

# Exploración de datos y estadísticas de WebRTC en WxCC Agent Desktop

## Contenido

---

[Introducción](#)

[Prerequisites](#)

[Requirements](#)

[Componentes Utilizados](#)

[Overview](#)

[Cromado](#)

---

## Introducción

En este documento se describen los datos y estadísticas de llamadas WebRTC disponibles mediante las herramientas de explorador integradas.

## Prerequisites

### Requirements

Cisco recomienda que tenga conocimientos y habilidades sobre la solución Webex Contact Center (WxCC), los navegadores Chrome, Edge, Firefox y el proyecto WebRTC.

### Componentes Utilizados

Este documento no tiene restricciones específicas en cuanto a versiones de software y de hardware.

La información que contiene este documento se creó a partir de los dispositivos en un ambiente de laboratorio específico. Todos los dispositivos que se utilizan en este documento se pusieron en funcionamiento con una configuración verificada (predeterminada). Si tiene una red en vivo, asegúrese de entender el posible impacto de cualquier comando.

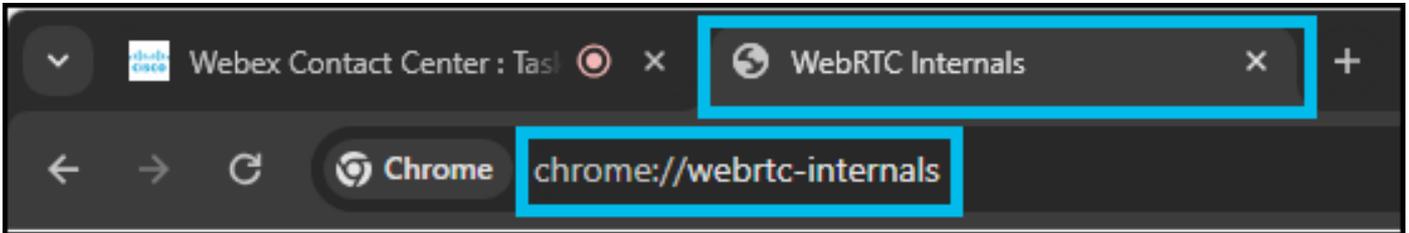
## Overview

La mayoría de los exploradores Web que admiten WebRTC disponen de herramientas WebRTC integradas que pueden resultar útiles para solucionar problemas relacionados con WebRTC.

