WebRTC Use Cases and Requirements for Real-Time Interactive Scenarios

Li Lin
China Mobile MIGU
13 September 2023

Background

With the continuous development of new technologies and emerging application scenarios, the livestreaming multimedia ecosystem is experiencing robust growth. Major technology companies in China are strategically involved in it.

中国移动 China Mobile

Large-scale livestreaming like sports events and music concerts, metaverse convention

ByteDance

Educational livestreaming, livestreaming e-commerce, next-gen conferencing

Alibaba

Livestreaming e-commerce, fundamental livestreaming cloud services

Scenario 1: Real-time interactive Cloud Box

“Cloud Box” :

- Customized on-line private room for watching sports event, concert, etc., together with families and friends , which has been applied in the Tokyo Olympics, Beijing Winter Olympics and Qatar World Cup, and music concerts like the MIGU Music Awards
- The participants in “Cloud Box” can chat via video, voice and text with each other, while watching sports events, movies and TV together – meets the social needs of the participants.
- The viewers in “Cloud Box” can watch the live broadcast and the interaction among the participants, but cannot participate the interaction.

Real-time interactive Cloud Box for sports events, concert, movies, TV, etc.

Use case 1: Real-time interactive Cloud Box

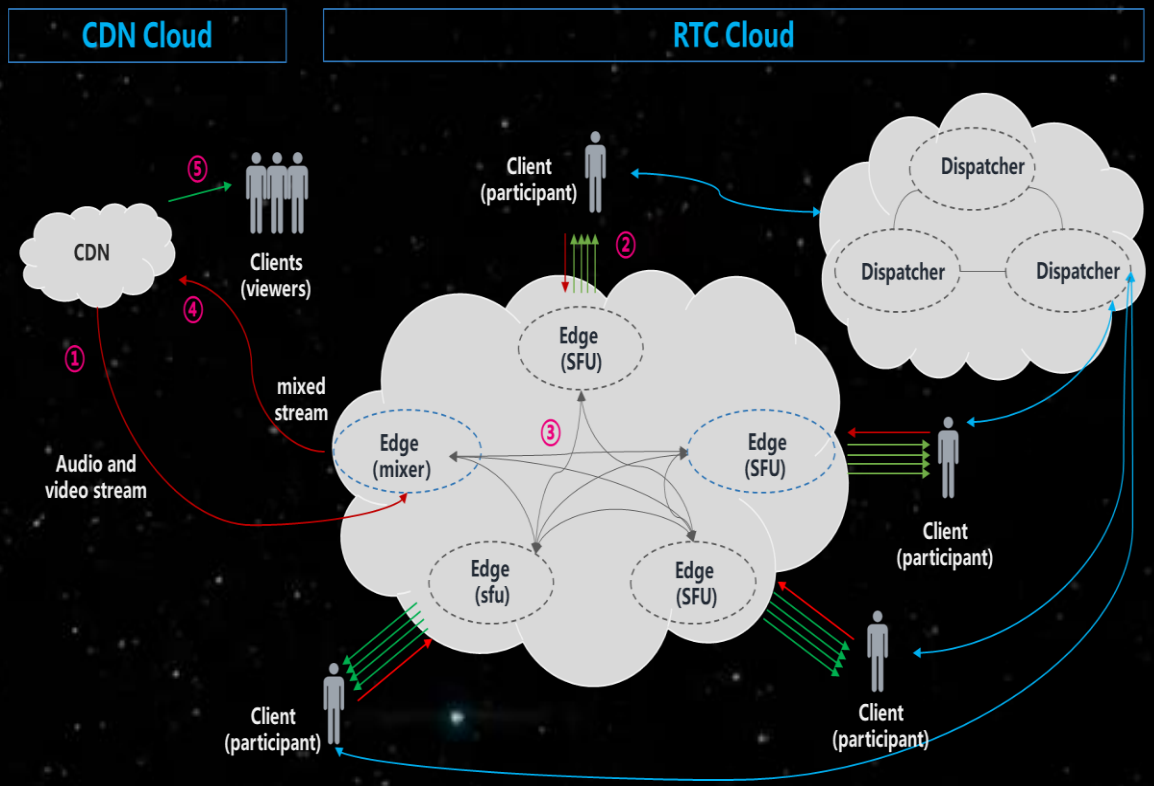
① The live and on-demand media streaming (sports events, movies, TV, etc.) are pushed from CDN via RTMP to the RTC edge node which distributes them to the RTC cloud

② The clients of the participants in the cloud box, pull the live and on-demand media streaming from the RTC cloud, and push the real-time interaction media streaming captured via the microphone and camera to the RTC cloud

