ALCATEL-LUCENT RAINBOW™

HDS Network Requirements

GETTING STARTED GUIDE Ed 21

NOVEMBER 2021

Author: Operations - Cloud Services

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
HDS :	Hébergement des données de santé

1 Introduction

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

Alcatel-Lucent Enterprise (ALE) is introducing Alcatel-Lucent Rainbow, an overlay cloud service operated by ALE.

HDS 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.

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

This document describes the high level principles of network flows implemented by the solution, provides detailed information on Rainbow network connectivity requirements, allowing network and security administrators to identify needed firewall rules, verify bandwidth availability, and tune intermediate security elements such as proxies when applicable.

2 Document History

Modifications (see last edition's changes in green)	**Date**	**Edition**
Add new file sharing DNS entries Add webinar DNS entry Precision of HTTP version supported by WRG (HTTP 1.1)	2021-11-03	Ed21
Addition of information for Teams and Google connectors Some fixes in WebRTC Gateway port ranges Some precisions on proxy dimensioning aspects Streamline document paragraph numbering Edition numbering aligned with Cloud edition	2021-09-29	Ed20
DC resilience after OVH incident	29/06/2021	Ed 17
Creation of document	17/09/2020	Ed 16

3 Solution Overview

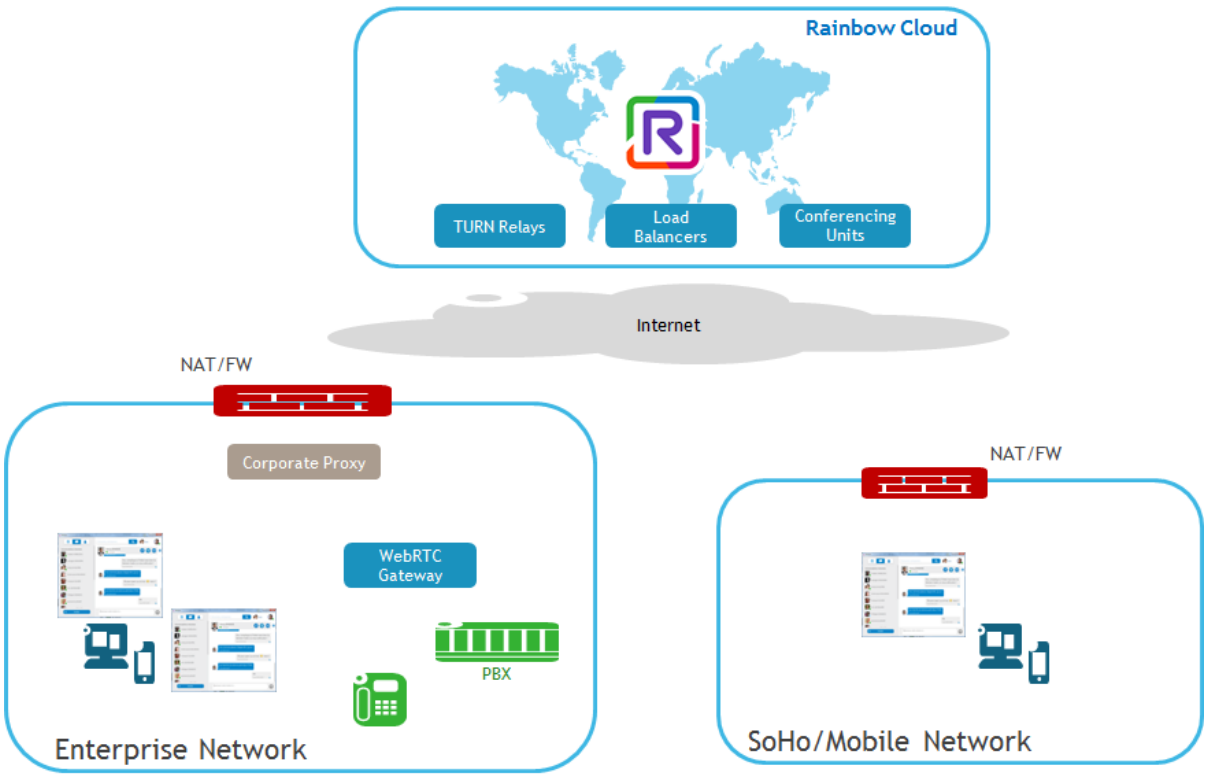
3.1 Global Overview

The HDS 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.health)
-

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



3.2 Used Protocols and High-Level Principles

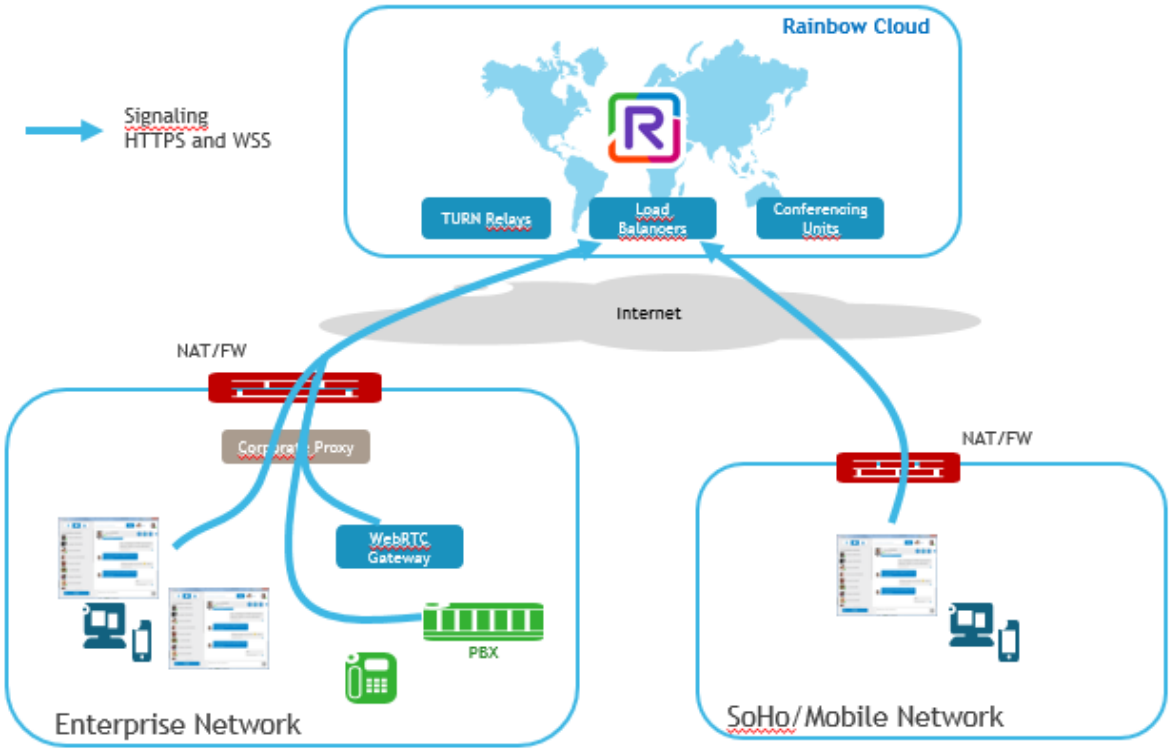
All applications aim at providing the same level of services and features and interact with server-side components for signaling and media. The basic principles are provided in this section, and a detailed list of protocols/ports in the following one.