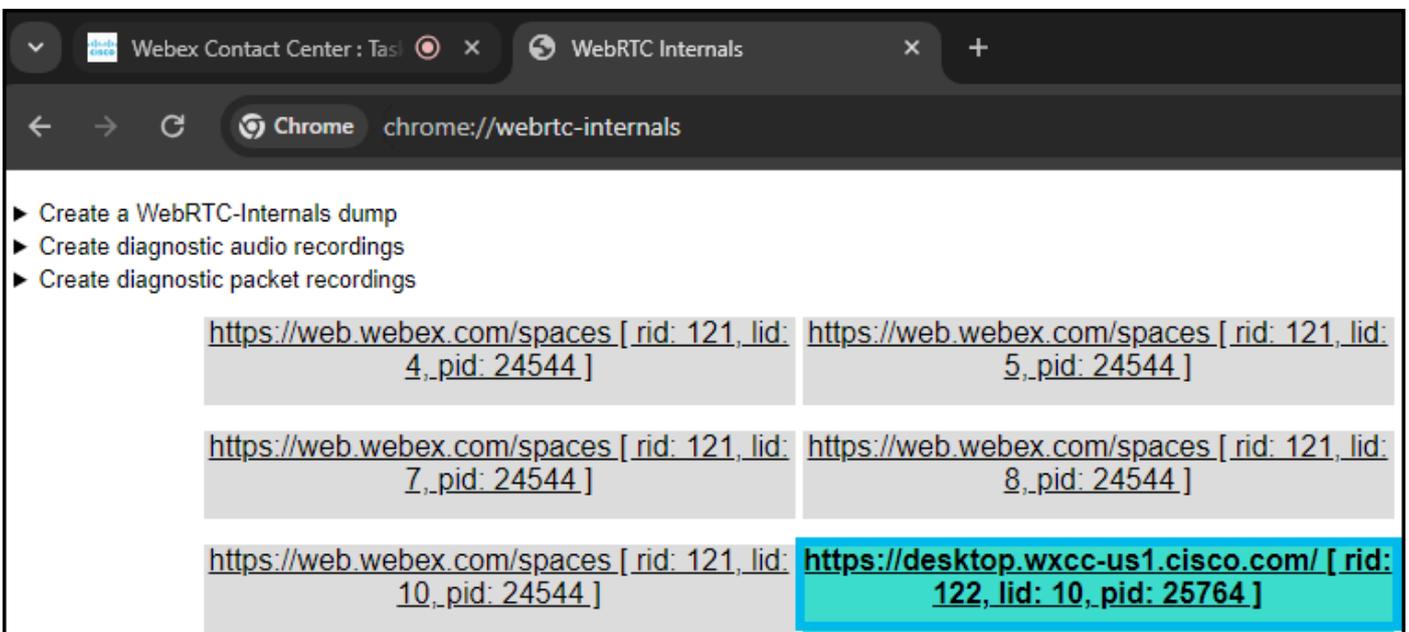
# Cromado

Para acceder a los datos y estadísticas de llamadas de WebRTC, se debe abrir una nueva pestaña en el navegador web Google Chrome mientras hay una llamada activa y se debe utilizar la siguiente URL:

chrome://webrtc-internals.



En la ficha WebRTC-internals debe seleccionarse la sesión WebRTC activa (desktop.wxcc-us1.cisco.com en su nombre; URL dependerá de la región WxCC).



Se puede ver la siguiente información:

-En el lado izquierdo (1), hay disponible un registro de las actividades de la API PeerConnection. RTCPeerConnection es la interfaz central de la API WebRTC. Representa la conexión entre el par local y remoto y proporciona todas las funciones y eventos necesarios para establecer la conexión (se pueden encontrar más detalles en <https://webrtc.org/getting-started/overview>). Estos seguimientos (1) muestran las interacciones con el objeto RTCPeerConnection y los parámetros utilizados.

-En el lado derecho (2), están disponibles las métricas recopiladas mediante la API getStats sobre la conexión WebRTC actual.

-En la parte inferior (3), están disponibles los gráficos generados mediante la API getStats (2) sobre la conexión WebRTC actual.

https://desktop.wxcc-us1.cisco.com/, { iceServers: [], iceTransportPolicy: all, bundlePolicy: max-compat, rtcpMuxPolicy: require, iceCandidatePoolSize: 0 },

ICE connection state: new => completed  
 Connection state: new => connected  
 Signaling state: new => stable  
 ICE Candidate pair: 10.24.139.247:49870 <=>23.89.1.162:22744

▶ ICE candidate grid

Time	Event
4/17/2024, 10:13:52 AM	▶ setRemoteDescription (type: "offer", 2 sections)
4/17/2024, 10:13:52 AM	▶ setLocalDescription (type: "answer", 2 sections)
4/17/2024, 10:13:52 AM	▶ signalingstatechange
4/17/2024, 10:13:52 AM	▶ iceconnectionstatechange
4/17/2024, 10:13:52 AM	▶ connectionstatechange