③ The real-time interaction media streaming pushed by the participants in the cloud box is distributed through the RTC cloud including the edge nodes which are capable of media stream mixing and forwarding

④ The RTC edge nodes mix the live and on-demand media streaming with real-time media streaming pushed by the participants, and forwards to the CDN via RTMP

⑤ The clients of the viewers in the cloud box pull the mixed media streaming via HLS, RTMP, HTTP-FLV, WebRTC, etc., from CDN



RTC and CDN cloud for real-time interactive Cloud Box

Scenario 2: Real-time interactive live e-commerce

Real-time interaction with large scale audiences:

- Live streaming e-commerce, enabling interactive product showcases, direct audience engagement, and instant purchases.
- The anchor introduces the features of the products, demonstrates how to use, address customer inquiries, and offers exclusive deals or limited-time promotions via live streaming
- The participants(invited guests) can real-time interact with the anchor aiding the sale through video chat powered by WebRTC or text messages
- The viewers(customers) watch the anchor live streaming and the interaction with the guests; interact with the anchor via text messages

Real-time interactive live e-commerce

Use case 2: Real-time interactive live e-commerce

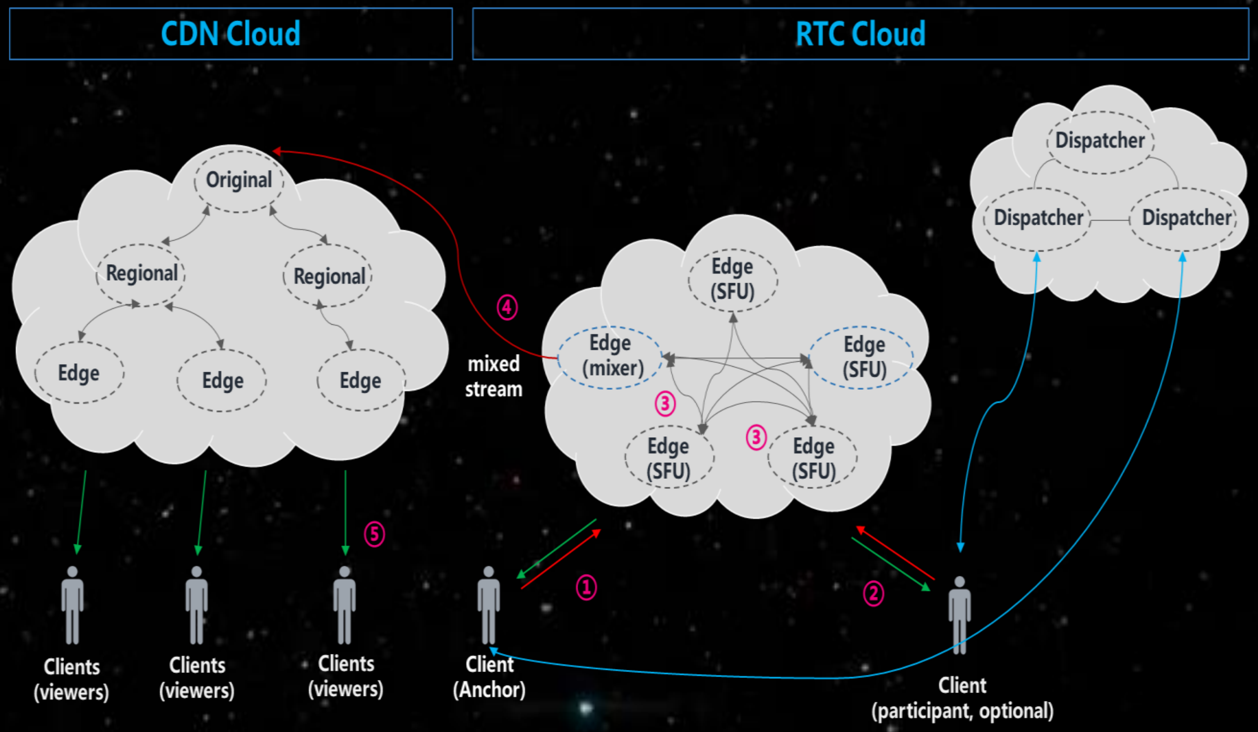
① The anchor initiates live streaming via RTC client, demonstrate the products via the camera and push the media contents of product demonstration to the RTC edge nodes

② Sometimes a few participants(invited guests) can real-time interact with the anchor on-line through the RTC network to assist product selling

③ The media streaming pushed by the anchor and participants is distributed through the RTC cloud including the edge nodes (mixer) which are capable of media stream mixing and forwarding