3.2.1 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.

3.2.2 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.

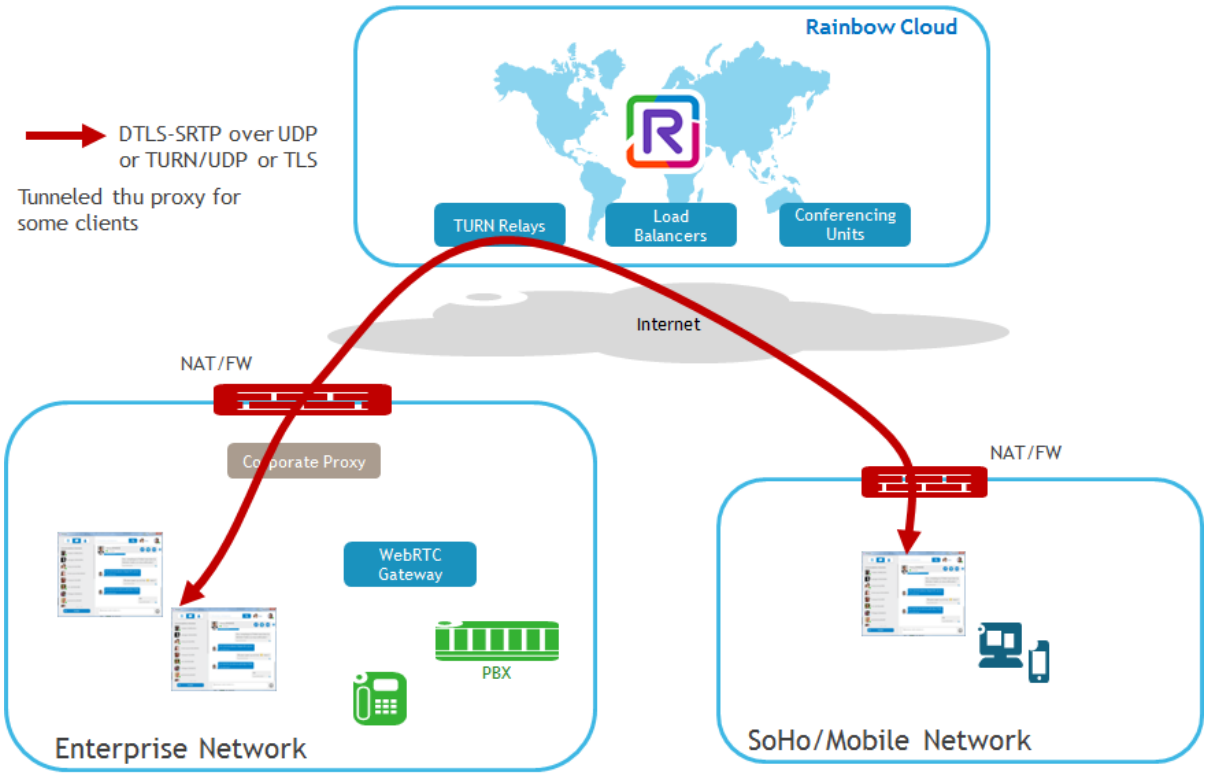
- The client exchanges candidates with the distant peer (other client or conferencing unit),
- Clients then check connectivity for candidates between both clients and select the most optimized working pair.

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.

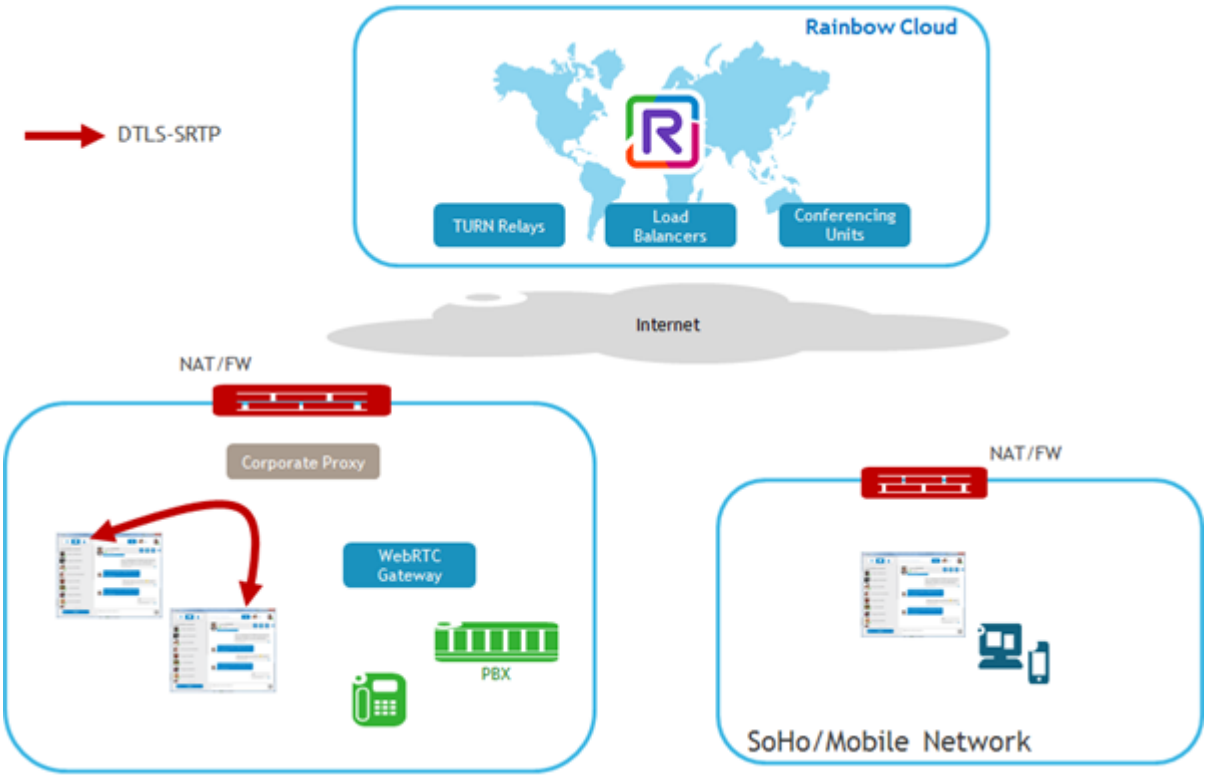
As illustrated below, in case a proxy is used on the enterprise network, the connection to the TURN servers, and consequently the media, can be tunneled thru the proxy for some Rainbow clients (see details in section **Error! Reference source not found.**).



