

Expected Behavior When Using the maxPeerVideoStreams Parameter in CMS Clusters

Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Components Used](#)

[What is the maxPeerVideoStreams parameter and when does it come into effect?](#)

[Example deployment and scenarios](#)

[1. maxPeerVideoStreams set to 4 with loadBalancing enabled](#)

[2. maxPeerVideoStreams set to 4 with loadBalancing disabled](#)

[3. maxPeerVideoStreams set to 9 with loadBalancing enabled](#)

[4. maxPeerVideoStreams set to 9 with loadBalancing disabled](#)

[Troubleshoot](#)

[Related Information](#)

Introduction

This document describes the expected behavior of the parameter maxPeerVideoStreams when it is used in a Cisco Meeting Server (CMS) cluster.

This parameter is mentioned in the [Administrator Quick Reference Guide](#).

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- Cisco Meeting Server Call Bridge component (and clustering of it)
- Cisco Meeting Server API configuration

Components Used

The information in this document is based on these software and hardware versions:

- CMS 2.9.x

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

What is the `maxPeerVideoStreams` parameter and when does it come into effect?

The `maxPeerVideoStreams` parameter was first introduced in CMS version 2.3. This parameter governs how many participant video streams can a CMS server send over a distributed call to another CMS server. It needs to be set on each CMS server separately. The `maxPeerVideoStreams` parameter is effective for a large, distributed, conference when there are more than 4 participants on each CallBridge.

Note: The `maxPeerVideoStreams` is only relevant in a CMS Cluster of two or more servers, it is not relevant with a single CMS server.

If `maxPeerVideoStreams` is not set, the default behavior of CMS is to send a maximum of 4 video streams over a distributed call to the other CMS server, this was the behavior prior to CMS 2.3. With CMS 2.3 and higher, it is now possible to change that behavior and configure CMS to send a maximum of 9 video streams over the distributed call instead of only 4.

This importance of this parameter becomes clearer with big conferences, hosting a large number of participants, and using an **AllEqual** layout, which allows for a maximum of 25 panes to be displayed on a single participant's screen. In this case, if a conference is distributed over two CMS servers (for example CMS1 and CMS2), and more than 4 participants are hosted on each CMS server for this conference (5 or more), the participants hosted on CMS1 are only able to see the video from a maximum of 4 participants from the remote participants which are hosted on CMS2, as addition to the video from all other local participants hosted on their local CMS (CMS1) server host, even if CMS2 has 8 active participants currently. The same applies for participants hosted on CMS2 - they are only able to see the video from a maximum of 4 participants from the remote participants hosted on CMS1 and the video of the other participants hosted on the same CMS2, even if CMS1 has 10 active participants.

Note: The `maxPeerVideoStreams` is still a beta (preview) feature.

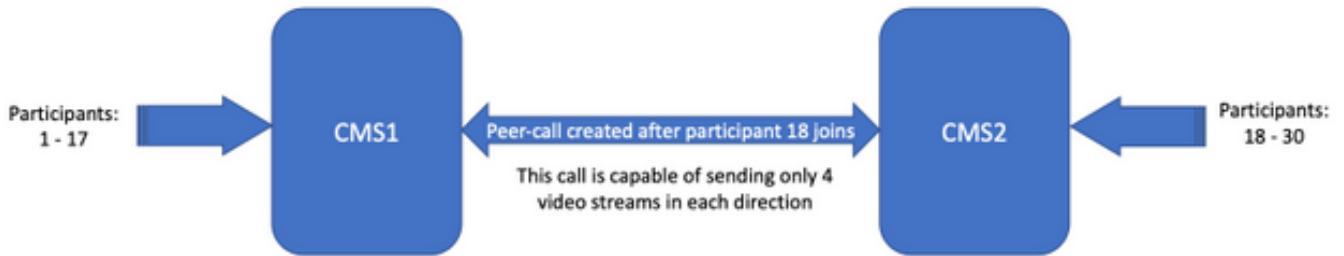
Example deployment and scenarios

The information in this document is based on this example deployment:

- **CMS Cluster** of two servers, CMS1 and CMS2
- The **Loadlimit** configured on those servers allows for 17 calls, after that call distribution starts
- **CUCM Route Group** for the CMS servers is configured with **circular** distribution
- **AllEqual** layout is used, or 5x5, this is to allow the maximum possible participants panes, which is 25
- 30 participants are joining **space1**, which has a **priority** (for loadbalancing) on CMS1

1. `maxPeerVideoStreams` set to 4 with `loadBalancing` enabled

- Since **loadbalancing** is enabled and the priority of **space1** is on CMS1, the first 17 participants join on CMS1, until it reaches its full capacity. The upcoming participant 18 joins on CMS2 and a distributed call is created

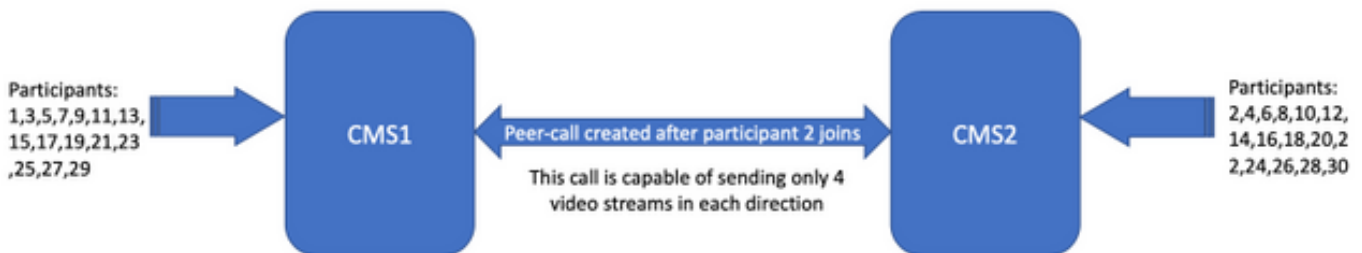


maxPeerVideoStreams set to 4 with loadbalancing enabled

- There are 17 participants on CMS1 (1 – 17), and 13 participants on CMS2 (18 – 30)
- Participants 1 – 17 see the other 16 local participants from CMS1, in addition to only 4 participants from CMS2, a total of 20 participants are displayed on the screens of participants 1 – 17
- Participants 18 – 30 see the other 12 local participants from CMS2, in addition to only 4 participants from CMS1, a total of 16 participants are displayed on the screens of participants 18 – 30
- In summary: CMS1-hosted participants see 20 participants, CMS2-hosted participants see 16 participants on their screens

2. maxPeerVideoStreams set to 4 with loadBalancing disabled

- Since **loadbalancing** is **not enabled**, participants join the conference on both CMS servers starting from the second call. This is because the **CUCM Route Group** is set to **circular**, which means that calls are sent to both CMS servers sequentially. Call 1 is sent to CMS1, call 2 is sent to CMS2, call 3 is sent to CMS1, call 4 is sent to CMS2
- This means that it is expected to find 15 participants hosted on each CallBridge - there are 15 participants on CMS1 and 15 participants on CMS2



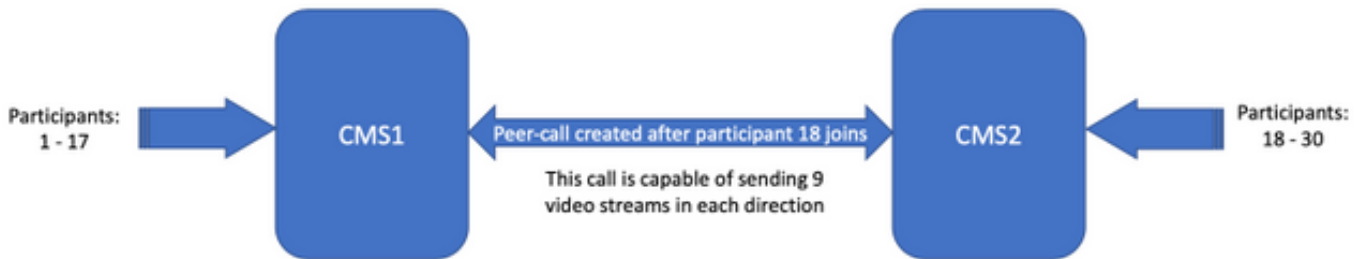
maxPeerVideoStreams set to 4 with loadbalancing disabled

