
# ALCATEL-LUCENT RAINBOW™

## Network Requirements

GETTING STARTED GUIDE Ed 12

MARCH 2019

*Author: Operations - Cloud Services*

Alcatel·Lucent Enterprise

**Disclaimer**

This documentation is provided for reference purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this documentation, this documentation is provided “as is” without any warranty whatsoever and to the maximum extent permitted.

In the interest of continued product development, ALE International reserves the right to make improvements to this document and the products it describes at any time without notice or obligation.

**Copyright**

# Contents

# Glossary

**ALE:** Alcatel-Lucent Enterprise

**PBX:** Private Branch Exchange

**HTTP:** Hyper Text Transfer Protocol

**HTTPS:** Hyper Text Transfer Protocol Secured

**ICE:** Interactive Connectivity Establishment – RFC 5245

**STUN:** Simple Traversal of UDP through NAT – RFC 5389

**TURN:** Traversal Using Relays around NAT – RFC 5766

**DTLS-SRTP:** Datagram Transport Layer Security – Secured Real Time Protocol

# 1 Introduction

This guide provides technical requirements to connect Rainbow clients and Agents to Rainbow Cloud services.

# 2 Overview

Alcatel-Lucent Enterprise (ALE) is introducing Alcatel-Lucent Rainbow, an overlay cloud service operated by ALE. Rainbow offers contact management, presence, persistent messaging, audio/video, screen and file sharing, with PSTN termination and API openness to integrate with existing customer PBXs, machines and apps.

Rainbow's clients and agents connect to Rainbow cloud services using Web protocols.

More details about Web protocols used are provided in this document.

# 3 History

| Modifications | Date | Edition |
|---|---|---|
| Updated section 5.4 with complete list of public IP addresses<br>Section 5.3.4: WebRTC Gw now supports crossing web proxy for media flows. | 15/03/2019 | Ed 12 |
| Improve and complete IP flows information presentation; some doc reorg; precisions on bandwidth for mobile devices; new TURN server | 10/01/2019 | Ed 11 |
| Extended WebRTC TURN endpoints configuration | 11/14/2018 | Ed 10 |
| Video Bandwidth requirements update, WebRTC GW requirements and SDK specificities | 05/17/2018 | Ed 09 |
| TURN endpoints update, bandwidth requirements update | 04/07/2017 | Ed 08 |
| HTTP vs. HTTPS cleanup | 04/04/2017 | Ed 05 |
| Minor change (legacy PBX Agent removed) | 31/03/2017 | Ed 04 |
| Information on bandwidth added (chapter 5.5) | 08/03/2017 | Ed 03 |
| Chapter 6.1 modified | 05/01/2017 | Ed 02 |
| Creation of document | 27/10/2016 | Ed 01 |

# 4 Related documents

None

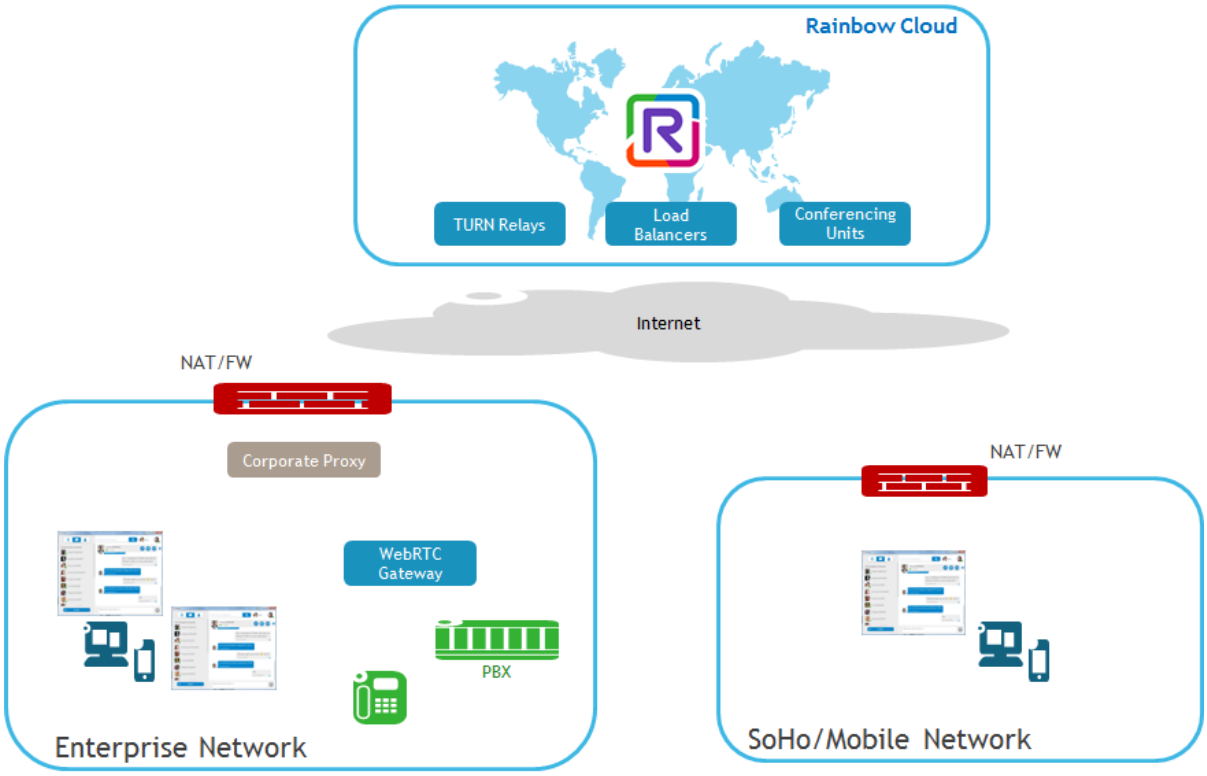
# 5 Requirements

## 5.1 Global Overview

The Rainbow solution provides multiple client-side applications to connect to the service:

- A Web-based Qt-contained Desktop application for Windows and OSX.
- A Web application for Chrome/Firefox browsers.
- An iOS native application.
- An Android native application.
- An Agent to connect the PBX (can be integrated with the PBX)
- A WebRTC Gateway to establish multimedia calls between PBX and Rainbow
- Various SDKs allowing developers building client and server applications leveraging the Rainbow CPaaS capabilities (see https://hub.openrainbow.com)

The following picture provides the global overview of Rainbow from network perspective:



## 5.2 Used Protocols and High-Level Principles

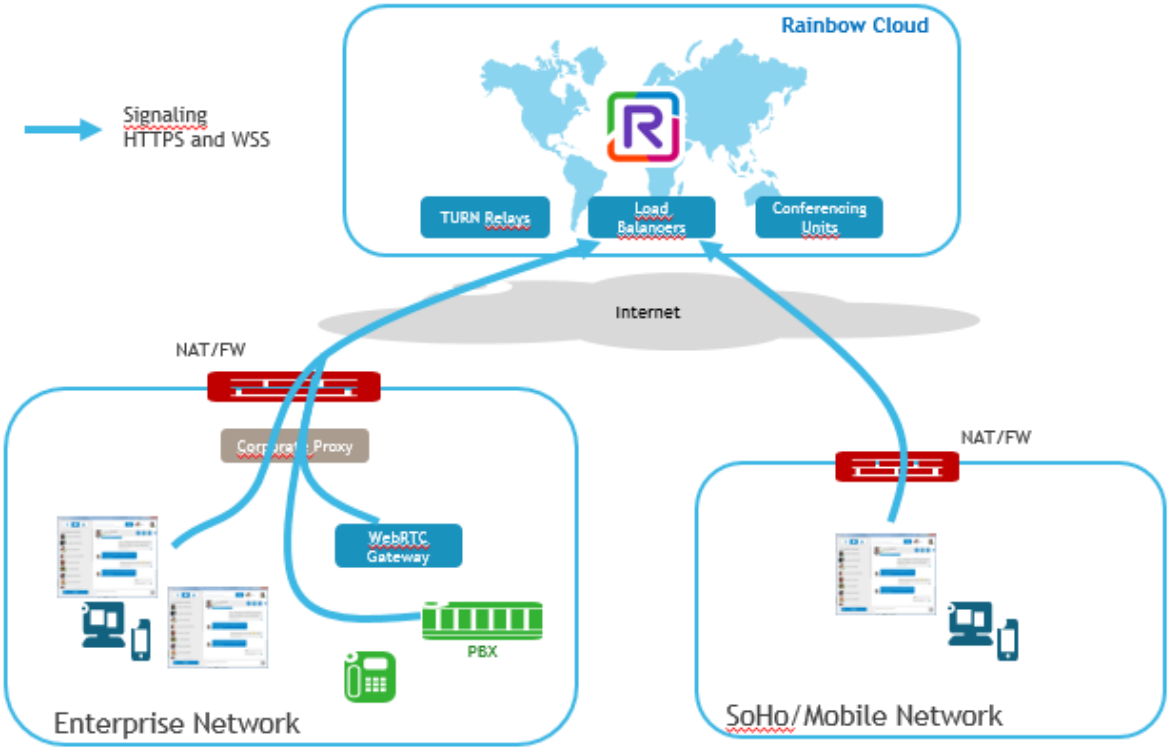
All applications aim at providing the same level of services and features and interact with server-side components for signaling and media. The basic principles are provided in this section, and a detailed list of protocols/ports in the following one.

### 5.2.1 Signaling

For signaling, HTTPS/REST and Secured Web Sockets protocols are used:

- HTTPS (443) for all REST API communications and resources loading.
- Secure Web Sockets (WSS, 443) for all XMPP messages and notifications.

If a HTTP Proxy is configured, HTTP Proxy is used. In such case, HTTP Proxy must support Secured WebSocket (HTTP Upgrade to switch to wss protocol).




Full details on the involved ports are provided in the next sections.

### 5.2.2 WebRTC/Media

Media communications between two clients, or between client and server-side conferencing components, use the WebRTC technology with DTLS-SRTP protocol for encrypting audio, video and desktop sharing media. The solution leverages ICE mechanisms and Rainbow TURN relays to achieve connectivity thru NAT/Firewalls.

ICE (Internet Connectivity Establishment) procedure and STUN/TURN protocols are used to dynamically determine how the media will be routed between two Rainbow clients.

Basically, when a WebRTC communication takes place, client proceeds to the following steps:

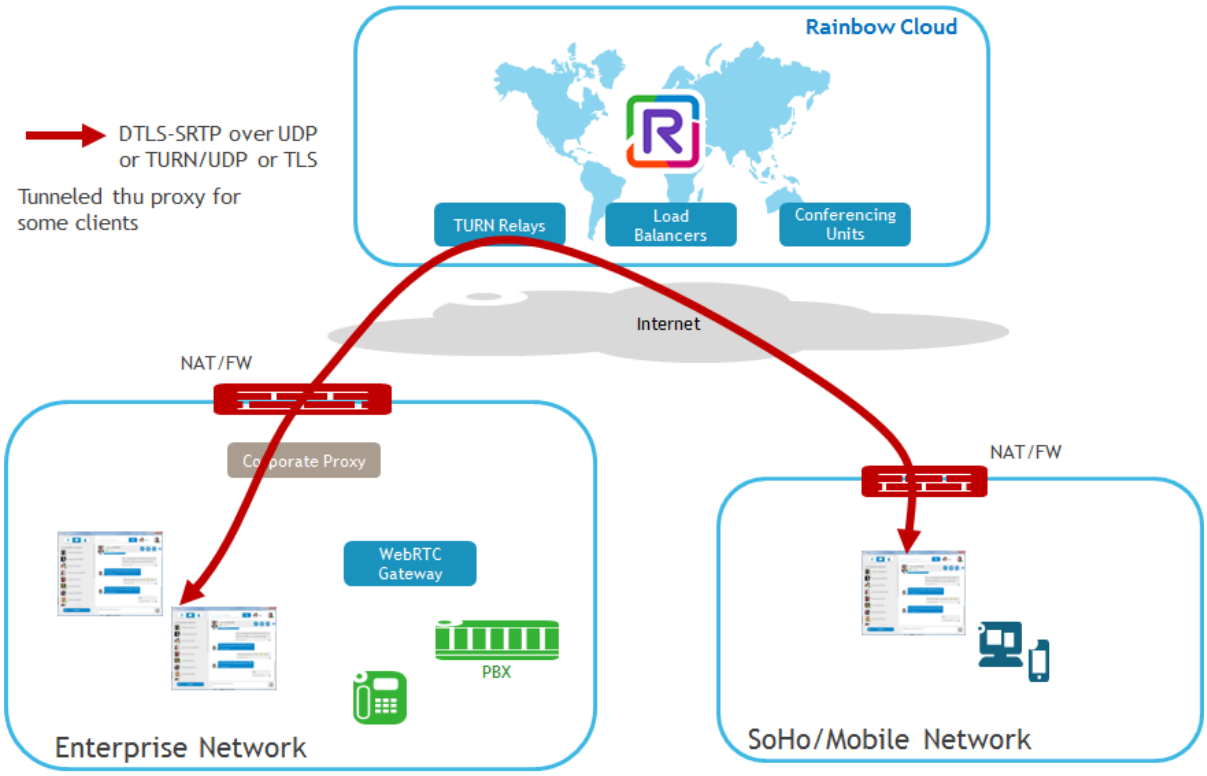
- Each client gathers candidates addresses.
    - A candidate is a transport address, combination of IP address and port for a particular transport protocol, allocated on local interface (for example wired Ethernet interface or WiFi interface for a PC), and on TURN cloud relay server that are necessary to allow cross network communications. The Rainbow infrastructure ensures TURN servers are located in all regions for providing world-wide coverage, however for optimizing the number of candidates for a WebRTC communication, Rainbow clients are automatically using only the nearest two Rainbow TURN servers, based on their IP geo-localization.
- The client exchanges candidates with the distant peer (other client or conferencing unit),
- Clients then check connectivity for candidates between both clients, and select the most optimized working pair.