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).

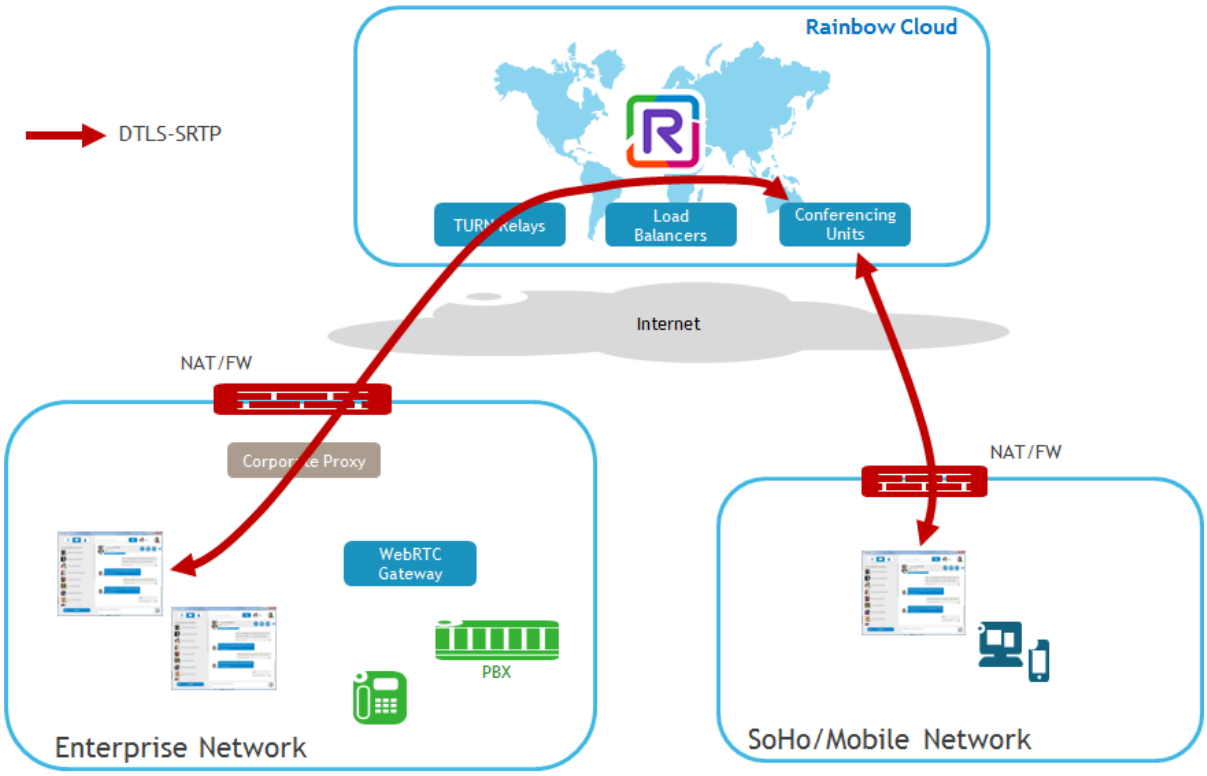
Example2: P2P WebRTC between two clients located on same LAN (Enterprise network)

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.

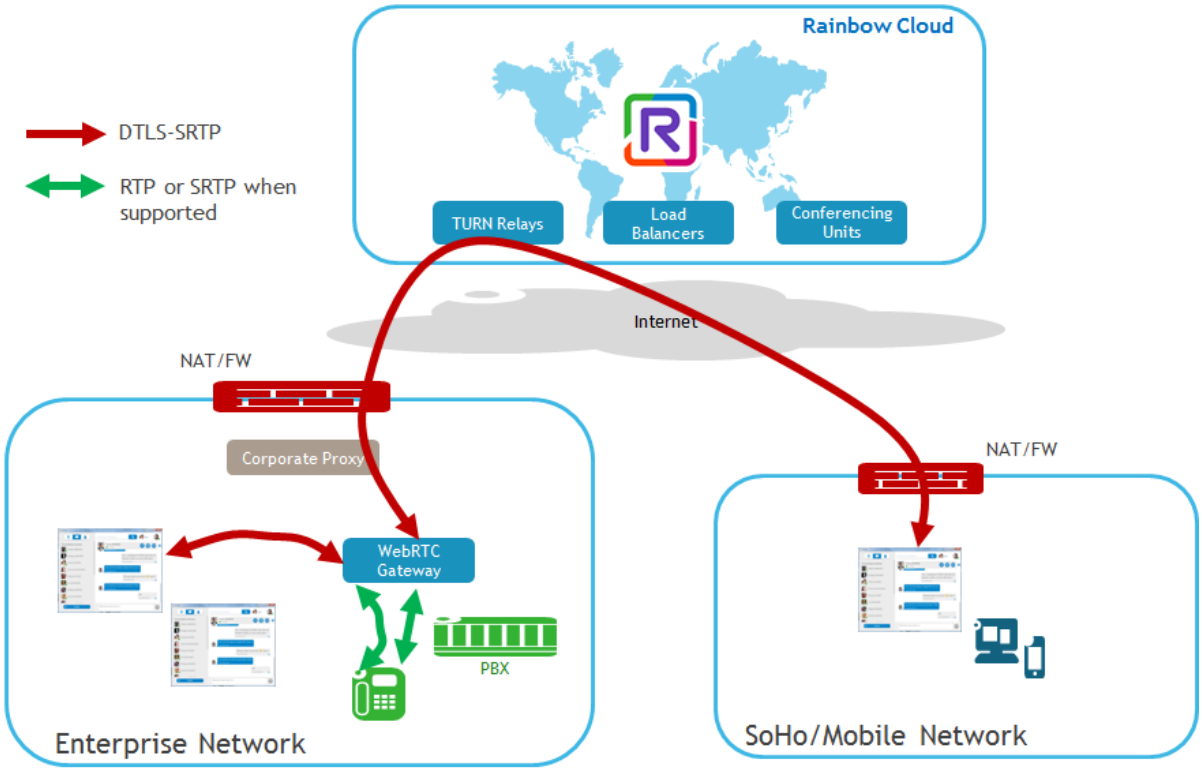


3.2.3 WebRTC for Hybrid Softphony calls

In hybrid mode, PBX telephony calls placed or taken with Rainbow clients also use the WebRTC technology and principles described in the above section, where one end of the WebRTC media call is the WebRTC Gateway. The latter acts as gateway between the Rainbow client used for the softphone call, WebRTC being used between Rainbow client and Gateway, whereas the gateway interfaces with the PBX in SIP and with PBX devices in RTP.