Filter statistics by type including

- ▶ media-playout (kind=audio, id=AP)
- ▶ certificate (id=CFE0AAE88DCEFAA00E634B6C16C3D0AD.C37B2BA7EE)
- ▶ codec (mimeType=audio/PCMU, payloadType=0, id=CIT01\_0)
- ▶ codec (mimeType=audio/PCMU, payloadType=0, id=COT01\_0)
- ▶ candidate-pair (state=succeeded, id=CPyDISaDsy\_T5aV8Gj8)
- ▶ local-candidate (candidateType=host, id=1toYJvKsp)
- ▶ local-candidate (candidateType=host, tcpType=active, id=I9OkJn/oQ)
- ▶ local-candidate (candidateType=host, tcpType=active, id=ILUW6L1VJ)
- ▶ inbound-rtcp (kind=audio, mid=0, ssrc=577774253, [codec]=PCMU (0), id=IT01A57774253)
- ▶ remote-candidate (candidateType=host, id=IT5aV8Gj8)
- ▶ local-candidate (candidateType=host, tcpType=active, id=IcCmyDGdn)
- ▶ local-candidate (candidateType=host, id=id4Nah5bb)
- ▶ local-candidate (candidateType=host, tcpType=active, id=Ih75e1cck)
- ▶ local-candidate (candidateType=host, id=IhDILG2ss)
- ▶ local-candidate (candidateType=host, id=IyDISaDsy)
- ▶ outbound-rtcp (kind=audio, mid=0, ssrc=3932673161, [codec]=PCMU (0), id=OT01A3932673161)
- ▶ peer-connection (id=P)
- ▶ remote-inbound-rtcp (kind=audio, ssrc=3932673161, id=RIA3932673161)
- ▶ remote-outbound-rtcp (kind=audio, ssrc=577774253, id=ROA57774253)
- ▶ media-source (kind=audio, id=SA1)
- ▶ transport (iceState=connected, dtlsState=connected, id=T01)

Filter statistics graphs by type including

- ▶ Stats graphs for media-playout (kind=audio, id=AP)
- ▶ Stats graphs for candidate-pair (state=succeeded, id=CPyDISaDsy\_T5aV8Gj8)
- ▶ Stats graphs for inbound-rtcp (kind=audio, mid=0, ssrc=577774253, [codec]=PCMU (0), id=IT01A57774253)
- ▶ Stats graphs for outbound-rtcp (kind=audio, mid=0, ssrc=3932673161, [codec]=PCMU (0), id=OT01A3932673161)
- ▶ Stats graphs for peer-connection (id=P)
- ▶ Stats graphs for remote-inbound-rtcp (kind=audio, ssrc=3932673161, id=RIA3932673161)
- ▶ Stats graphs for remote-outbound-rtcp (kind=audio, ssrc=577774253, id=ROA57774253)
- ▶ Stats graphs for media-source (kind=audio, id=SA1)
- ▶ Stats graphs for transport (iceState=connected, dtlsState=connected, id=T01)

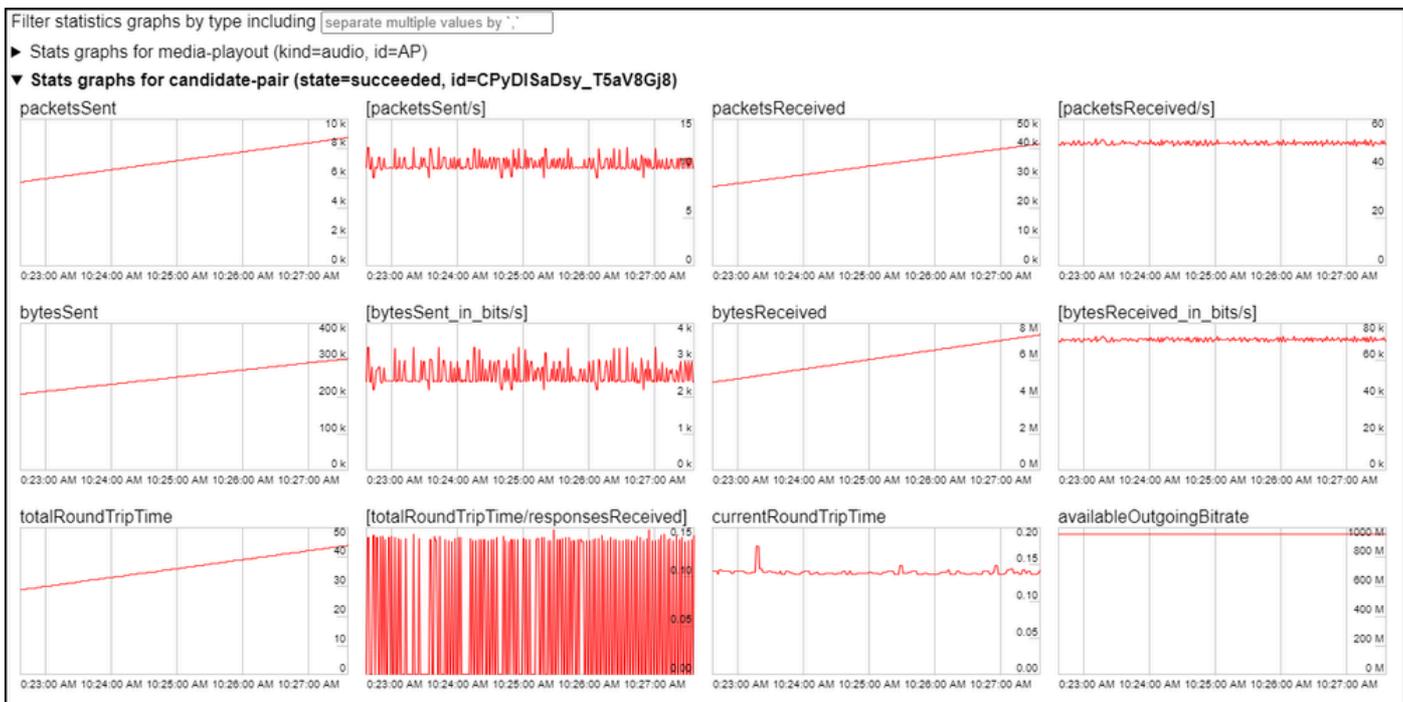
Las IP y los puertos locales y remotos que se utilizan para esta conexión WebRTC concreta se pueden ver en los seguimientos de las llamadas API de PeerConnection.

