

Cloud Streaming Services

Ops Guide

Product Documentation



Copyright Notice

©2013-2024 Tencent Cloud. All rights reserved.

Copyright in this document is exclusively owned by Tencent Cloud. You must not reproduce, modify, copy or distribute in any way, in whole or in part, the contents of this document without Tencent Cloud's the prior written consent.

Trademark Notice



All trademarks associated with Tencent Cloud and its services are owned by Tencent Cloud Computing (Beijing) Company Limited and its affiliated companies. Trademarks of third parties referred to in this document are owned by their respective proprietors.

Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

Contents

Ops Guide

Video Stuttering

Troubleshooting Push Failure

Troubleshooting Playback Failure

Troubleshooting High Latency

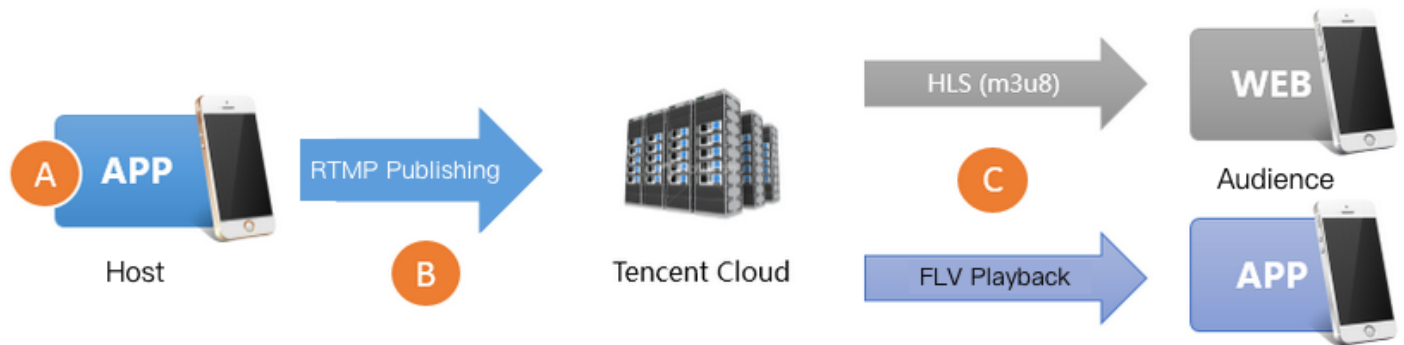
Troubleshooting Poor Quality of Pulled Video

Authorizing CSS to Store Screenshots in a COS Bucket

Ops Guide

Video Stuttering

Last updated : 2022-06-22 16:15:15



There are three main reasons for playback stuttering.

- **Reason 1: low upstream frame rate**

Low upstream frame rate may be a result of poor performance of the host's mobile phone or the running of CPU-intensive apps in the background. Normally, to ensure smooth playback, the upstream frame rate must be 15 FPS or higher. Frame rate lower than 10 FPS is deemed **too low**, which will cause **all viewers** to experience stuttering. However, when the host's video image changes little, for example, when the host is showing a static image or PowerPoint, low frame rate will not cause stuttering.

- **Reason 2: upstream congestion**

The host's mobile phone keeps pushing audio and video data during live streaming. If the phone's upstream bandwidth is too low, the data waiting to be pushed will accumulate, causing congestion in data transfer and stuttering playback experience for **all viewers**.

Even though **ISPs in the Chinese mainland** offer broadband packages with downstream bandwidth as high as 10 Mbps or 20 Mbps, or even 100 Mbps or 200 Mbps, upstream bandwidth remains small. In many small cities, upstream bandwidth is up to 512 Kbps, which means that a maximum of 64 KB data can be uploaded per second.

Wi-Fi uses the carrier-sense multiple access and collision avoidance (CSMA/CA) strategy specified in IEEE 802.11. To put it simply, a Wi-Fi hotspot can communicate with only one phone at a time, and other phones must query if communication is possible before initiating a connection to a hotspot. Therefore, the more people using a Wi-Fi hotspot, the slower the connection is. Furthermore, Wi-Fi signals are weakened significantly when passing through walls. Among average Chinese households, few take this into consideration when designing and decorating their

houses, and hosts probably pay little attention to how many walls they are apart from their routers when they stream at home.

- **Reason 3: poor downstream connection**

Poor downstream connection means slow download speed or network instability for viewers. A bitrate of 2 Mbps in live streaming means 2 Mb of data needs to be downloaded per second. Viewers will experience stuttering if their network bandwidth is low, but those with sufficient bandwidth will not.

Checking SDK Performance Metrics

The MLVB SDK has a feedback mechanism that reports different performance metrics every 2 seconds. If you use the MLVB SDK for push, you can register a [V2TXLivePusherObserver](#) listener and get the statistics in the

`onStatisticsUpdate` callback. The table below lists the metrics included in

`V2TXLivePusherStatistics` and their meanings.