As for previous examples, the media path depends on whether the WebRTC Gateway and the Rainbow client are located on a same network and can therefore have a direct connection between each other. If this connection is possible, the call is direct between the WebRTC Gateway and the Rainbow application. If no direct connectivity exists, then the call is relayed thru a TURN server.

The two scenarios are depicted below.



4 List of used Protocols and Ports

4.1 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
Signaling and APIs (mandatory)			
HTTPS **(Resources and REST API. Apps updates)**	**Rainbow client** OS dynamic port range (see 5.3.5)	**Rainbow servers,** TLS/443	**Yes**
Secure Web Sockets - WSS	**Rainbow client**	**Rainbow servers,** TLS/443	**Yes**

(XMPP)	OS dynamic port range		
Pure WebRTC Audio/Video/ScreenSharing			
DTLS-SRTP for Peer-to-Peer WebRTC comm on same LAN **(Rainbow clients have direct connectivity between each-other)**	**Rainbow client** OS dynamic port range	**Peer Rainbow client** UDP OS dynamic port ranges	Not applicable (such flows remain on LAN)
DTLS-SRTP for Peer-to-Peer WebRTC comm thru Internet	**Rainbow client** OS dynamic port range	**Rainbow TURN Servers** using several connectivity alternatives: UDP/3478 TLS/443 TCP/80 (see note1) (see note3)	**Yes**, requesting proxy to connect to TURN servers via ports TLS/443 and TCP/80 (see note1&2)
DTLS-SRTP to Rainbow WebRTC Conference servers	**Rainbow client** OS dynamic port range	**Rainbow conference servers** UDP port range UDP/49152-65535 As fallback if outgoing UDP range is not opened: **Rainbow TURN Servers** using several connectivity alternatives: UDP/3478 TLS/443 TCP/80 (see note1)	**Yes**, requesting proxy to connect to TURN servers via ports TLS/443 and TCP/80 (see note1&2)
Hybrid Softphony (on ALE OXE/OXO/OTEC or supported 3d party PBX)			
DTLS-SRTP for Softphone call when the app is on LAN and can reach the WebRTC Gateway	**Rainbow client** OS dynamic port range	**WebRTC Gateway** UDP 20000-29999	Not applicable (such flows remain on LAN)
DTLS-SRTP for Softphone call when the app is not on same LAN as WebRTC Gateway	**Rainbow client** OS dynamic port range	**Rainbow TURN Servers** using several connectivity alternatives: UDP/3478 TLS/443	**Yes**, requesting proxy to connect to TURN servers via ports TLS/443 and TCP/80 (see note1&2)

		TCP/80 (see note1) (see note3)	

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: DPI compatibility. 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.

Note3: The firewall will actually see tentative traffic to other destinations/ports, but the above table only lists the minimum required ports for ensuring correct functional behavior with less possible opened rules. Indeed, ICE connectivity checks, as described in 5.2.2, test all possible combinations of IP/ports between the local client and the remote peer candidates. As each peer generally provides candidates with local address (LAN), public relayed address (exposed at a TURN Server), and reflexive address (corresponding to their Internet access gateway), STUN connectivity checks are exchanged between all possible pairs, some being between the WebRTC Gateway and any possible remote IP address and port, which would be enabled by opening a firewall rule towards <any>/<any> IP/port tuple. Only allowing traffic to the TURN server avoids opening such a wide rule. It results in part of the connectivity checks to fail (the ones targeted at the peer reflexive address for example) but ensures that a media path is always found via a rainbow TURN server.

4.2 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
Signaling and APIs (mandatory)			
HTTPS **(Resources and REST API)**	**Rainbow client** OS dynamic port range (see 5.3.5)	**Rainbow servers,** TLS/443	**Yes**
Secure Web Sockets - WSS **(XMPP)**	**Rainbow client** OS dynamic port range	**Rainbow servers,** TLS/443	**Yes**
Apple Push Notification **(iOS App)**	**Rainbow device** OS dynamic port range	**APNS** TCP/443 (*)	**No** If the ports are not opened on the firewall, the app automatically falls back to mobile data network if such connectivity is available 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 **(Android app)**	**Rainbow device** OS Dynamic port range	**Google FCM servers** TCP/5228-5229-5230 (**)	
Pure WebRTC Audio/Video/ScreenSharing			
DTLS-SRTP for Peer-to-Peer WebRTC comm on same LAN (Rainbow clients have direct connectivity between each-other)	**Rainbow client** OS dynamic port range	**Peer Rainbow client** UDP OS dynamic port ranges	Not applicable (such flows remain on LAN)
DTLS-SRTP for Peer-to-Peer	**Rainbow client**	**Rainbow TURN Servers** using several connectivity alternatives:	**No**, proxy not supported for media

WebRTC comm thru Internet	OS dynamic port range	UDP/3478 TLS/443 (note3)	If ports are blocked on the firewall, the app automatically falls back to mobile data network if such connectivity is available 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** OS dynamic port range	**Rainbow conference servers** UDP port range UDP/49152-65535 As fallback if outgoing UDP range is not opened: **Rainbow TURN Servers** using several connectivity alternatives: UDP/3478 TLS/443	
Hybrid Softphony (on ALE OXE/OXO/OTEC or supported 3d party PBX)			
DTLS-SRTP for WiFi Softphone call when the app is on LAN and can reach the WebRTC Gateway	**Rainbow client** OS dynamic port range	**WebRTC Gateway** UDP 20000-29999	Not applicable (such flows remain on LAN)
DTLS-SRTP for Softphone call when the app is not on LAN as WebRTC Gateway	**Rainbow client** OS dynamic port range	**Rainbow TURN Servers** using several connectivity alternatives: UDP/3478 TLS/443 (note3)	**No**, proxy not supported for media If ports are blocked on the firewall, the app automatically falls back to mobile data network if such connectivity is available For WiFi-only mobile devices, the IS/IT must open UDP/3478 and/or TLS/443 to Rainbow Servers

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

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

(note3): see note3 of 4.1

4.3 Rainbow Teams and Google Connectors

Teams and Google connectors rely on Rainbow Web SDK and allow leveraging Rainbow Hybrid and Cloud telephony capabilities from within Microsoft and Google software suites. The network flows involved are identical to the ones of the Rainbow desktop/web, please refer to section 4.1.

4.4 Rainbow Room

Rainbow rooms are Android clients and therefore rely on the same flows as Android smartphones applications.