ICE connection state: new => completed  
 Connection state: new => connected  
 Signaling state: new => stable  
**ICE Candidate pair: 10.24.139.247:49870 <=>23.89.1.162:22744**

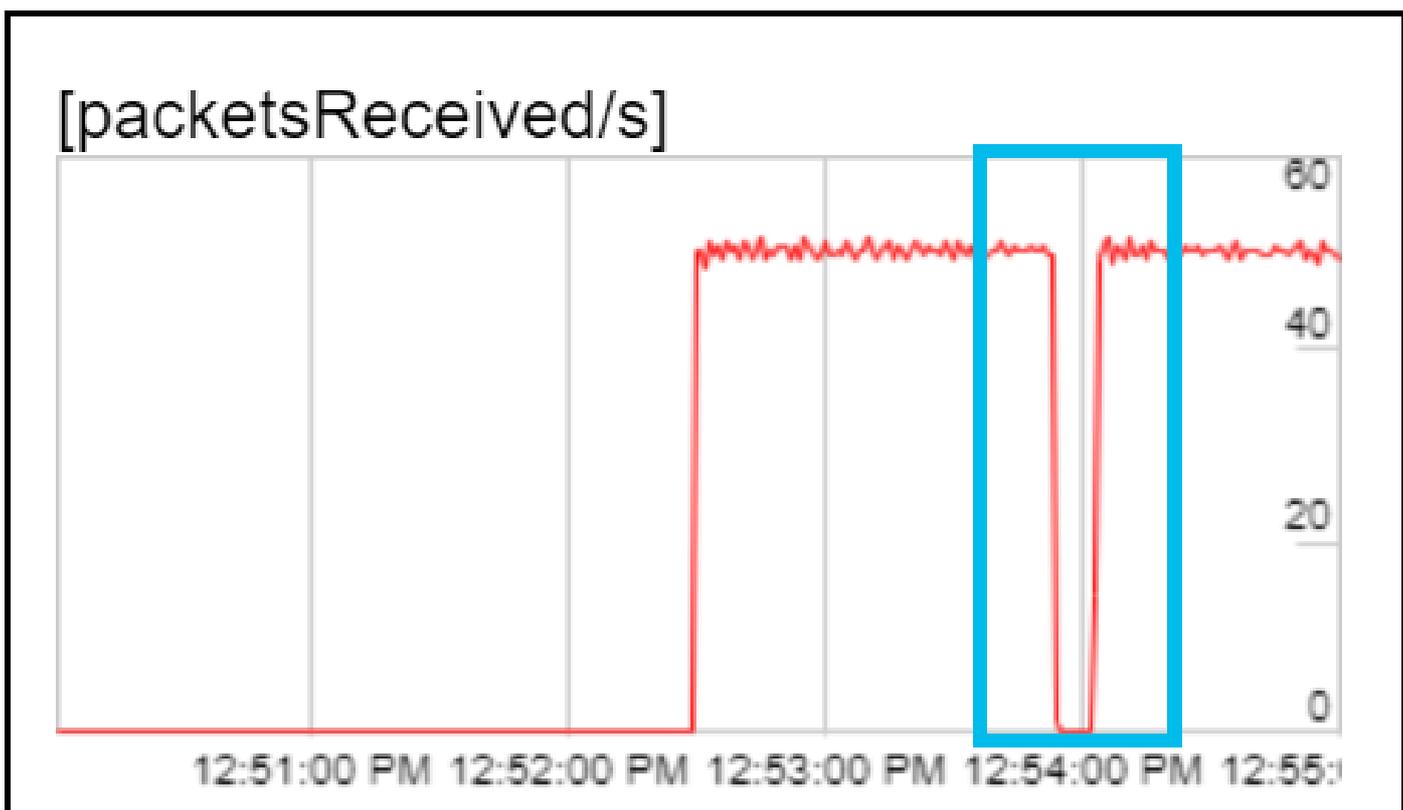
▶ ICE candidate grid

Time	Event
4/17/2024, 10:13:52 AM	▶ setRemoteDescription (type: "offer", 2 sections)
4/17/2024, 10:13:52 AM	▶ setLocalDescription (type: "answer", 2 sections)