Metric	Description
appCpu	CPU usage of the application (%)
systemCpu	CPU usage of the system (%)
width	Video width
height	Video height
fps	Frame rate (FPS)
audioBitrate	Audio bitrate in Kbps
videoBitrate	Video bitrate in Kbps

Fixing Low Frame Rate

1. How to know whether the frame rate is too low

The `V2TXLivePusherStatistics.fps` field in the [onStatisticsUpdate](#) callback of `V2TXLivePusherObserver` indicates the frame rate for push. Normally, the frame rate must be 15 FPS or higher to ensure smooth playback. Viewers usually experience obvious stuttering when the push frame rate is lower than 10 FPS.

2. How to fix the problem

- **2.1 Monitoring `appCpu` and `systemCpu`**

You can learn about the **CPU usage of the app and system** from the `V2TXLivePusherStatistics.appCpu` and `V2TXLivePusherStatistics.systemCpu` fields in the `onStatisticsUpdate` callback of

`V2TXLivePusherObserver`. If the system CPU usage exceeds 80%, both video capturing and encoding may be affected; if it reaches 100%, it is difficult to even ensure smooth push, let alone playback experience for viewers.

- **2.2 Identifying CPU consumers**

The MLVB SDK is not the only CPU consumer in a live streaming application. Leaving on-screen comments, sending hearts, and text messaging all consume CPU. To monitor the CPU usage of the MLVB SDK only, you may use its demo.

- **2.3 Choosing an appropriate resolution**

High resolution does not necessarily result in high video quality. To begin with, high resolution translates into improved video quality only when the bitrate is also high. Low bitrate and high resolution usually produce lower video quality than high bitrate and low resolution. In addition, viewers may be able to sense obvious differences between a resolution of 1280 x 720 px and 960 x 540 px when watching videos full screen on PCs, but not on mobile phones, whose average screen size is only around 5 inches. High resolution increases the CPU usage of the SDK significantly. Therefore, you are advised to set video quality to **HD** using the `setVideoQuality` API in

`V2TXLivePusher` of the MLVB SDK. You may not get the high video quality expected by setting the resolution too high.

Fixing Upstream Congestion

Statistics show that upstream congestion at the host end is responsible for over 80% of playback stuttering in live streaming.

1. Informing hosts of poor network conditions

In scenarios where video quality is important, you are advised to inform the hosts of bad network conditions through UI notifications. For example, you may send this notification to hosts: **Bad network conditions. Please move closer to your router or make sure that your Wi-Fi signal does not have to pass through walls.**

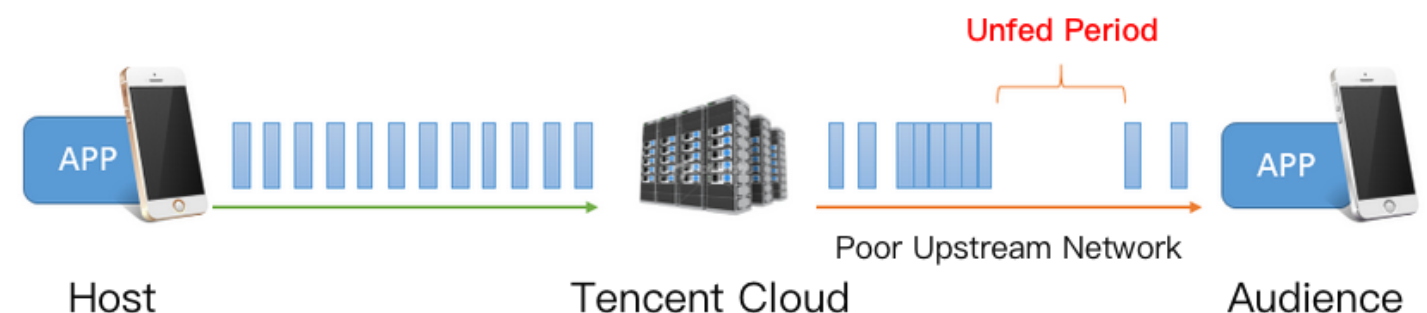
For more information on how to do this, please see **Mobile Live Video Broadcasting > Basic Features > Camera Push > Event Handling**. You are advised to remind hosts to check their network conditions if your application receives the `V2TXLIVE_WARNING_NETWORK_BUSY` event multiple times within a short period of time. Hosts are often unable to notice upstream congestion until reminded by viewers or an application message.

2. Using the recommended encoding settings

We recommend the following encoding settings. You can use the `setVideoQuality` API in `V2TXLivePusher` to set the parameters.

Scenario	resolution	resolutionMode
Live showroom	<ul style="list-style-type: none"> V2TXLiveVideoResolution960x540 V2TXLiveVideoResolution1280x720 	Landscape or portrait
Live game streaming	V2TXLiveVideoResolution1280x720	Landscape or portrait
Mic connect (primary image)	V2TXLiveVideoResolution640x360	Landscape or portrait
Mic connect (small image)	V2TXLiveVideoResolution480x360	Landscape or portrait
Blue-ray live streaming	V2TXLiveVideoResolution1920x1080	Landscape or portrait

Fixing Player-End Issues



1. Stuttering and latency

As shown in the figure above, both downstream network fluctuations and insufficient downstream bandwidth can result in **unfed periods** (during which the application cannot get any audio/video data for playback) in the playback process. To avoid playback stuttering, the application needs to cache video data enough to cover the unfed periods. However, caching too much data causes a new problem: **high latency**, which is undesirable for interactive scenarios. The latency could **build up** over time if not fixed, meaning that it increases as the playback continues. The capability to fix latency is a key performance indicator for players. **Latency and playback smoothness are like the two ends of a scale.** To ensure low latency, you may have to compromise network stability, which causes playback stuttering, and to ensure smooth playback, you must deal with high latency. A typical example of the latter is the introduction of a 20-30 second delay in the playback of HLS (M3U8) streams to ensure smooth playback experience.