In case network conditions loss/change during an established communication, the network path for media is automatically renegotiated on-the-fly thru the above ICE mechanisms, allowing to keep communication active with only a small media interruption (no more than a few seconds max for device to change network connection and Rainbow WebRTC stack to perform re-negotiation). This typically happens in case of Wi-Fi/3G-4G handover for mobile devices, or network connectivity change on computer (wired to Wi-Fi connection).

**Example1: P2P WebRTC between local client (Enterprise network) and Remote client (external to the Enterprise network)**

In such a case, a direct connection is not possible and the communication is generally achieved by leveraging a TURN server, acting as a cloud relay for routing media. It is reminded here that TURN relays are simple traffic redirectors, and have no access to the relayed media that remains encrypted end-to-end between peers.

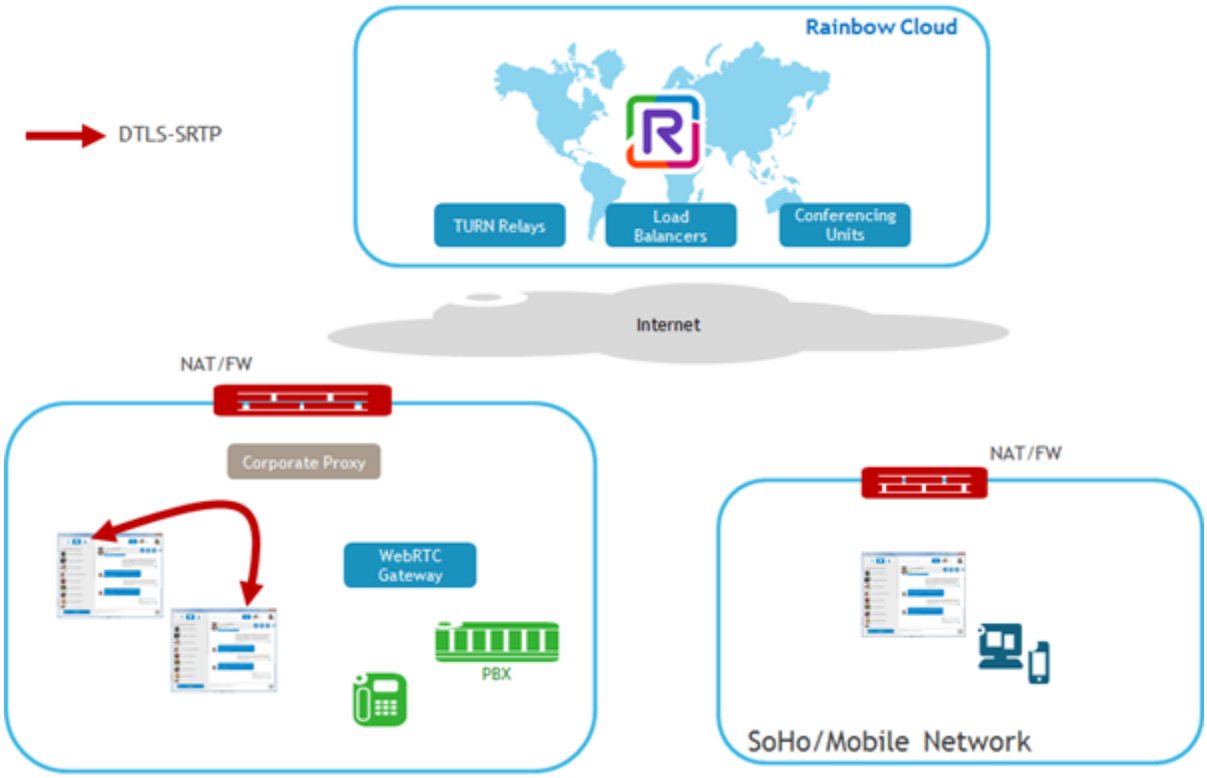
As illustrated below, in case a proxy is used on the enterprise network, the connection to the TURN servers, and consequently the media, can be tunneled thru the proxy for some Rainbow clients (see details in section 0).




Note: to simplify figures, only one TURN server is illustrated. For a P2P communication, depending on geography and network performance, up to two TURN servers could be used to establish a communication (a different one for each client).

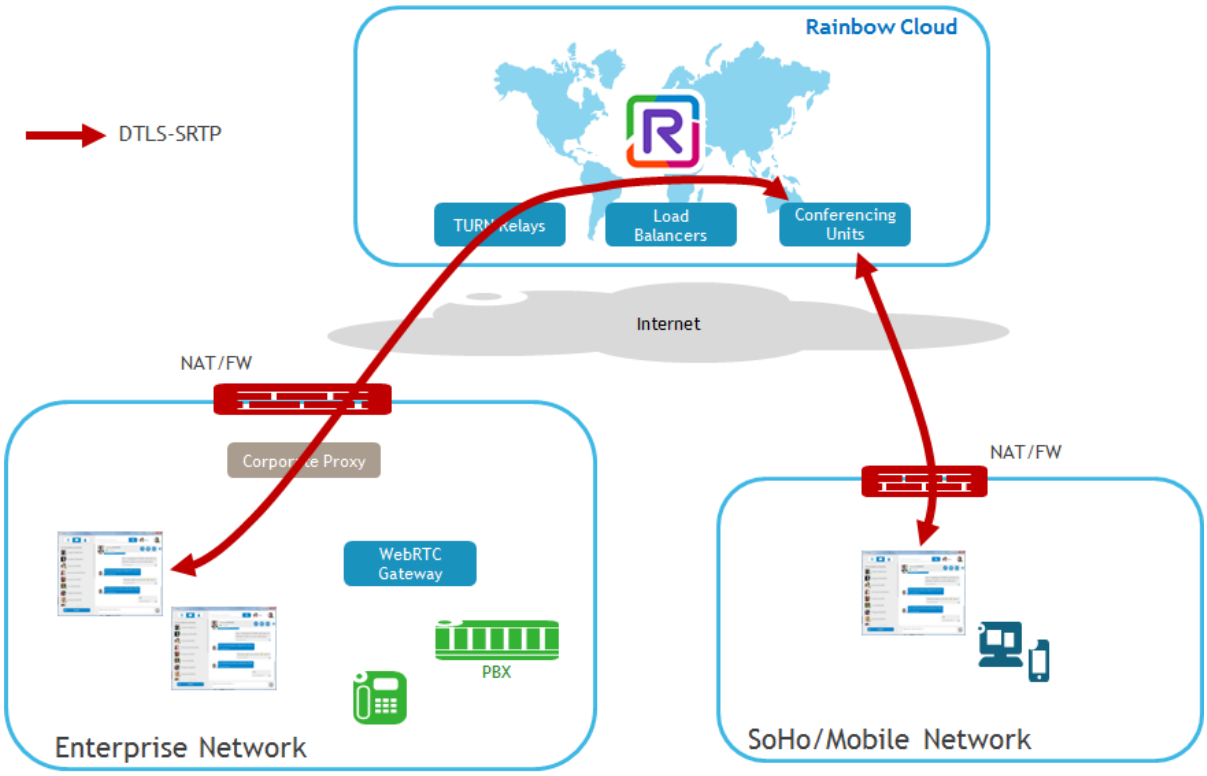
**Example2: P2P WebRTC between two clients located on same LAN (Enterprise network)**

In such a case, a direct connection is possible and the ICE negotiation results in clients choosing the direct path, as preferred path over the one going thru TURN.