4/17/2024, 10:13:52 AM	▶ signalingstatechange
4/17/2024, 10:13:52 AM	▶ iceconnectionstatechange
4/17/2024, 10:13:52 AM	▶ connectionstatechange

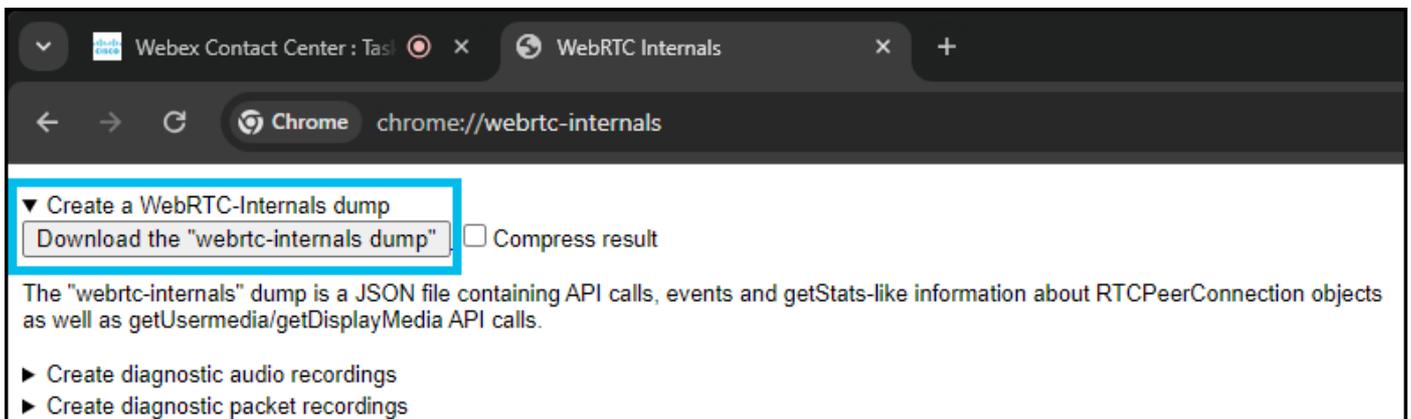
Diversas estadísticas de llamadas en directo (es decir, paquetes enviados, paquetes recibidos, latencia, paquetes descartados, etc.). En la parte inferior (texto en negrita) se pueden ver gráficos relacionados.