2. How to fix the problem

To allow you to deliver superior playback experience without having to learn QoS control, we have developed an automatic latency control technology, which we optimized from version to version, and launched three latency control schemes. You can use the `setCacheParams` API in `V2TXLivePlayer` to select one of the schemes.

- **Auto mode:** use this mode if you are not sure what scenarios you will deal with.

Note :

In this mode, the player adjusts latency automatically based on network conditions to minimize the latency between hosts and viewers and ensure host-viewer interaction quality while delivering smooth playback experience. By default, the player adjusts latency in the range of 1-5 seconds, which you can modify using the `setCacheParams` API.

- **Speedy mode:** suitable for **live showroom** and other interactive scenarios that require low latency.

Note :

You can switch to this mode by **setting both `minTime` and `maxTime` to 1 second**. The auto and speedy modes differ only in terms of `maxTime`, whose value is generally lower in the speedy mode. This flexibility is made possible by the SDK's automatic latency control technology, which can automatically adjust latency without causing stuttering. `maxTime` is associated with adjustment speed. The higher the `maxTime` value, the more conservative the adjustment is, and the less likely stuttering will occur.

- **Smooth mode:** suitable for **live game streaming** and other high-bitrate and high-definition scenarios.

Note :

- In this mode, the player adopts a strategy similar to the caching policy of Adobe Flash Player. When stuttering occurs, the player switches to the loading mode, and when enough data is cached, it returns to the playing mode, until the network connection fluctuates again. By default, 5 seconds of video data is cached, which you can change using the `setCacheParams` API.
- This seemingly simple mode reduces playback stuttering by moderately increasing latency and is therefore the more reliable option for scenarios that do not require low latency.

Troubleshooting Push Failure

Last updated : 2022-03-21 11:47:35

If you follow the steps in [Best Practice - Live Push](#) but still find that the push is not successful, please troubleshoot common push issues as instructed in this document.

1. Check whether a CNAME record that points to a Tencent Cloud address has been configured for your domain name.

Your push can succeed only if your domain name has a CNAME record that points to a Tencent Cloud address. You can check whether an added push domain name has a CNAME record in the **CNAME** column of [Domain Management](#). You will see a green check if the CNAME record is configured, as shown below:

<input type="checkbox"/> Domain Name	CNAME ⓘ	Type ▾	Scenario	Region	Status	Added Time	Operation
<input type="checkbox"/> [Domain Name]	<input checked="" type="checkbox"/>	Push Domain	LVB	Global	Enabled	2018-07-20 21:32:48	Manage Disable Delete

If your domain name does not have a CNAME record, you can [configure one](#).

2. Check whether the network is normal.

RTMP push uses the **1935** port by default. If the port is not enabled on the firewall of the network for testing, you will be unable to connect to the server. You can check whether this issue caused the push failure by switching to another network (a 4G network for example).

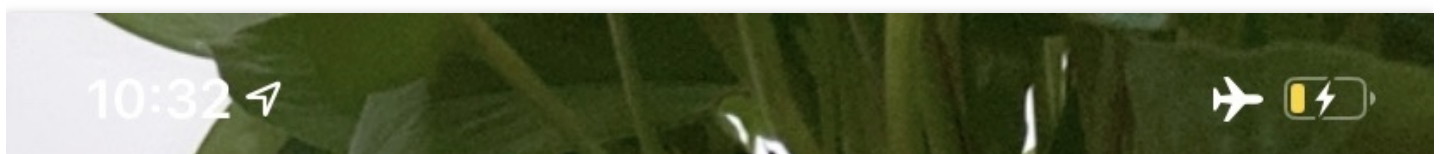
3. Check whether the validity period of the push URL is too short.

To prevent the traffic from being stolen, some clients may set `txTime` (the validity period of the push URL) to a very short period, such as 5 minutes from the current time. This is unnecessary as the `txSecret` signature guarantees security. If the validity period is too short, the push URL may expire after a live stream is interrupted due to network disconnection, and consequently the push cannot be resumed.