**Example3: Media with WebRTC Rainbow conference (Bubble)**

When joining an Audio/Video Rainbow conference (bubble), clients connect to Rainbow cloud Conferencing Unit. Depending on the type of network infrastructure (Firewall/NAT type) on client side, clients either join the conferencing unit directly, or by getting relayed thru a TURN server, typically if UDP is not allowed directly between endpoints and the conferencing unit.




## 5.3 Connections and Ports used

### 5.3.1 Rainbow Desktop and Web clients and Web SDK

Note that IP flows are always at the initiative of the clients, so no inbound firewalls rules from Internet to the enterprise network are needed.

The following connections take place between Rainbow client/Agent and Rainbow Cloud Services, possibly going thru a proxy if one is configured on the computer.

| Protocols | Source | Destination | HTTP Proxy Compatibility |
|---|---|---|---|
| **HTTPS**<br>**(Resources and REST API)** | **Rainbow client**<br>OS dynamic port range (see 5.3.5) | **Rainbow servers,**<br>TLS/443 | **Yes** |
| **Secure Web Sockets - WSS**<br>**(XMPP)** | **Rainbow client**<br>OS dynamic port range | **Rainbow servers,**<br>TLS/443 | **Yes** |
| **DTLS-SRTP for Peer-to-Peer** | **Rainbow client** | **Peer Rainbow client** | Not applicable (such flows remain on LAN) |

| | | | |
|---|---|---|---|
| **WebRTC comm on same LAN**<br>**(Rainbow clients have direct connectivity between each-other)** | OS dynamic port range | UDP OS dynamic port ranges<br>**WebRTC Gateway**<br>UDP 20000-29999 | |
| **DTLS-SRTP for Peer-to-Peer WebRTC comm thru Internet** | **Rainbow client**<br>OS dynamic port range | **Rainbow TURN Servers** using several connectivity alternatives:<br>UDP/3478<br>TLS/443<br>TCP/80 (see note1) | **Yes**, for connections on ports TLS/443 and TCP/80 only<br>(see note1&2) |
| **DTLS-SRTP to Rainbow WebRTC Conference servers** | **Rainbow client**<br>OS dynamic port range | **Rainbow conference servers**<br>UDP port range UDP/49152-65535<br>As fallback if outgoing UDP range is not opened:<br>**Rainbow TURN Servers** using several connectivity alternatives:<br>UDP/3478<br>TLS/443<br>TCP/80 (see note1) | **Yes**, for connections to TURN Servers on ports TLS/443 and TCP/80 only<br>(see note1&2) |

Note1: Firefox does not correctly support TURN-TLS thru proxy at present time, ie version 64 for this document edition. TCP-80 is offered as a workaround, and port 80 must therefore be opened in firewalls for outgoing traffic when Firefox is being used thru a proxy. It is reminded that the usage of port TCP-80 does not imply clear media traffic. This port is only used as transport channel to the TURN server, and the applicative flow conveyed over it is encrypted end-to-end with DTLS-SRTP

Note2: The connection between a browser and the TURN Server, using TCP-80 or TCP-443 ports, are not using HTTP protocol beyond the HTTP CONNECT allowing the proxy to open the tunnel, but STUN/TURN and DTLS-SRTP protocols. In case Deep Packet Inspection is applied on the customer network and expects to examine HTTP traffic, exception rules must be applied for traffic to Rainbow TURN IP addresses, so the DPI gear allows this legitimate Rainbow TURN connections without attempt for intermediate decryption neither HTTP inspection.

### 5.3.2 Rainbow Android and iOS clients and associated SDKs

The flows involved with mobile clients are similar to the ones used by computer apps, at the exception of proxy compatibility for media, and of the push notification channel required for users to properly receive incoming events (IM, call) when the app is not in foreground.

Proxy settings are inherited from device network configuration.

| Protocols | Source | Destination(s) | HTTP Proxy Compatibility |
|---|---|---|---|
| **HTTPS**<br>**(Resources and REST API)** | **Rainbow client**<br>OS dynamic port range (see 5.3.5) | **Rainbow servers**, TLS/443 | **Yes** |
| **Secure Web Sockets - WSS**<br>**(XMPP)** | **Rainbow client**<br>OS dynamic port range | **Rainbow servers**, TLS/443 | **Yes** |
| **DTLS-SRTP for Peer-to-Peer WebRTC comm on same LAN**<br>(Rainbow clients have direct connectivity between each-other) | **Rainbow client**<br>OS dynamic port range | **Peer Rainbow client**<br>UDP OS dynamic port ranges<br>**WebRTC Gateway**<br>UDP 20000-29999 | Not applicable (such flows remain on LAN) |
| **DTLS-SRTP for Peer-to-Peer WebRTC comm thru Internet** | **Rainbow client**<br>OS dynamic port range | **Rainbow TURN Servers** using several connectivity alternatives:<br>UDP/3478<br>TLS/443 | **No**, proxy not supported for media<br>If ports are blocked on the firewall, the app automatically falls back to mobile data network if such connectivity is available<br>For WiFi-only mobile devices, the IS/IT must open UDP/3478 and/or TLS/443 to Rainbow Servers |
| **DTLS-SRTP to Rainbow WebRTC Conference servers** | **Rainbow client**<br>OS dynamic port range | **Rainbow conference servers**<br>UDP port range UDP/49152-65535<br>As fallback if outgoing UDP range is not opened:<br>**Rainbow TURN Servers** using several connectivity alternatives:<br>UDP/3478<br>TLS/443 | |

| | | | |
|---|---|---|---|
| **Apple Push Notification**<br>**(iOS App)** | **Rainbow device**<br>OS dynamic port range | **APNS**<br>TCP/443 (*) | **No**<br>If the ports are not opened on the firewall, the app automatically falls back to mobile data network if such connectivity is available<br>For WiFi-only mobile devices, the IS/IT must open firewall rules to allow direct outgoing traffic to Push Notification ports |
| **Google FCM Push Notification**<br>**(Android app)** | **Rainbow device**<br>OS Dynamic port range | **Google FCM servers**<br>TCP/5228-5229-5230 (**) | |

(*) reference: https://support.apple.com/en-ph/HT203609

(**) reference: https://firebase.google.com/docs/cloud-messaging/concept-options

### 5.3.3 PBX Agents

PBX agents, embedded in ALE PBX or deployed as external component for third party PBX, require connecting to Rainbow Cloud to deliver hybrid telephony services. As for other Rainbow CPE components, all flows are initiated from PBX Agent to Rainbow cloud, avoiding opening any incoming firewall pinholes from Internet to the corporate network.