Protocols	Source	Destination(s)	HTTP Proxy Compatibility
Signaling and APIs (mandatory)			
HTTPS **(Resources and REST API)**	**Rainbow client** OS dynamic port range (see 4.7)	**Rainbow servers**, TLS/443	**Yes**
Secure Web Sockets - WSS **(XMPP)**	**Rainbow client** OS dynamic port range	**Rainbow servers**, TLS/443	**Yes**
Google FCM Push Notification **(Android app)**	**Rainbow device** OS Dynamic port range	**Google FCM servers** TCP/5228-5229-5230 (**)	**No** The IS/IT must open firewall rules to allow direct outgoing traffic to Push Notification ports
Pure WebRTC Audio/Video/ScreenSharing			
DTLS-SRTP for Peer-to-Peer WebRTC comm on same LAN **(Rainbow clients have direct connectivity between each-other)**	**Rainbow client** OS dynamic port range	**Peer Rainbow client** UDP OS dynamic port ranges	Not applicable (such flows remain on LAN)
DTLS-SRTP for Peer-to-Peer WebRTC comm thru Internet	**Rainbow client** OS dynamic port range	**Rainbow TURN Servers** using several connectivity alternatives: UDP/3478 TLS/443	**No**, proxy not supported for media The IS/IT must open UDP/3478 and/or

DTLS-SRTP to Rainbow WebRTC Conference servers	**Rainbow client** OS dynamic port range	**Rainbow conference servers** UDP port range UDP/49152-65535 As fallback if outgoing UDP range is not opened: **Rainbow TURN Servers** using several connectivity alternatives: UDP/3478 TLS/443	TLS/443 to Rainbow Servers

4.5 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** PBX dynamic port range	**Rainbow servers**, TLS/443	**Yes**
DNS	**PBX Agent**	**DNS Server** (*) UDP/53	**No**

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

4.6 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 **(Resources and REST API)**	**WebRTC Gw** OS dynamic port range (see 4.7)	**Rainbow servers**, TLS/443	**Yes**
Secure Web Sockets - WSS **(XMPP)**	**WebRTC Gw** OS dynamic port range	**Rainbow servers**, TLS/443	**Yes**
ICE/TURN(s) **Media Connectivity Checks**	**WebRTC Gw** UDP 20000-29999	**Rainbow TURN Servers** UDP/3478	**Yes*** 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 (Rainbow clients have direct connectivity between each-other)	**WebRTC Gw** UDP 20000-29999	**Peer Rainbow client on same private network** UDP OS dynamic port ranges	Not applicable (such flows remain on LAN)
DTLS-SRTP for Peer-to-Peer WebRTC comm thru Internet	**WebRTC Gw** UDP 20000-29999	**Rainbow TURN Servers** UDP/3478	**Yes*** 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, by having the admin use the mpmproxy command so proxy is used instead of direct connection to TURN UDP/3478. Note that a direct connection to TURN servers, using UDP, remains the recommended way for Voice quality reasons, as it avoids TCP-related retransmissions if the network experiences packet loss. **Connection via proxy requires the proxy to implement HTTP version 1.1***

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

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

4.7 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 Domains and IP addresses used

This section lists the domains and IP addresses used for the Rainbow HDS service.

To simply your firewall configuration, please consider allowing all subdomains of openrainbow.com and create a rule allowing any connection to *.openrainbow.com (wildcard). As alternative, you'll find below the current list of services and domains.

Used by	Purpose	Domains
Rainbow Clients	Resources (website, images, client package, Agent package, …)	web.openrainbow.health cdn.openrainbow.health meet.openrainbow.health webinar.openrainbow.health
Rainbow Clients/SDK	REST API	openrainbow.health
Rainbow Clients/SDK	XMPP over Secured WebSockets	openrainbow.health
Rainbow Clients/SDK	STUN/TURN	turn*.openrainbow.health
PBX agent	PBX connection to Rainbow	agent.openrainbow.health
WebRTC GW	PBX media connection to Rainbow	As for Rainbow clients rtc*.openrainbow.health
Rainbow Clients/SDK	File sharing	files-hds.openrainbow.health
Mail Server	For mails sent from Rainbow	smtp.openrainbow.health mail.openrainbow.health

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	France	51.83.83.128/27 and 135.125.46.128/27
TURN Media Relays	Europe / France	
Conferencing Media Servers	Europe / France	
Mail	Europe / France	

6 Bandwidth requirement

6.1 WebRTC for Rainbow peer-to-peer calls and multiparty conferences

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

These WebRTC codecs are able to dynamically throttle both their resolution and bitrates, depending on network performance observed.

Peer to peer (P2P) WebRTC communications video resolution is 720p.

For Rainbow video conferences, Rainbow clients allows users to decide the way they want to see other participants video, ending up with different possible views:

- Active talker only: 1 high resolution 720p video displayed (if bandwidth permits). If screensharing is done in parallel, the active talker video is displayed in lower resolution
- Active talker with additional thumbnails: 1 high resolution 720p video (if bandwidth permits), and 180p videos for thumbnails[(1)]. If screensharing is done in parallel, the active talker video is also displayed in lower resolution. Web/Desktop application supports up to 5 thumbnails, 7 for Rainbow Room. This mode is not supported on mobile/tablet apps.
- Grid view with up to 12 videos for Web/Desktop or Room, and up to 6 for mobile applications): lower resolution (360p or 180p)[(1)] video streams (depending on network conditions)

The logic to manage different video resolutions relies on a simulcast technique. In upstream direction, depending on the available bandwidth and on server-side instructions, clients send up to 3 different video streams in different resolutions, enabling remote applications to subscribe to the most

appropriate and optimized stream according to the selected view (active talker full screen, thumbnails or grid). Sending simulcast streams is currently only supported by Web/Desktop application, but all applications implement the selection of the appropriate stream on receiver end.

The following table provides maximum bandwidth requirement per media, from the perspective of a Rainbow application. Note that Rainbow sets an upper limit to the bandwidth consumed for video in 720p, 360p and 180p to respectively 800 kbps, 300 kbps and 100 kbps.

Media Type	Maximal Bandwidth	Average Bandwidth	Lowest Bandwidth	Comment
Audio **(bi-directional)**	100 kbps	40kbps	15kbps	
Screen Sharing **(upstream for person who shares, downstream for others)**	1.5 Mbps (720p)			Depends on screen motion figures here are max
Video p2p **(bi-directional** assuming the two person show their video**)**	P2P all Clients: 800 Kbps (720p)			actual bandwidth depends on network conditions, figures here are max
Video Conference **Downstream direction**	**Active talker-only view**: 800 Kbps (720p) **Act talk + thumbnails**: 800Kbps (720p) + N*100kbps (180p)(1) *(max N = 5 for web/desk, 7 for Room, N/A mobiles)* **Grid view**: 12 * 100 kbps (180p)(1)			actual bandwidth depends on network conditions, figures here are max
Video Conference **Upstream direction**	Video upstream for Web/Desktop: 1.2 Mbps (720p+360p+180p) Video upstream for Mobile/Room: 800 Kbps (720p)(2) Sharing upstream: 1.5Mbps (720p)			actual bandwidth depends on network conditions, figures here are max

(1) if video is originated from an app that does not support simulcast, max video thumbnail is 720p rather than 180p

(2) to allow users to limit the used bandwidth on mobile devices where data plans are expensive, a parameter can be set to limit the video bandwidth (480p instead of 720p for upstream video, and downstream video throttled to 500 kbps).