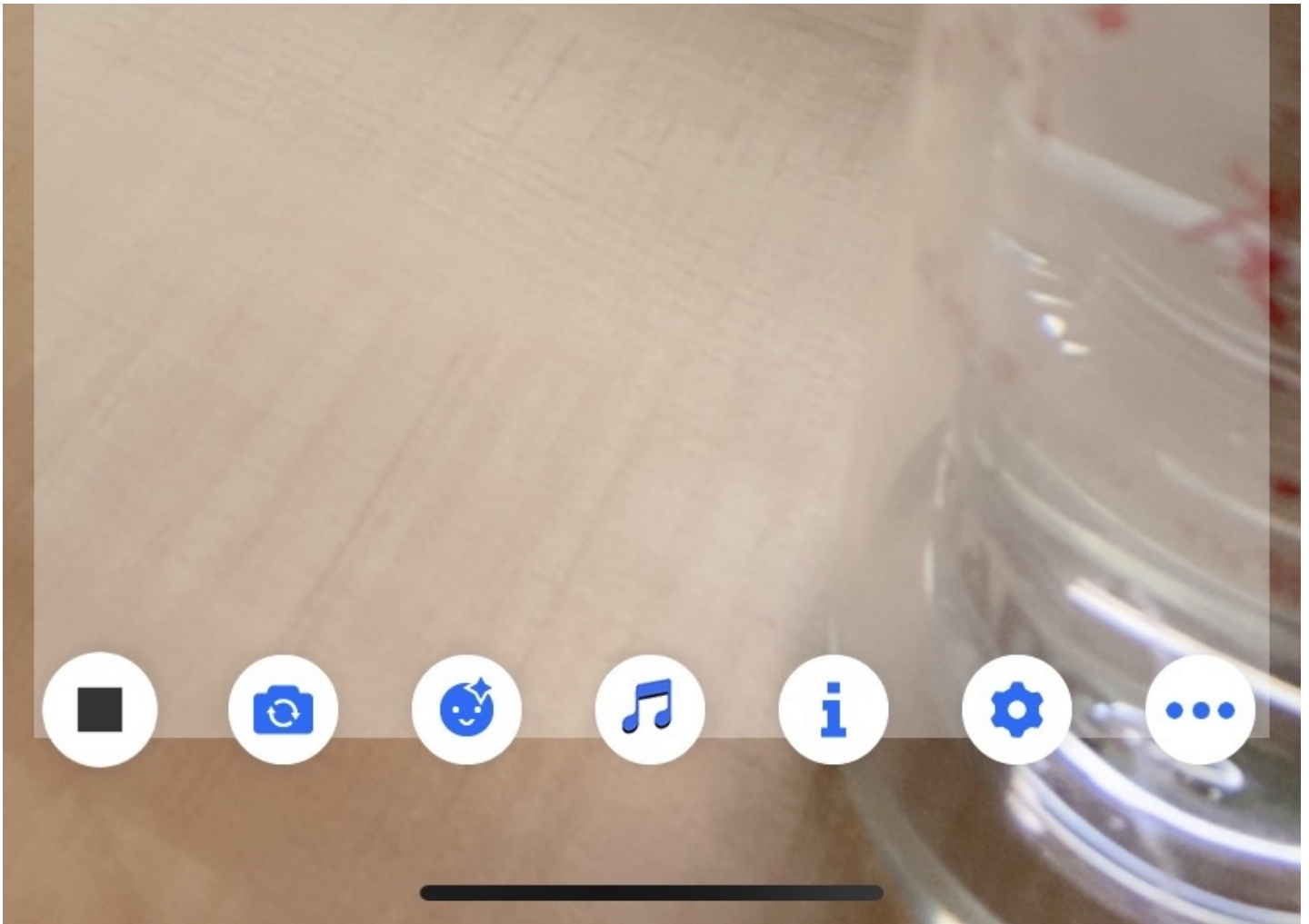
You are advised to set `txTime` to 12 or 24 hours from the current time, longer than a common live streaming session.

4. Check whether `txSecret` is correct.

To ensure security, Tencent Cloud requires configuring hotlink protection for all push URLs and **rejects** all hotlink protection URLs that have expired or are miscalculated. If a push is rejected, the CSS SDK will throw a **PUSH_WARNING_SERVER_DISCONNECT** event.







See [Best Practice > Live Push](#) for how to get reliable push URLs.

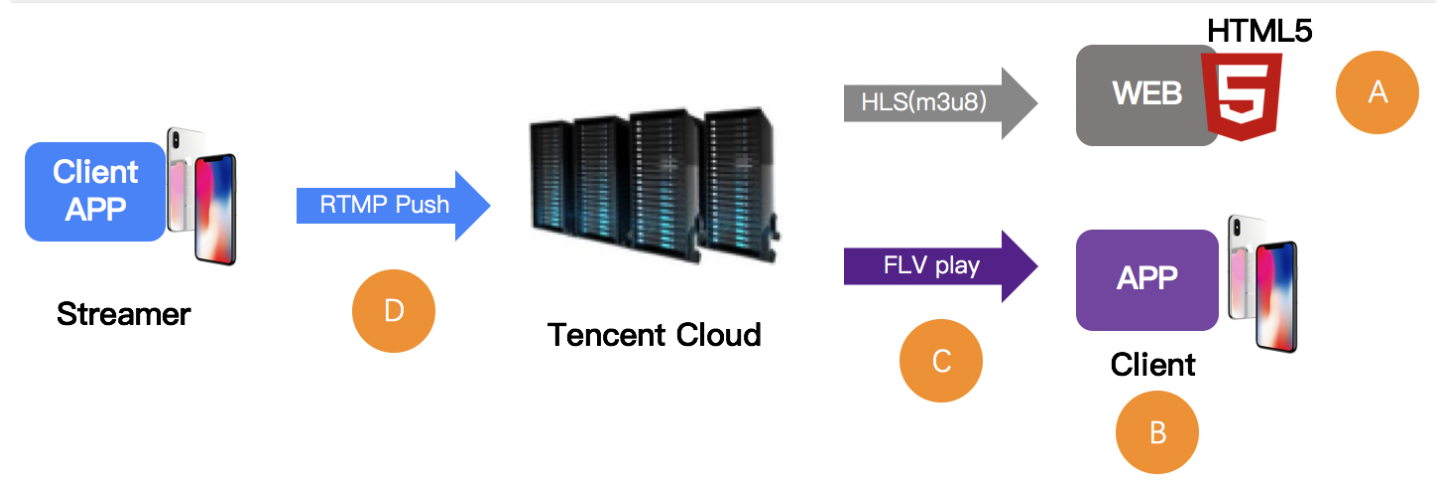
5. Check whether the push URL is occupied.