The following table details IP flows required for the agent to connect to Rainbow servers.

| Protocols | Source | Destination(s) | HTTP Proxy Compatibility |
|---|---|---|---|
| **Secure Web Sockets - WSS** | **PBX Agent**<br>PBX dynamic port range | **Rainbow servers**, TLS/443 | **Yes** |
| **DNS** | **PBX Agent** | **DNS Server** (*)<br>UDP/53 | **No** |

(*) The DNS server, generally located on corporate network, is required to resolve Rainbow server names.

### 5.3.4 WebRTC gateway

The WebRTC Gateway acts as a bridge enabling media communications between Rainbow WebRTC clients and telephony extensions reached thru a PBX. It is deployed on the same network as the PBX.

The following table details the flows involved between the WebRTC Gateway and other Rainbow components.

| Protocols | Source | Destination | HTTP Proxy Compatibility |
|---|---|---|---|
| **HTTPS**<br>**(Resources and REST API)** | **WebRTC Gw**<br>OS dynamic port range (see 5.3.5) | **Rainbow servers**, TLS/443 | **Yes** |
| **Secure Web Sockets - WSS**<br>**(XMPP)** | **WebRTC Gw**<br>OS dynamic port range | **Rainbow servers**, TLS/443 | **Yes** |
| **ICE/TURN(s)**<br>**Media Connectivity Checks** | **WebRTC Gw**<br>UDP 20000-29999 | **Rainbow TURN Servers**<br>UDP/3478 | **Yes***<br>By default using connection to TURN servers on TCP/80, use of TLS/443 is possible for specific cases** |
| **DTLS-SRTP for Peer-to-Peer WebRTC comm on same LAN**<br>(Rainbow clients have direct connectivity between each-other) | **WebRTC Gw**<br>UDP 20000-29999 | **Peer Rainbow client on same private network**<br>UDP OS dynamic port ranges | Not applicable (such flows remain on LAN) |
| **DTLS-SRTP for Peer-to-Peer WebRTC comm thru Internet** | **WebRTC Gw**<br>UDP 20000-29999 | **Rainbow TURN Servers**<br>UDP/3478 | **Yes***<br>By default using connection to TURN servers on TCP/80, use of TLS/443 is possible for specific cases** |

** the WebRTC Gateway supports HTTP proxy for media since version 1.71. On previous versions, outgoing port to TURN servers on UDP/3478 was mandatory. Note that a direct connection to TURN servers, using UDP, remains the recommended way for Voice quality reasons. Routing media thru proxy must be done only when IS/IT does not allow such direct connectivity*

*** Using TCP/80 is the default because the channel to the TURN only conveys DTLS-SRTP encrypted media. Therefore TCP/80 avoids double encryption which impacts performances and may affect QoS. Proxying to TURN servers on TLS/443 is supported as a workaround for proxies or policies that may not allow proxying to TCP/80, but impacts the number of simultaneous communications the component can support*

Besides these flows with the Rainbow ecosystem, the WebRTC Gateway communicates with the PBX ecosystem on the LAN. The flows are described hereafter in case firewalling is applied between WebRTC Gw and the LAN side.

| Protocols | Source | Destination |
|---|---|---|
| **SIP** | **WebRTC Gw**<br>UDP/5060 | **PBX (physical IP address)**<br>UDP/5060 |
| **SIP** | **PBX (physical IP Address)**<br>UDP/5060 | **WebRTC Gw**<br>UDP/5060 |
| **RTP Media** | **WebRTC Gw**<br>UDP 30000-40000 | **PBX Gateway**<br>UDP port range (*) |
| **RTP Media** | **PBX Gateway**<br>UDP port range (*) | **WebRTC Gw**<br>UDP 30000-40000 |
| **RTP Media** | **WebRTC Gw**<br>UDP 30000-40000 | **IP Phones**<br>UDP port range (*) |
| **RTP Media** | **IP Phones**<br>UDP port range (*) | **WebRTC Gw**<br>UDP 30000-40000 |
| **DNS** | **WebRTC Gw**<br>OS Dynamic range | **DNS server**<br>UDP/53 |
| **NTP** | **WebRTC Gw**<br>OS Dynamic range | **NTP server**<br>UDP/123 |
| **SSH**<br>***If enabled*** | **SSH client** | **WebRTC Gw**<br>TCP/22 |

(*) please refer to IP Flows PBX documentation on BPWS, for precisions on gateway and devices port ranges.

### 5.3.5 OS Dynamic port range

As complement to the info provided in the previous sections, the table below reminds the current default dynamic/ephemeral ports ranges used by the different operating systems Rainbow clients can run on. These ports are allocated by the OS and Rainbow apps have no control over this selection.

| Supported Platforms | Dynamic Port Range (UDP and TCP) |
|---|---|
| **Windows** | 49152-65535 |
| **MacOS** | 49152-65535 |

| | |
|---|---|
| **iOS** | 49152-65535 |
| **Android (>=7)** | 37000-50000 |
| **Linux (WebRTC Gw)** | 49512-65535 |

## 5.4 Domains and IP addresses used

Rainbow cloud services use the following domains:

| Used by | Purpose | Domains |
|---|---|---|
| **Rainbow Clients** | Resources (website, images, client package, Agent package, …) | web.openrainbow.com<br>cdn.openrainbow.com<br>meet.openrainbow.com |
| **Rainbow Clients/SDK** | REST API | openrainbow.com |
| **Rainbow Clients/SDK** | XMPP over Secured WebSockets | openrainbow.com |
| **Rainbow Clients/SDK** | STUN/TURN | turn-sjc1.openrainbow.com (US West),<br>turn-dal1.openrainbow.com (US West),<br>turn-wdc1.openrainbow.com (US East),<br>turn-bhs1.openrainbow.com (Canada),<br>turn-sao1.openrainbow.com (Brazil),<br>turn-eri1.openrainbow.com (UK),<br>turn-sbg1.openrainbow.com (France),<br>turn-lim1.openrainbow.com (Germany),<br>turn-che1.openrainbow.com (India),<br>turn-sgp1.openrainbow.com (Singapore),<br>turn-hkg1.openrainbow.com (Hong-Kong),<br>turn-seo1.openrainbow.com (South Korea),<br>turn-tok1.openrainbow.com (Japan),<br>turn-syd1.openrainbow.com (Australia)<br>turn-dli2.openrainbow.cn.com (China)<br>Deprecated, valid until 31th March 2019:<br>turn-eu1.openrainbow.com (France),<br>turn-eu2.openrainbow.com (Germany),<br>turn-na1.openrainbow.com (Canada),<br>turn-as1.openrainbow.com (Singapore),<br>turn-oc1.openrainbow.com (Australia) |
| **PBX agent** | PBX connection to Rainbow | agent.openrainbow.com |
| **WebRTC GW** | PBX media connection to Rainbow | As for Rainbow clients |

| Used by | Purpose | Domains |
|---|---|---|
| **Node Cli and SDKs in development mode** | Sandbox connections for applications development tests | sandbox.openrainbow.com<br>web-sandbox.openrainbow.com |
| **Mail Server** | For mails sent from Rainbow | smtp.openrainbow.com<br>mail.openrainbow.com |