6.2 Hybrid softphony calls

Business calls made or taken from CPE PBX and through the WebRTC gateway use G711 or G722 codecs for audio.

These codecs run at 64 kbps rate, which with addition of UDP and IP headers leads to 87.5kps.

It is reminded that real-time voice media is sensitive to the network quality, and that a good quality communication with G711 requires:

- One-way latency to be maximum 150ms
- Jitter to be maximum 30ms
- Packet loss to be maximum 1%

6.3 PBX Agent traffic

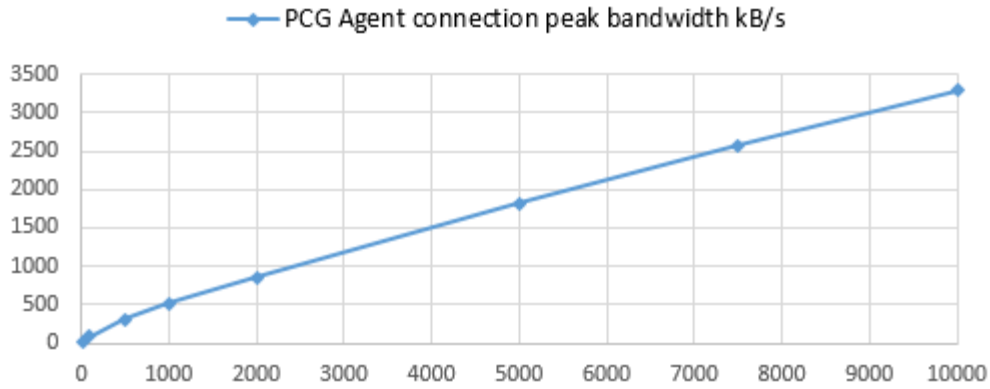
The PBX Agent acts as a CTI gateway between a local CPE PBX and the Rainbow Cloud, handling the following exchanges:

Connection to Rainbow Cloud

involving phonebook synchronization, as well as CTI monitoring initiation and phone state synchronization. This exchange takes place when the PBX Agent is started as well as when reconnecting following a broken connection due to an issue along the path.

The connection incurs a substantial burst of messages lasting anywhere from less than 1 second to 10 seconds depending on the number of PBX subscribers subject to Rainbow monitoring, whose rate is characterized in the following table:

# subscribers	20	100	500	1000	2000	5000	7500	10000
Connection Peak bandwidth kB/s	26	80	304	510	850	1818	2696	3294

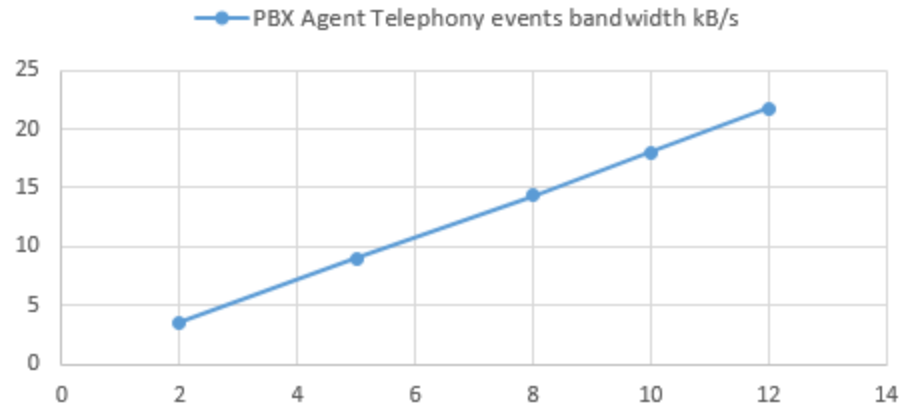


Telephony events

Once connected, CSTA telephony events flowing in both directions in 1pcc or 3pcc contexts. The level of traffic here is directly proportional to call activity initiated on PBX side from endpoints or external inbound calls or initiated from a Rainbow client in 3pcc mode. As such, this bandwidth is

much lower than connection bursts, characterized as a function of the overall number of calls per second in busy hour rather than number of subscribers alone.

Calls / second	2	5	8	10	12
BHCA	7200	18000	28800	36000	43200
Bandwidth kB/s	3.5	9	14.4	18	21.7



Notes:

- The figures above are given in kilobytes per second, not kbps.
- The peak bandwidth is based on measurements conducted in real conditions from an enterprise site benefiting from an enterprise-grade WAN connection to the Rainbow Cloud with low latency, high bandwidth capacity. Actual WAN connections and/or local concurrent traffic may restrict the bandwidth available to the PBX Agent connection without consequence except for a longer time to connect.
- The peak bandwidth indicated here only accounts for the burst mostly induced by the phonebook synchronization. This burst is followed by a lower rate of CSTA exchanges to initiate the monitoring of devices, which for large PBXs can take several minutes for the connection to be fully operationnal.
- The simultaneous connection or reconnection of several PBX Agents cumulates the overall bandwidth requirements indicated here, possibly up to the common link capacity. If only one or some PBX Agents are to reconnect at the same time while others remain connected, the peak bandwidth requirement must be cumulated with the current telephony events traffic generated by already connected PBX Agents.
- The figures may be linearly interpolated to the actual number of subscribers.

7 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 **Error! Reference source not found.** and 5.

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.

8 Configuration of border elements in enterprise

8.1 DNS, Firewall, Proxy configuration

To allow Rainbow to operate properly, border elements like DNS, HTTP Proxy or Firewall must be configured to allow resolving accessing domains and protocols listed in the table of chapter **Error! Reference source not found.** and 5.

8.2 HTTP Proxy and DPI

If an HTTP Proxy is configured on the device where Rainbow applications are running, Rainbow Web and Desktop clients 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). Mobile application use the proxy for signaling, but media has to be enabled directly or by fallback on mobile data network (see chapter **Error! Reference source not found.**).

In case Deep Packet Inspection is implemented by proxy/firewall, some exceptions have to be configured for Rainbow, according to the Note of 4.1:

> *The connection between Rainbow clients or WebRTC Gateway and the TURN Servers, 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 Servers, so the DPI policy allows this legitimate Rainbow TURN connections without attempt for intermediate decryption neither HTTP inspection.*

DPI exception can be applied on *.openrainbow.com. Note however that the WebRTC Gateway requires version 1.75 to send TURN FQDNs rather than IP addresses, in HTTP CONNECT. WRG must therefore be at minimum in version 1.75 for complying with DPI that only accept configuring exceptions based on FQDNs.

