the time, that would happen over UDP. When we create a peer connection, media tracks will be added with instructing what you want it to send or receive. WebRTC will then allocate and bind local IP addresses and ports to handle that traffic. As with outgoing TCP connections, the local ports are going to be arbitrary and ephemeral. In most cases, the addresses that they will use will be public IP addresses, but the ports will be arbitrary. That's because media servers prefer handling each incoming device separately, by receiving its traffic on a dedicated socket connected to a specific port.

STUN "connections"

Since we're all behind NATs with our private IP addresses, we need to know our public IP address so we can connect to others directly (peer-to-peer).

WebRTC will take the media local IP:port it created (in that section above), and use it to "connect" over UDP to a STUN server.

TURN connections

With TURN, the server is relaying our media towards the other user. UDP, TCP and TLS work similarly in TURN when it comes to address and port allocation. What is important to notice here is



how the TURN server opens up and allocates ports on its public IP address whenever someone tries to connect through it.

Understanding port ranges in WebRTC configurations

WebRTC makes use of a range of addresses, ports and transport protocols. In WebRTC the addresses and ports that get allocated by the browsers, media servers and TURN servers are dynamic. This means that in many cases we have to deal with port ranges.

Our recommendations regarding firewall configuration:

open all UDP ports for host live-webrtc.thespeech.app, stage-webrtc.thespeech.app, streaming.thespeech.app

open a few TCP ports (80, 433, 8089, 18098) for host <https://live-webrtc.thespeech.app>, <https://stage-webrtc.thespeech.app>, streaming.thespeech.app

Please enable following ports on firewall:

35.157.187.99 for live-webrtc.thespeech.app

3.68.38.156 for streaming.thespeech.app

3.125.29.169 for <https://stage-webrtc.thespeech.app>

- add rules for connections used the Twilio stun/turn servers:

```
[
  {
    "url": "stun:global.stun.twilio.com:3478?transport=udp",
    "urls": "stun:global.stun.twilio.com:3478?transport=udp"
  },
  {
    "url": "turn:global.turn.twilio.com:3478?transport=udp",
    "username": "1abd1ea429e88f8fe258acf634d8a14aac672997819bc9f97770e9b2001ac05e",
    "urls": "turn:global.turn.twilio.com:3478?transport=udp",
    "credential": "Mt4eB526V1LN8j170XgmEjnX5aA4mhM1iy01LYWrmio="
  },
  {
    "url": "turn:global.turn.twilio.com:3478?transport=tcp",
    "username": "1abd1ea429e88f8fe258acf634d8a14aac672997819bc9f97770e9b2001ac05e",
```



```
"urls": "turn:global.turn.twilio.com:3478?transport=tcp",  
"credential": "Mt4eB526V1LN8j170XgmEjnX5aA4mhM1iy01LYWrmio="  
},  
{  
  "url": "turn:global.turn.twilio.com:443?transport=tcp",  
  "username": "1abd1ea429e88f8fe258acf634d8a14aac672997819bc9f97770e9b2001ac05e",  
  "urls": "turn:global.turn.twilio.com:443?transport=tcp",  
  "credential": "Mt4eB526V1LN8j170XgmEjnX5aA4mhM1iy01LYWrmio="  
}  
]
```