**Nota**: It is highly recommended that customers always use FQDNs, Rainbow servers IP addresses being subject to change. In the unfortunate event that using or whitelisting DNS entries is not an option, the table below references all public IP addresses used by the various Rainbow services. Please keep in mind that this map is subject to change and can be updated at any time without any further notice. Also keep in mind that due to the multiple high-availability and failover mechanisms in place, both at DNS and application levels, it is mandatory to whitelist all the IP addresses aforementioned, including those in regions and geographies that might not be explicitly those intended by end customer.

| Services | Region / Country | Associated IP Addresses |
|---|---|---|
| **Main Load Balancers** | North America | 142.4.216.72<br>167.114.175.31<br>149.56.179.10<br>144.217.123.252<br>169.44.201.2<br>184.173.161.213 |
| | South America | 169.57.174.108 |
| | Europe | 79.137.65.42<br>54.36.121.125<br>178.32.173.187<br>178.33.89.143 |
| | Germany | 54.36.108.169<br>54.36.108.164<br>51.38.111.143<br>51.38.111.145 |
| | Asia | 139.99.122.102<br>139.99.122.99<br>139.99.4.245<br>139.99.4.244<br>169.38.94.246<br>169.56.128.9<br>169.56.36.238 |
| | Oceania | 139.99.148.135<br>139.99.161.35 |
| **TURN Media Relays** | North America / US West | 169.44.201.11 |
| | North America / US Central | 169.46.173.182 |
| | North America / US East | 184.173.161.212 |
| | North America / Canada | 149.56.18.31 |
| | South America / Brazil | 169.57.174.107 |
| | Europe / UK | 145.239.255.197 |
| | Europe / France | 149.202.202.213 |
| | Europe / Germany | 217.182.197.194 |
| | Asia / India | 169.38.94.242 |
| | Asia / Singapore | 139.99.120.111 |
| | Asia / Hong-Kong | 169.56.128.10 |
| | Asia / South-Korea | 169.56.161.6 |

| | | |
|---|---|---|
| | Asia / China | 42.202.135.13 |
| | Asia / Japan | 169.56.36.231 |
| | Oceania / Australia | 139.99.148.99 |
| **Conferencing Media Servers** | North America / Canada | 144.217.75.16 |
| | Europe / UK | 145.239.255.198 |
| | Europe / France | 149.202.200.34 |
| | Europe / Germany | 54.36.108.168 |
| | Asia / Singapore | 139.99.122.114 |
| | Asia / China | 42.202.135.11 |
| **Mail** | Europe / France | 151.80.179.56 |

## 5.5 Bandwidth requirement

### 5.5.1 WebRTC

Rainbow WebRTC communication currently rely on the following codecs:

WebRTC P2P:

- OPUS for audio
- VP8 or H.264 for Video and Screen Sharing,

WebRTC Conference:

- OPUS for audio
- VP8 for Video and Screen Sharing

WebRTC GW

- G711 or G722 for audio

WebRTC codecs are able to dynamically throttle both their resolution and bitrates, depending on network performance observed. Peer to peer (P2P) WebRTC communications maximal resolution is 720p using a web or desktop application and 480p when a mobile is involved. In a WebRTC conference, maximal resolution is 480p. The following table provides bandwidth requirement per media:

| Media Type | Maximal Bandwidth | Average Bandwidth | Lowest Bandwidth | Comment |
|---|---|---|---|---|
| **Audio** | 100 kbps | 80kbps | 15kbps | |
| **Video** | 1.2 Mbps (P2P - 720p)<br>500 kbps (Conference & mobile - 480p) | | | Depends on video resolution / quality |

| | | |
|---|---|---|
| **Screen Sharing** | 15 – 1500 kbps (P2P – 1080p)<br>15 – 800 kbps (Conference & mobile – 720p) | Depends on screen motion |

Note: In a WebRTC conference a participant using the web or Desktop application can receive up to 5 video media streams (4 video from other participants and 1 desktop sharing). For a given participant, the maximum upstream bandwidth is 1.3 Mbps (0.5 + 0.8 Mbps) if the participant shares his video and his desktop. The maximum downstream is 2.8 Mbps (4 x 0.5 + 0.8 Mbps).

## 5.6 Configuration of border elements in enterprise

To allow Rainbow to operate properly, border elements like DNS, HTTP Proxy or Firewall must be configured to allow accessing domains and protocols listed in the table chapter 5.3 and 5.4.

In case Deep Packet Inspection is in place on the network, some exceptions might have to be configured according to the Note of 5.3.1.

# 6 Limitations, Restrictions and workarounds

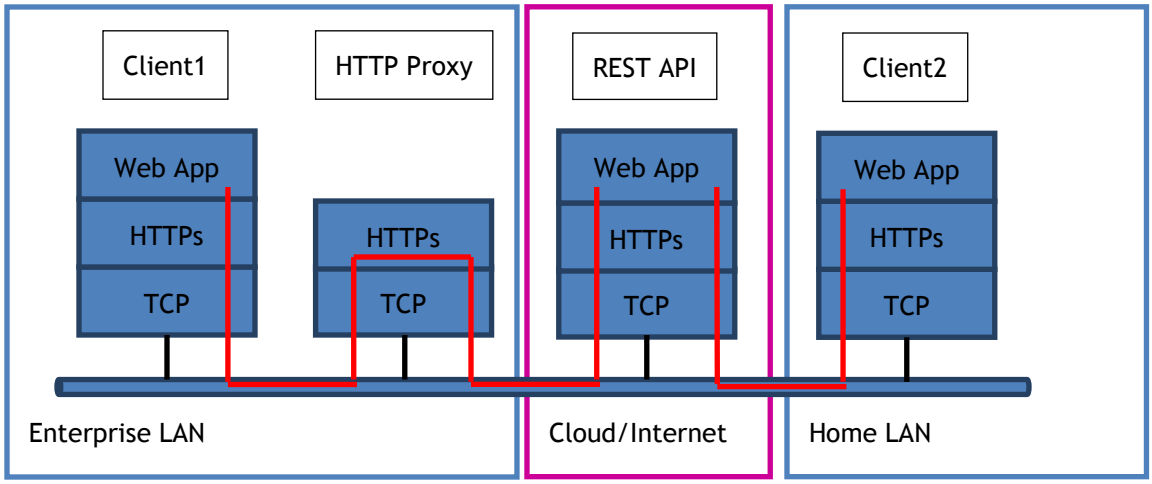
## 6.1 HTTP Proxy

If an HTTP Proxy is configured and/or selected at OS level (fixed configuration, auto-detect, script, …), by design, Browsers and Rainbow desktop client as well will always rely on this HTTP Proxy to reach Rainbow cloud services (for all protocols used, including HTTPS/REST, XMPP over Secured Web Sockets and TURN). It could append that the proxy in place in enterprise is not able or is not willing to pass-through some protocols (TURN for example). Capability and willingness of the HTTP Proxy must be checked by IS/IT team in case of trouble to start Rainbow clients or to establish Audio/Video call between Enterprise Lan and external users.