A push URL can be used by only one client at a time. A second client trying to push using the URL will be rejected by Tencent Cloud. You can log in to the CSS console and check whether a stream is already being pushed in [Stream Management](#) > **Live Streams**. You can also check whether a stream is disabled in **Disabled Streams**.

Troubleshooting Playback Failure

Last updated : 2022-03-14 14:44:09

If you're unable to watch the CSS and have no idea what goes wrong with it, you can identify the cause of the problem in a short time by following the steps below:



Step 1. Check the playback URL

First of all, check whether the playback URL is correct. An incorrect URL is the most likely cause of most problems. Tencent Cloud's CSS URLs include push URL and playback URL. You need to first verify whether **the push URL is accidentally used CSS as the playback URL**.

RTMP Push address	<code>rtmp://6666.livepush.myqcloud.com/live/6666_XXXXXXXXXXXX?bizid=6666</code>
RTMP Play address	<code>rtmp://6666.liveplay.myqcloud.com/live/6666_XXXXXXXXXXXX</code>
FIV Play address (recommended)	<code>http://6666.liveplay.myqcloud.com/live/6666_XXXXXXXXXXXX.flv</code>
HLS Play address	<code>http://6666.liveplay.myqcloud.com/6666_XXXXXXXXXXXX.m3u8</code>

Playback URL for Mini CSS:

The playback URL for Mini CSS can be obtained through debugging. You can search for the keyword

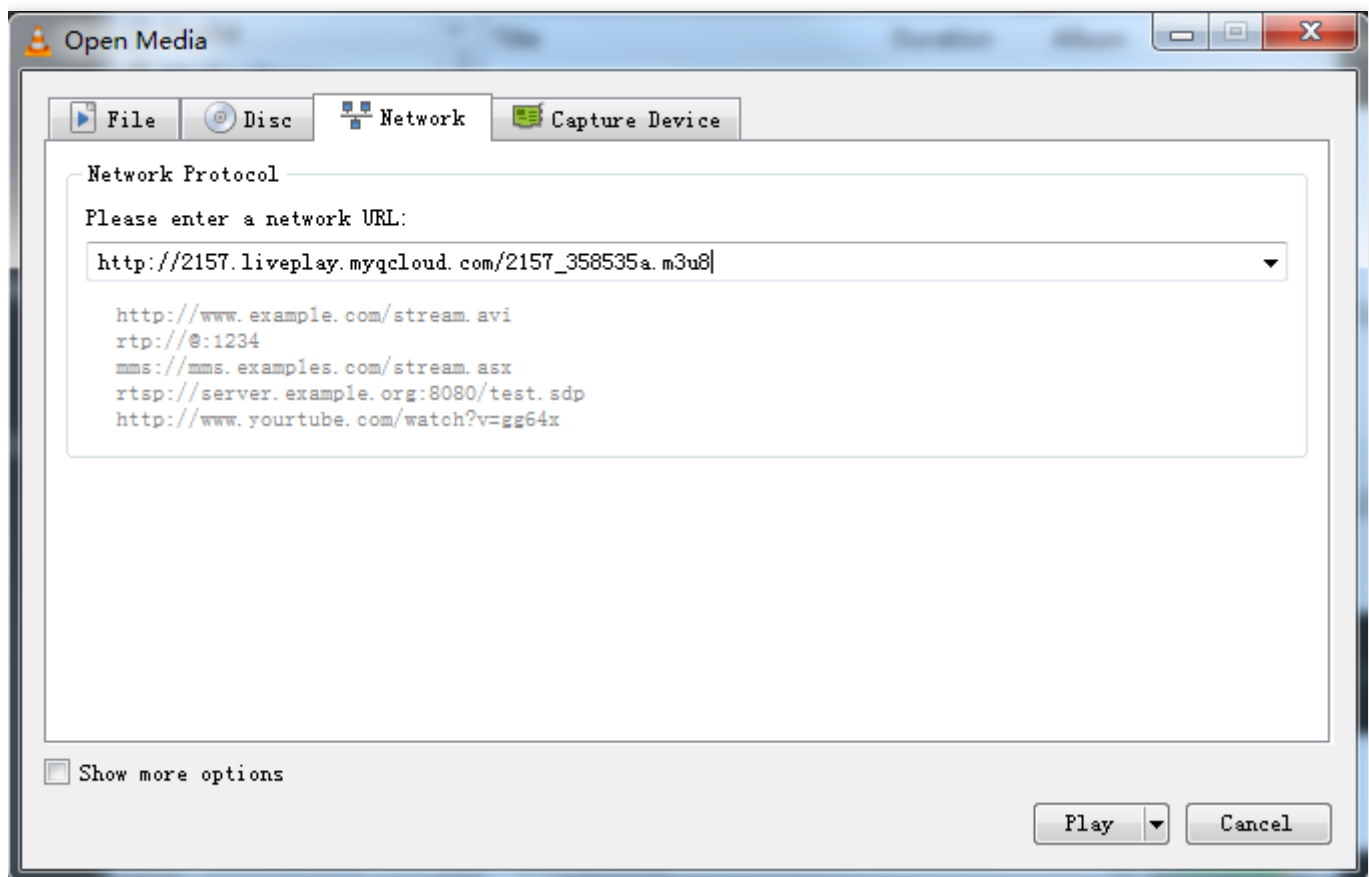
startPlay in the global search, then set a debugging breakpoint, where the RTMP SDK is called by Mini CSS. The parameter startPlay is the playback URL.