If must also be verified that the proxy is properly dimensioned, so it supports the number of simultaneous ports inferred by Rainbow usage:

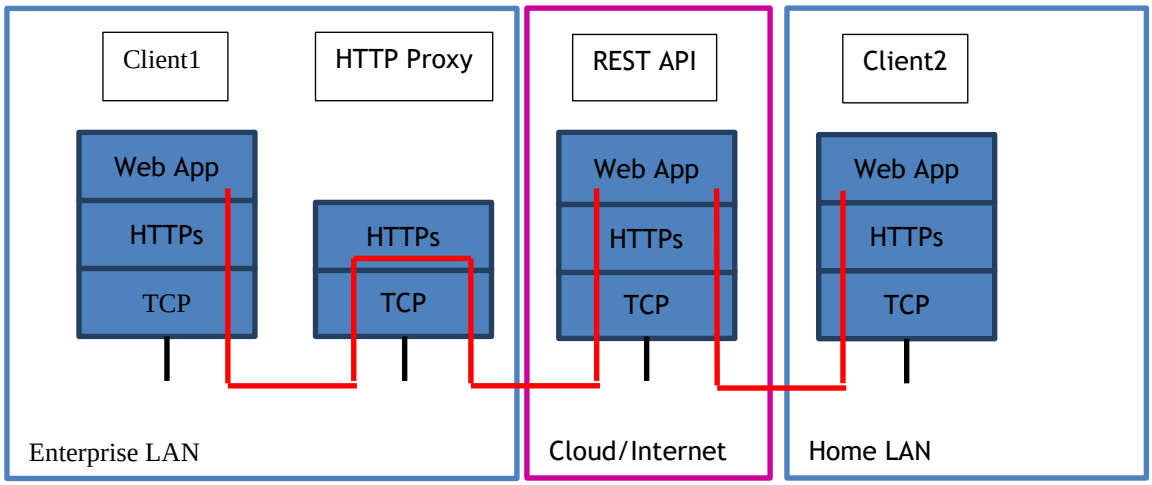
- For signaling and APIs, each Rainbow client uses a permanent WSS towards Rainbow servers, and can perform parallel HTTPS requests for resources and APIs calls
- In addition to this, each WebRTC call via the proxy (ie involving a remote user) consumes additional ports for audio and video and sharing (one each in conference). Calls can be generated from Rainbow clients, and from the WebRTC gateway (possibly several hundred of simultaneous audio calls, refer to WebRTC Gateway capacity in TBE067 available on business partner web site)

Note finally that HTTP version 1.1 must be supported and used by the proxy, typically for the WebRTC Gateway to properly allow connecting to TURN server when a proxy is used.

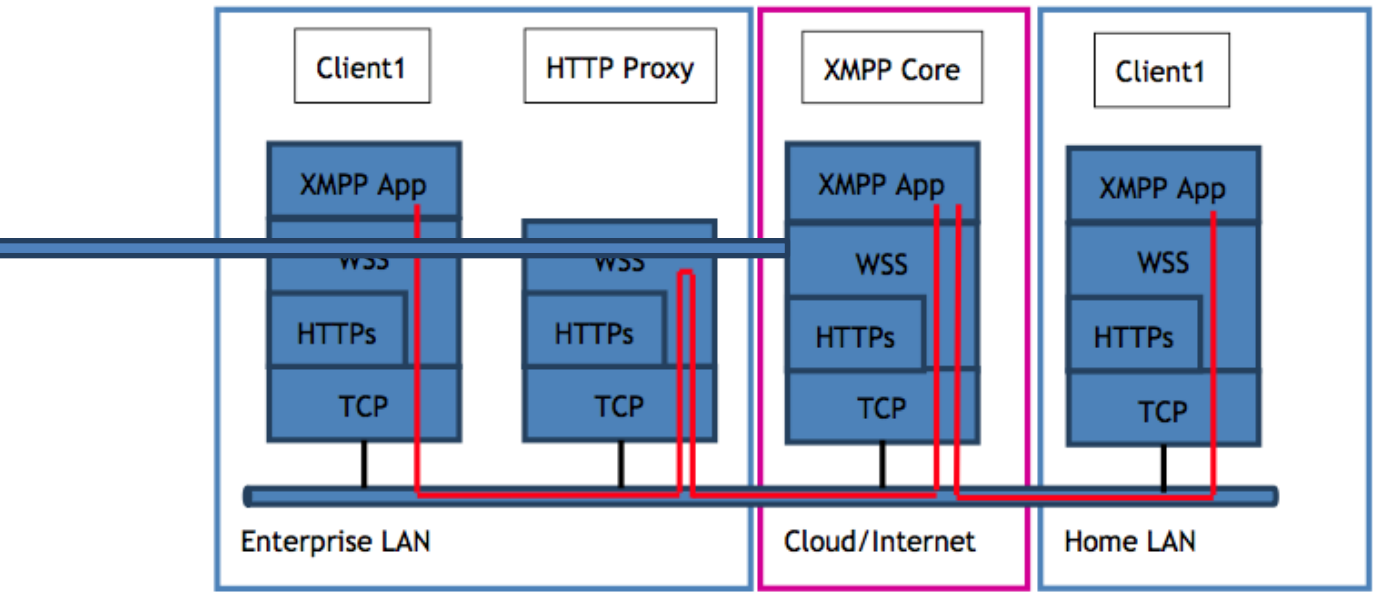
9 Annexes: Detailed call-flow of HTTPS/REST, XMPP and ICE connections

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):