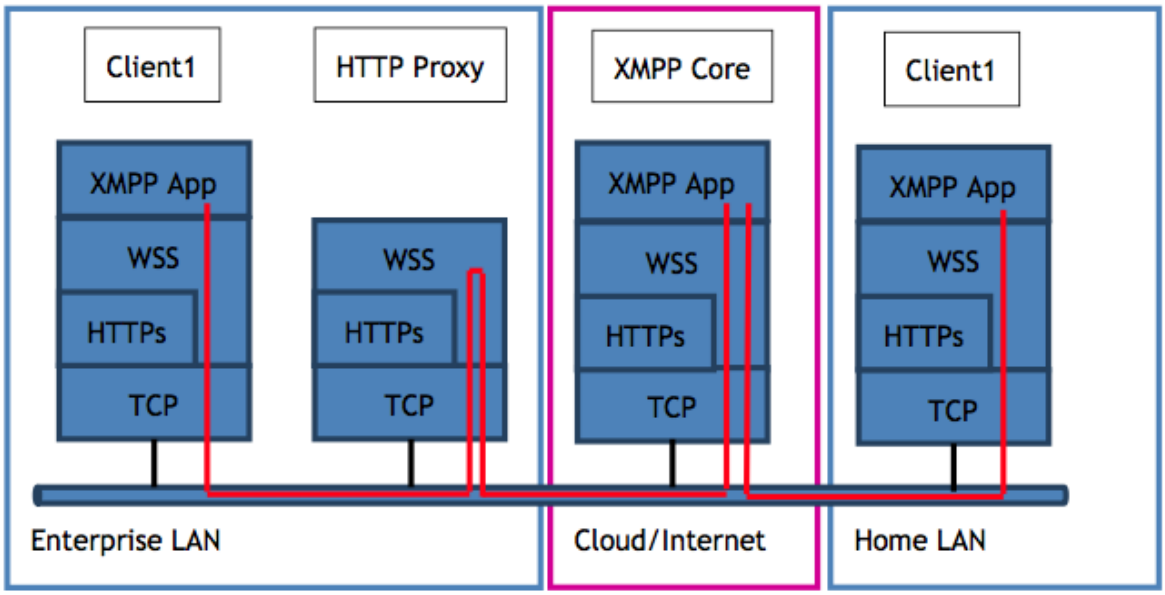
# 7 Annexes: Detailed call-flow of HTTPS/REST, XMPP and ICE connections

The following figures illustrate a case where Rainbow Client1 make an Audio/Video call to Rainbow Client2. Rainbow Client1 is in an enterprise environment with NAT/FW and HTTP Proxy border elements. Rainbow Client2 is in a Home network with simple NAT/FW as border element (home router/box).

Network layers (Rest API):

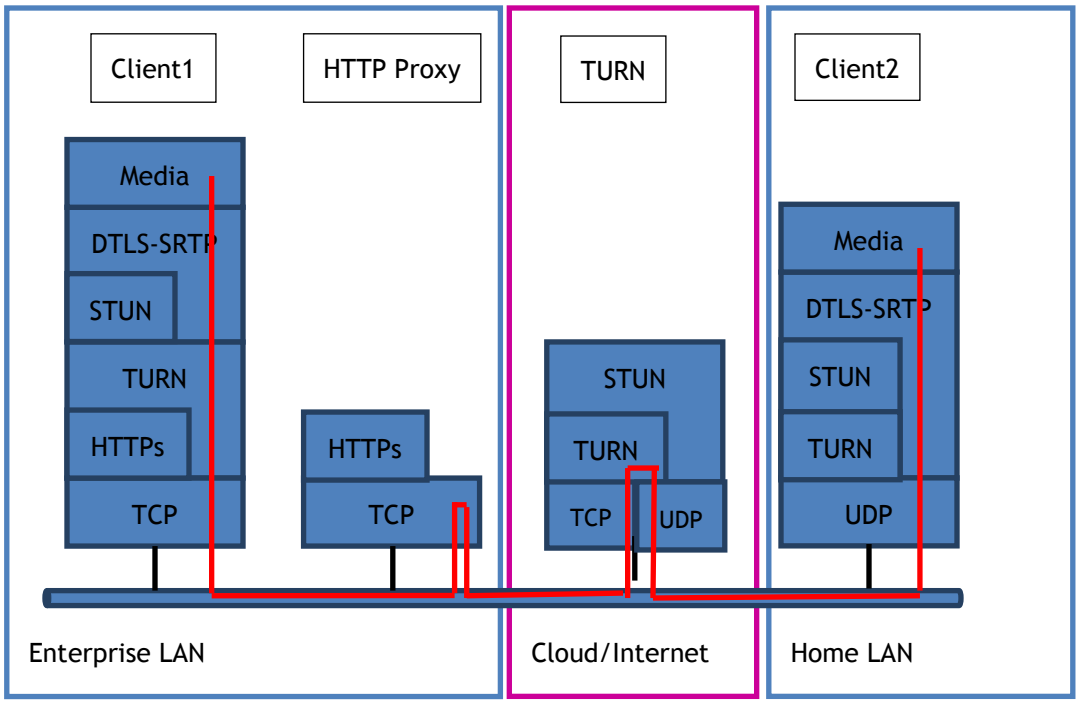



Network layers (XMPP):




HTTPs is used to setup WSS protocol (RFC6455)

Network layers (Media):