Step 2. Check the video stream

A correct playback URL does not always mean a normal playback. Next, you need to check whether the video stream is normal:

- In **CSS**, the CSS URL becomes unavailable once the VJ stops the push.
- In **VOD**, if the video files have been removed, watching videos is also impossible.

A frequently used solution is making a check using VLC, an open-source player on PC that supports many protocols.



Step 3. Check the player

If there's no problem with the video stream, then you need to check whether the player is normal on a case-by-case basis:

3.1 Web browser (A)

- **Format:** Mobile browsers only support playback URLs in **HLS (m3u8) and MP4** formats.
- **HLS (m3u8):** Tencent Cloud HLS protocol is based on "Lazy Start". In short, Tencent Cloud only starts the transcoding for HLS format when a viewer requests a playback URL in an HLS format. The purpose is to prevent waste of resources. But it also creates a problem: **The playback URL in an HLS format cannot be played until 30 seconds after the first user in the world initiates a request.**

3.2 RTMP SDK (B)

If RTMP SDK DEMO works normally for playback, it's recommended to check whether the interfacing logic is incorrect by referring to the RTMP SDK playback document [iOS] & [Android].

Step 4. Check for firewall blocking (C)

It is common that the corporate network environments of many customers restrict video playback through firewalls that detect whether the resources requested by HTTP are streaming media resources (After all, no boss wants his employees to watch videos during working hours). The fact that you can watch the CSS normally over 4G network but cannot watch it over your company's Wi-Fi network indicates your company has imposed restrictions on the network policies. In this case, contact the administrator for a special treatment of your IP.

Step 5. Check the pusher (D)

If the CSS URL does not work and there is no possibility of firewall blocking described in Step 4, it is likely that the push is unsuccessful. Go to Why the Push is Unsuccessful for a further troubleshooting.

Troubleshooting High Latency

Last updated : 2024-03-01 11:40:47

If you push streams via RTMP and play streams via HTTP-FLV, the latency is generally about 2-3 seconds. If you experience high latency, follow the steps below to troubleshoot the problem.

Step 1. Check your playback protocol

The latency tends to be high if you use HLS (M3U8) for playback. HLS is a streaming protocol developed by Apple. It works by breaking streams into (usually 3 or 4) TS segments of 5 seconds or longer, which results in an overall latency of 10-30 seconds.

Therefore, if you have to use HLS (M3U8) for playback, you can reduce latency by cutting the number of segments or the length of each segment, but this may increase stuttering. You can [submit a ticket](#) or contact our technical support engineers for help.

For specific configuration methods, please refer to [troubleshooting high latency](#).

Step 2. Check player settings

The player of the MLVB SDK supports three latency control modes: Speedy, Smooth, and Auto, For specific settings, please refer to [Latency Control](#):

Speedy: this mode keeps latency at 2-3 seconds or lower in most application scenarios and is suitable for live showroom.

Smooth: this mode keeps latency at 5 seconds or lower in most application scenarios and is suitable for application scenarios that require smooth playback but are not sensitive to latency, such as game streaming.

Step 3. Watermark videos on the client side

Tencent Cloud allows you to watermark videos in the cloud, but this will increase latency by 1-2 seconds. Therefore, if you use the MLVB SDK, we recommend you watermark videos at the host end instead of in the cloud to reduce latency.

Step 4. Check third-party pushers

We guarantee superior streaming experience via our integrated solution, but if you use third-party software to push streams, we recommend that you compare your pusher with Tencent Cloud Live Streaming Console's [Web Push](#), to see if your pusher is the cause of high latency. Many third-party pushers tend to keep increasing the buffer size to mitigate the problem of low upstream bandwidth.