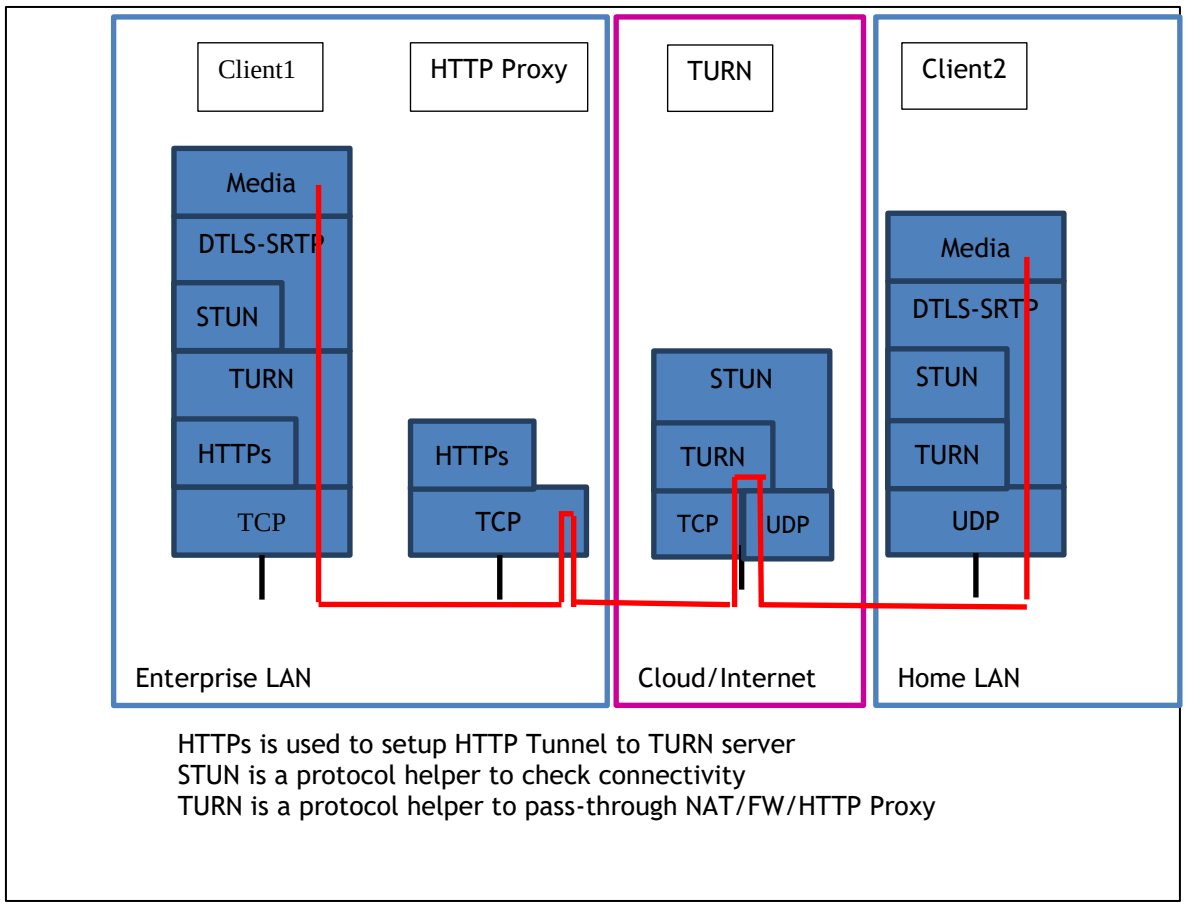


Network layers (XMPP):



HTTPs is used to setup WSS protocol (RFC6455)

Network layers (Media):



Call Flow:

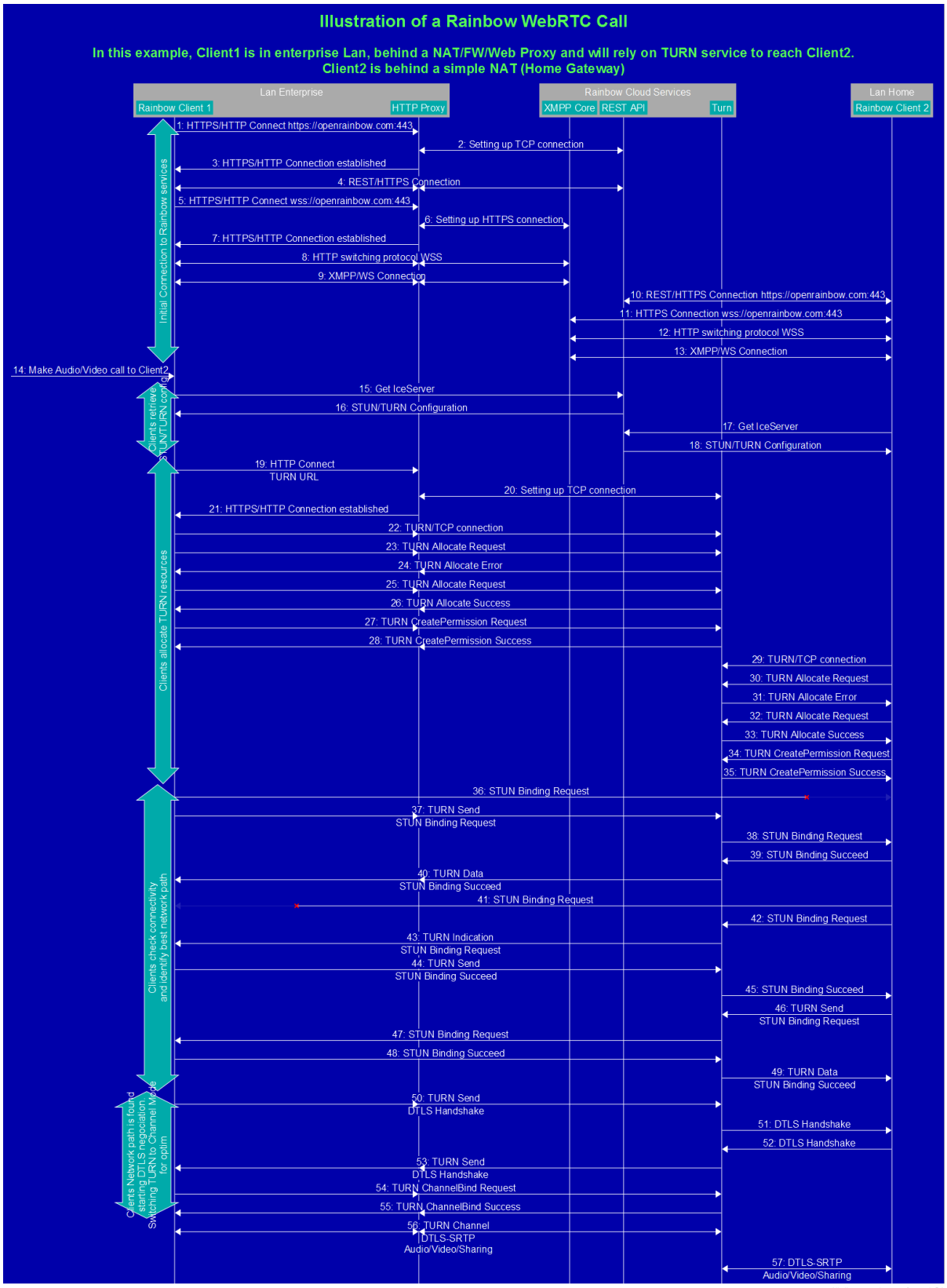


Illustration of a Rainbow WebRTC Call
In this example, Client1 is in enterprise Lan, behind a NAT/FW/Web Proxy and will rely on TURN service to reach Client2.
Client2 is behind a simple NAT (Home Gateway)
Lan Enterprise
Rainbow Client 1
HTTP Proxy
Rainbow Cloud Services
XMPP Core
REST API
Turn
Lan Home
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
http://msc-generator.sourceforge.net v5.4

- 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