HTTPs is used to setup HTTP Tunnel to TURN server

STUN is a protocol helper to check connectivity

TURN is a protocol helper to pass-through NAT/FW/HTTP Proxy

## Call Flow:

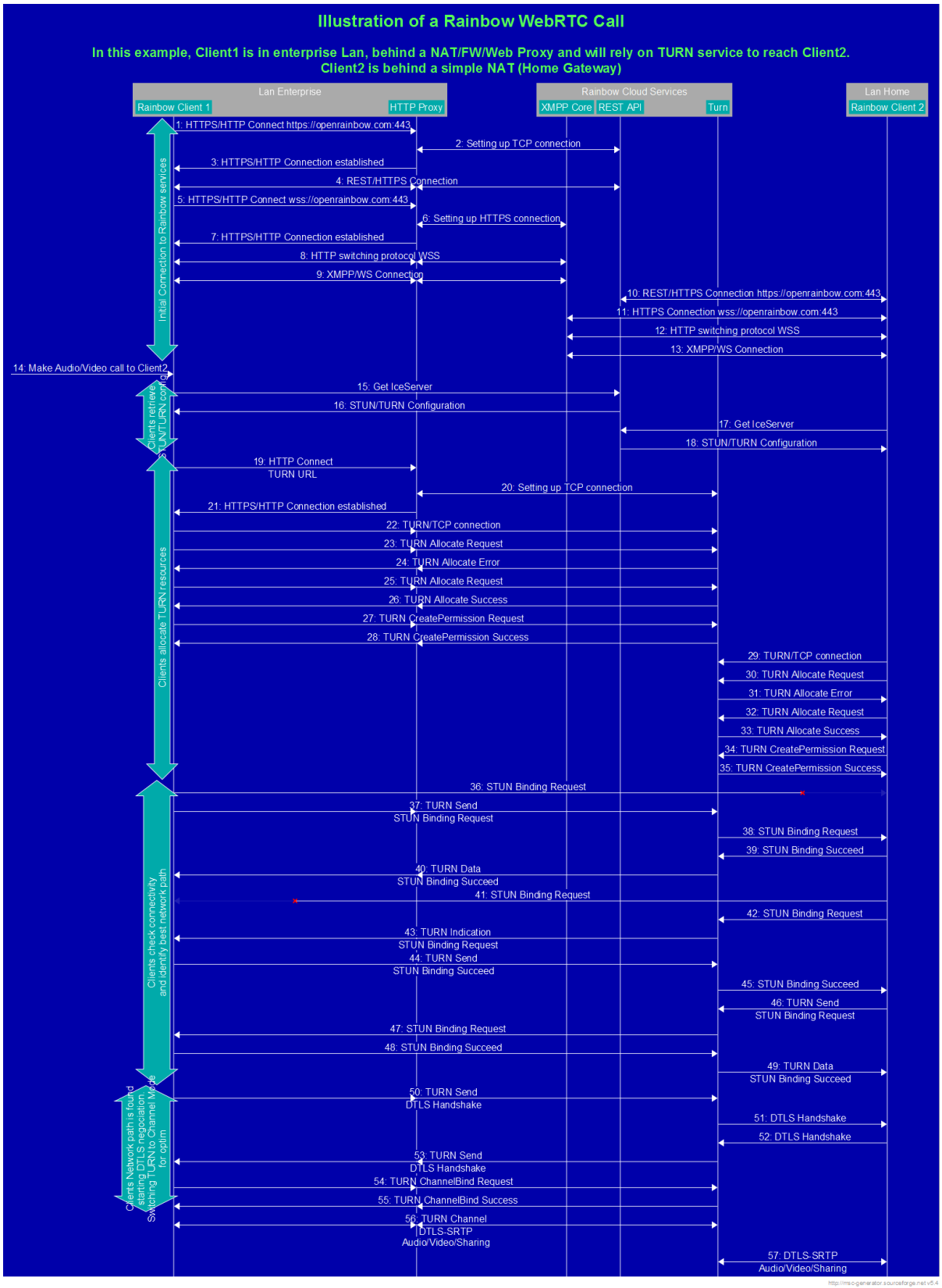


Illustration of a Rainbow WebRTC Call
In this example, Client1 is in enterprise Lan, behind a NAT/FW/Web Proxy and will rely on TURN service to reach Client2.
Client2 is behind a simple NAT (Home Gateway)
Lan Enterprise
Rainbow Client 1
HTTP Proxy
Rainbow Cloud Services
XMPP Core
REST API
Turn
Lan Home
Rainbow Client 2
Initial Connection to Rainbow services
1: HTTPS/HTTP Connect https://openrainbow.com:443
2: Setting up TCP connection
3: HTTPS/HTTP Connection established
4: REST/HTTPS Connection
5: HTTPS/HTTP Connect wss://openrainbow.com:443
6: Setting up HTTPS connection
7: HTTPS/HTTP Connection established
8: HTTP switching protocol WSS
9: XMPP/WS Connection
10: REST/HTTPS Connection https://openrainbow.com:443
11: HTTPS Connection wss://openrainbow.com:443
12: HTTP switching protocol WSS
13: XMPP/WS Connection
14: Make Audio/Video call to Client2
Clients retrieve STUN/TURN config
15: Get IceServer
16: STUN/TURN Configuration
17: Get IceServer
18: STUN/TURN Configuration
Clients allocate TURN resources
19: HTTP Connect
TURN URL
20: Setting up TCP connection
21: HTTPS/HTTP Connection established
22: TURN/TCP connection
23: TURN Allocate Request
24: TURN Allocate Error
25: TURN Allocate Request
26: TURN Allocate Success
27: TURN CreatePermission Request
28: TURN CreatePermission Success
29: TURN/TCP connection
30: TURN Allocate Request
31: TURN Allocate Error
32: TURN Allocate Request
33: TURN Allocate Success
34: TURN CreatePermission Request
35: TURN CreatePermission Success
Clients check connectivity and identify best network path
36: STUN Binding Request
37: TURN Send
STUN Binding Request
38: STUN Binding Request
39: STUN Binding Succeed
40: TURN Data
STUN Binding Succeed
41: STUN Binding Request
42: STUN Binding Request
43: TURN Indication
STUN Binding Request
44: TURN Send
STUN Binding Succeed
45: STUN Binding Succeed
46: TURN Send
STUN Binding Request
47: STUN Binding Request
48: STUN Binding Succeed
49: TURN Data
STUN Binding Succeed
Clients Network path is found starting DTLS negociation Switching TURN to Channel Mode for optim
50: TURN Send
DTLS Handshake
51: DTLS Handshake
52: DTLS Handshake
53: TURN Send
DTLS Handshake
54: TURN ChannelBind Request
55: TURN ChannelBind Success
56: TURN Channel
DTLS-SRTP
Audio/Video/Sharing
57: DTLS-SRTP
Audio/Video/Sharing

- Steps 1-9: Client1, behind HTTP Proxy, establish HTTPs sockets and Secured Web Sockets (for XMPP) through the HTTP Proxy.
- Steps 10-13: Client2, behind simple NAT, establish regular HTTPs sockets and Secured Web Sockets (for XMPP) directly to Rainbow Cloud Services.
- Step 14: Client1 make an Audio/Video call to Client2
- Steps 15-18: Client1 and Client2 retrieves ICE configuration (list of TURN servers)
- Steps 19-28: Client1 allocates TURN resources (through HTTP Proxy)
- Steps 29-35: Client2 allocates TURN resources (direct connection to TURN)
- Steps 36-49: Client1 and Client2 perform STUN connectivity check for various options (in this example, we illustrate the case where direct STUN connectivity check would failed)
- Steps 50-53: Client1 and Client2 have found a network path through the TURN server. They start to initiate the DTLS-SRTP handshake.
- Steps 54-55: Client1 ask TURN to switch to Channel mode to optimize network bandwidth (reduce TURN header overhead).
- Steps 56-57: DTLS-SRTP is established, Audio/Video media starts to flow between Client1 and Client2

*End of Document*