Step 5. Check OBS settings

If you use OBS to push streams and experience high latency, check your configuration against [Push via OBS](#). Make sure you set the keyframe interval to 1 or 2 seconds.

Step 6. Use LEB

If none of the above solves your problem, you can try using Tencent Cloud's LEB service, which features lower latency than LVB and offers streaming with millisecond latency. For details, please see [LEB](#).

Step 7. Submit a Ticket

Since live streaming uses the caching mechanism of audio and video to cache data, using pure audio for pushing and pulling streams may result in higher data latency. You can [submit a ticket](#) to resolve the latency issue through configuration.

Troubleshooting Poor Quality of Pulled Video

Last updated : 2021-03-12 11:32:56

The video of the host is captured by the local camera, encoded and pushed by the SDK for client, and then distributed to the viewers by Tencent Cloud CDN. If you provide a multi-bitrate address, the video stream will also be re-encoded in Tencent Cloud. The quality of output video mainly depends on the quality of input video captured by camera, and the frame rate, keyframe interval, resolution, and bitrate configured for encoding. Considering the influence on video delay and bitrate, we recommend setting the keyframe interval to 2-3 seconds.

This document describes how to troubleshoot the problem that the video pulled to the following **two** playback addresses is pixelated and how to optimize the video quality.

Playback Address of Input Video Stream

If the video pulled to the **playback address of the input video stream** (neither watermarked nor mixed) is pixelated, we recommend locating the problem at the push end:

1. Troubleshoot problems of the camera. For example, check whether there is dust, or whether the camera has focused or captured properly.
2. Check whether the frame rate and bitrate of the pushed stream are as expected.

Resolution	Frame Rate	Expected Bitrate
640 × 368	15 fps	800 Kbps
960 × 544	15 fps	1,000 Kbps
1280 × 720	15 fps	1,500 Kbps
1920 × 1080	15 fps	2,500 Kbps

Optimization

- If you use a third-party SDK, please refer to the above recommended bitrates to adjust and control the video quality, or contact the third-party SDK manufacturer to solve the problem.
- If the video in the preview window has a good quality but the pulled video is pixelated, this may be because the quality of the video in the preview window is inconsistent with that of the encoded and pushed one. Please refer to the above recommended settings to adjust the quality of the encoded and pushed video.

Playback Address of Video Stream at Low Bitrate and Resolution

If the video pulled to the **playback address of a low-bitrate and low-resolution video stream** is pixelated, you need check the video quality of the playback address of the input video stream. If the input video has a good quality, so does the video pushed by the client. In this case, we recommend that you modify the transcoding parameters in Tencent Cloud. For example, you can increase the output bitrate to a recommended value in the transcoding template.

Resolution	Frame Rate	Expected Bitrate
640 × 368	15 fps	800 Kbps
960 × 544	15 fps	1,000 Kbps
1280 × 720	15 fps	1,500 Kbps
1920 × 1080	15 fps	2,500 Kbps

For example, if the video resolution is 640 × 368 and the template frame rate is 30 fps, we recommend increasing the output bitrate to 1.5 times of its original value, so the recommended bitrate is calculated by $800 \text{ Kbps} \times 1.5 = 1,200 \text{ Kbps}$.

Note :

If the problem persists after all the above steps are completed, please [submit a ticket](#).

Authorizing CSS to Store Screenshots in a COS Bucket

Last updated : 2023-10-07 11:50:44

This document describes how to store screenshots or porn detection data in a COS bucket. You need to create a COS bucket, authorize CSS to store data in it, and then configure live screencapture and porn detection settings in the CSS console. After that, screenshots and porn detection data can be stored in the bucket. This feature is available in the new console.

Creating a COS Bucket

1. Log in to the COS console and select [Bucket List](#) on the left sidebar.
2. Click **Create Bucket**. In the pop-up window, enter the basic information, select a permission for the bucket, and click **Next**.

Create Bucket

1 Information > 2 Advanced optional configuration > 3 Confirm

Region: Europe

Services within the same region can be accessed through private network

Name: test -130 592 ✓

The value can contain only lowercase letters, digits, and hyphens (-). The total number of characters in a domain name cannot exceed 60 characters. Once set, the bucket name cannot be changed

Access Permission: ☒ Private Read/Write ☐ Public Read/Private Write ☐ Public Read/Write

Identity verification is required before accessing objects.

Endpoint: test-130592.cos.eu-moscow.myqcloud.com

Request endpoint

Cancel Next

3. Complete the advanced configuration (optional) and click **Next**.

Create Bucket

✓ Information

2 Advanced optional configuration

3 Confirm

Versioning

☐

Keeping multiple versions of an object in the same bucket will incur storage usage fees.[Learn More](#)

Bucket Tag

+

You can also create 49 labels to manage buckets in groups by adding bucket labels.[Learn More](#)

Server-Side Encryption

☒ None

☐ SSE-COS

?

Previous

Next

4. Confirm the configuration information and click **Create**.

Create Bucket

Information

Advanced optional configuration

3 Confirm

Name ⓘ	test-130592
Region	
Access Permission	Private Read/Write
Endpoint	test-130592.cos.eu-moscow.myqcloud.com
Versioning	Disable
Server-Side Encryption	None

Previous

Create

Caution

In the example above, the bucket name is `test` (`-130****592` is not part of the name).