- Participants on CMS1 see the other 14 local participants from CMS1, in addition to 4 participants from CMS2, a total of 18 participants are displayed on the screens of CMS1 participants
- Participants on CMS2 see the other 14 local participants from CMS2, in addition to 4 participants from CMS1, a total of 18 participants are displayed on the screens of CMS2 participants
- In summary: CMS1 participants and CMS2 participants both see 18 participants on their screens

3. maxPeerVideoStreams set to 9 with loadBalancing enabled

- Since **loadbalancing** is **enabled** and priority of **space1** is on CMS1, participants join on

CMS1 until it reaches its full capacity. The upcoming participant 18 joins on CMS2 and a distributed call is created

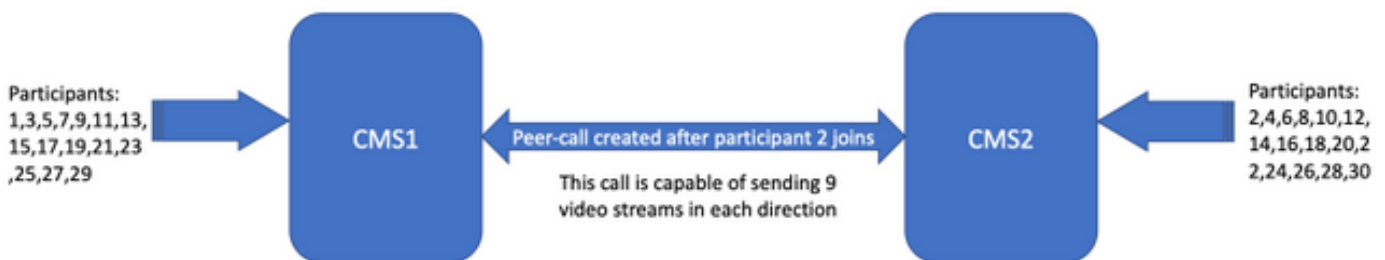


maxPeerVideoStreams set to 9 with loadBalancing enabled

- There are 17 participants on CMS1 (1 – 17), and 13 participants on CMS2 (18 – 30)
- Participants 1 – 17 see the other 16 local participants from CMS1, in addition to 9 participants from CMS2, a total of 25 participants are displayed on the screens of participants 1 – 17
- Participants 18 – 30 see the other 12 local participants from CMS2, in addition to 9 participants from CMS1, a total of 21 participants are displayed on the screens of participants 18 – 30
- In summary: CMS1 participants see 25 participants, CMS2 participants see 21 participants on their screens

4. maxPeerVideoStreams set to 9 with loadBalancing disabled

- Since **loadbalancing** is **not enabled**, participants join the conference on both CMS servers starting from the second call. This is because the **CUCM Route Group** is set to **circular**, which means that calls are sent to both CMS servers sequentially. Call 1 is sent to CMS1, call 2 is sent to CMS2, call 3 is sent to CMS1, call 4 sent to CMS2
- This means that it is expected to find 15 participants hosted on each CallBridge - 15 participants are on CMS1 and 15 participants are on CMS2



maxPeerVideoStreams set to 9 with loadbalancing disabled

- Participants on CMS1 see the other 14 local participants from CMS1, in addition to 9 participants from CMS2, a total of 23 participants are displayed on the screens of CMS1 participants
- Participants on CMS2 see the other 14 local participants from CMS2, in addition to 9 participants from CMS1, a total of 23 participants are displayed on the screens of CMS2 participants
- In summary: CMS1 participants and CMS2 participants both see 23 participants on their screens

Troubleshoot

There is currently no specific troubleshooting information available for this configuration.

You can use the [Collaboration Solutions Analyser](#) tool for log analysis.

Related Information

- [Load Balancing Logic on Cisco Meeting Server](#)
- [CMS configurational documentation](#)
- [CMS API and MMP programming guides](#)