A modo de ejemplo, la falta de paquetes entrantes provocaría la falta de audio:

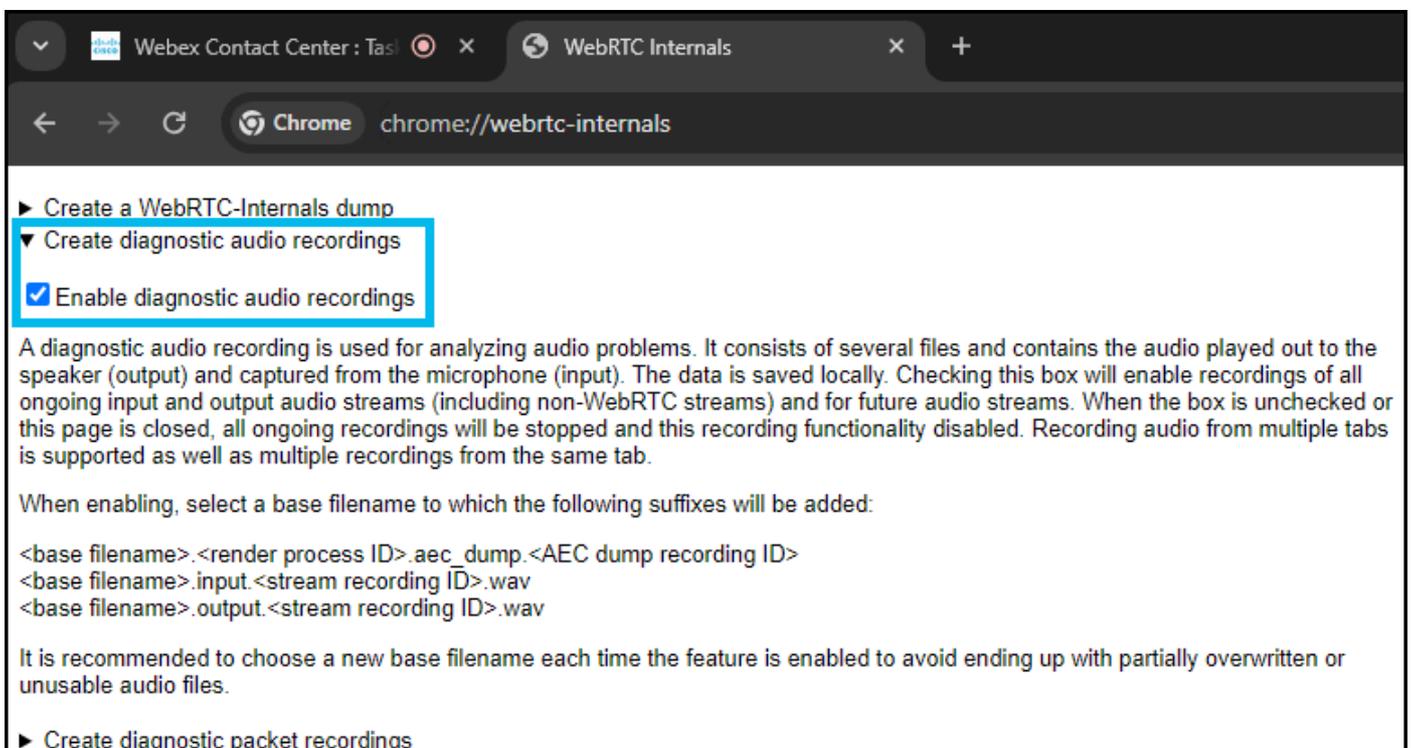


Para recuperar el volcado de registro de WebRTC, haga clic en 'Crear un volcado interno de WebRTC' y en 'Descargar la opción "volcado interno de WebRTC".'



Nota: El archivo de volcado de registro de WebRTC se puede analizar utilizando herramientas específicas, es decir. <https://fippo.github.io/webrtc-dump-importer/>

El audio WebRTC también se puede capturar con fines de diagnóstico.



## Acerca de esta traducción

Cisco ha traducido este documento combinando la traducción automática y los recursos humanos a fin de ofrecer a nuestros usuarios en todo el mundo contenido en su propio idioma.

Tenga en cuenta que incluso la mejor traducción automática podría no ser tan precisa como la proporcionada por un traductor profesional.

Cisco Systems, Inc. no asume ninguna responsabilidad por la precisión de estas traducciones y recomienda remitirse siempre al documento original escrito en inglés (insertar vínculo URL).