Complete the above settings based on your actual needs.

5. To enable CDN acceleration, click the name of your bucket or click **Configure** in the bucket list, and select **Domains and Transfer > Default CDN Acceleration Domain** on the left sidebar. Then, click **Edit**, toggle on **Status**, and complete the settings. For detailed directions, see [Enabling Default CDN Acceleration Domain Names](#). After the configuration, click **Save**.

Default CDN Acceleration Domain

Edit

Status Disable

Note: If the default CDN acceleration domain is enabled, you can access websites using this domain for acceleration. The use of CDN accelerated domain name will generate CDN back traffic. Please purchase [CDN origin-pull traffic pack](#). COS's CDN acceleration domain is powered by Tencent Cloud's CDN service. Therefore, all COS-side CDN configurations. For more information, please see [Default CDN Acceleration Domain Guide](#).

Authorizing CSS to store screenshots in COS

1. Grant the root account 3508645126 **write access** and **read access** to the COS bucket.

1.1 In the [Bucket List](#), find the bucket you created, and click **Configure**. On the bucket configuration page, select **Permission Management > Bucket ACL (Access Control List)**. Then, click **Add User**, select **Root account** as the user type, **enter the root account ID 3508645126** , and click **Save**.

Create Bucket	Manage Permissions	Bucket Name
Bucket Name ↓	Access ▾	Region ▾
test-130-592 ⓘ	Specified user	2021-11-17 09:27:40

Bucket ACL(Access Control List)

Public Permission ☐ Private Read/Write ☒ Public Read/Private Write ☐ Public Read/Write

User ACL

User Type	Account ID ⓘ	Permission
Root account		Full control
Root account ▾	3508645126	<input checked="" type="checkbox"/> Reads control <input checked="" type="checkbox"/> Write <input type="checkbox"/> Read ACL (

Add User

Save Cancel

Alternatively, in the bucket list, click **Manage Permissions**, select the bucket you created, Then, click **Add User**, select **Root account** as the user type, enter the root account ID 3508645126 , click **Save** to save the configuration, and then click **OK**.

Create Bucket	Manage Permissions	Bucket Name
Bucket Name ↓	Access ▾	Region ▾
test-130-592 ⓘ	Specified user	2021-11-17 09:27:40

Manage Permissions

You can manage permissions (such as modifying public permissions or adding permissions) for multiple buckets at a

Select (2 buckets)

Search bucket name



Bucket Name

Region ▼



1

Mumbai



te

1

Nanjing

(1) selected

Bucket Name

t

,1



Public Permission



Private Read/Write



Public Read/Private Write



Public Read/Write

User ACL

User Type	Account ID ⓘ	Permission
Root account	200021000557	Full control
Root account ▼	3508645126	<input checked="" type="checkbox"/> Reads <input type="checkbox"/> Write ACL ⓘ <input checked="" type="checkbox"/> Write <input type="checkbox"/> Full control
Add User		

The authorization does not require a permission consistency check. The new permission will overwrite the permission already been added to the bucket.

OK

Cancel

Caution

3508645126 is the APPID of CSS. You need to enter this ID for the authorization to succeed.

1.2 For information about how to use an API to set bucket access, see [PUT Bucket acl](#).

2. Get the information of the COS bucket to which CSS is granted access.

2.1 Click the name of the bucket you created in the bucket list and select [Overview](#) to view its information. You can find in the endpoint the bucket name, COS APPID, and bucket region.

Usage Overview STANDARD

Number of Objects 0 individual
VS Yesterday: ↑ 0%
VS Last Month: ↑ 0%

Storage 0 B
VS Yesterday: ↑ 0%
VS Last Month: ↑ 0%

Month-to-Date Total Traffic 0 B
last monthTotal Traffic 0 B

Information [Delete bucket](#)

Bucket Name test01-130****051 (Buckets do not support renaming)
Region Frankfurt (Europe) (eu-frankfurt)
Creation Time 2023-09-11 11:04:47
Access Permission Private Read/Write

Alarm Configuration [Configure Alarm Policy](#)

☒ Current Alarms 0
☐ Configured Alarm Policies 0

Data Processing

You have not enabled the CI service.

Note: COS uses the CI service to provide data processing, the fee will be charged by CI. For details, click [Introduction of Functions](#) and [Billing Description](#).

Bucket Configuration

MAZ configuration Disabled
CORS 0 rule(s)
Origin-Pull Not configured
Hotlink Protection Not configured
Cross-Bucket Replication Not configured
Tag Empty

Domain Information

Endpoint https://test01-130****051.cos.eu-frankfurt.
Custom CDN Acceleration Disabled
Custom Endpoint 0 per second
Global Acceleration Endpoints 0
Static Website Endpoint

Note: COS domains use smart DNS. If your other Tencent Cloud serv address. Cross-region access cannot use a private network and thus [Creation Overview](#)More.

Bucket name: test01

COS appid: 130****051

Bucket region: eu-frankfurt

2.2 Provide the above information to CSS and the system will store live screenshots in the COS bucket you created.