④ The RTC edge nodes (mixer) mix the real-time media streaming pushed by the anchor and the participants, and then forward to the CDN via RTMP

⑤ The clients of the viewers pull the mixed media streaming via HLS, RTMP, HTTP-FLV, WebRTC, etc., from the CDN



RTC and CDN cloud for real-time interactive live e-commerce

Scenario 3: Metaverse Convention Center (MCC)

Immersive, real-time interactive platform:

- Kind of cloud game, built on the foundation of the technologies including audio-video interactive communication capabilities powered by WebRTC, 3D resources rendering, and AI-generated avatars, etc.
- Upgrade current traditional audio and video conferencing products to virtualize, gamify, and enhance interactivity for remote meetings, office, events, etc.
- The participants can create their own virtual characters, including cartoon and realistic styles
- With a variety of meeting rooms, the participants can choose the venue according to their own preferences

MIGU Metaverse Convention Center

Use case 3: Metaverse Convention Center (MCC)

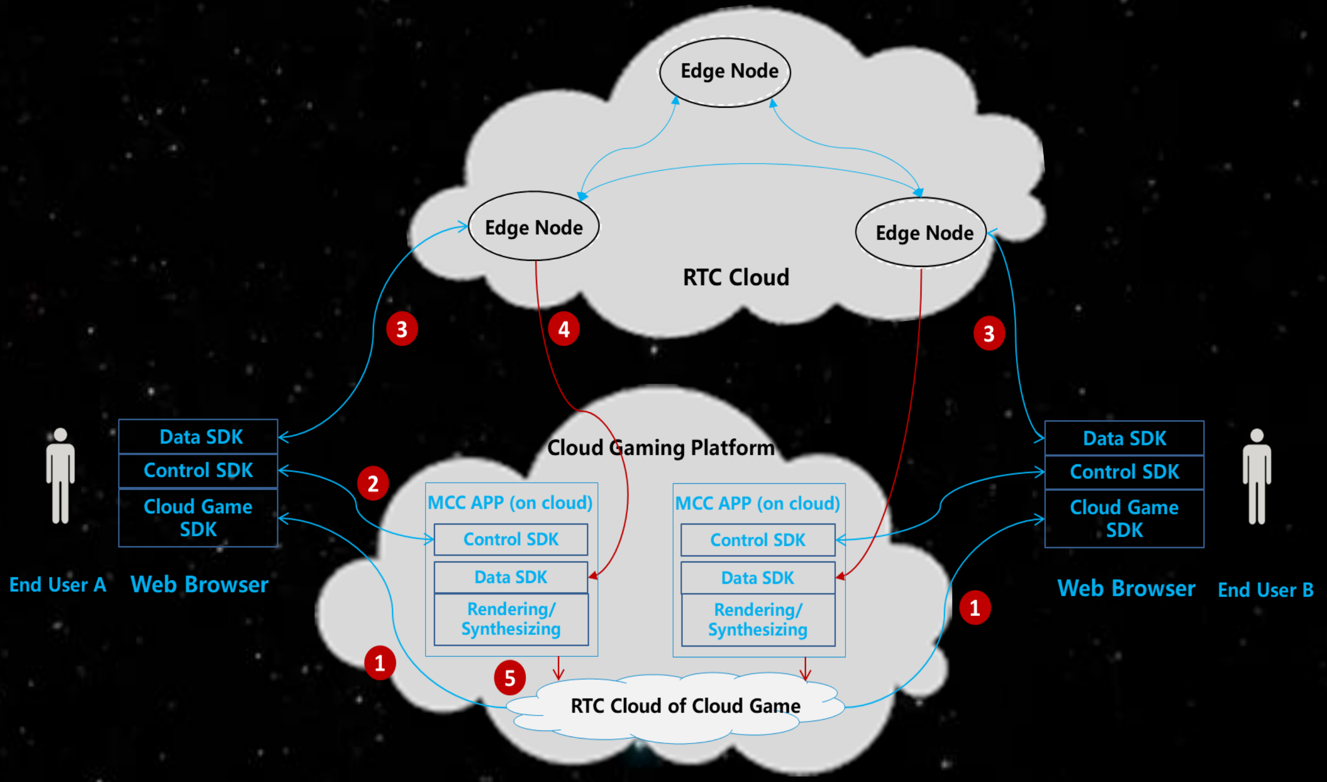
① The cloud game screen streaming is pushed to the Web browser through RTC cloud of cloud game.

② The browser of the user communicates control info through WebRTC Data Channel with the MCC application on the RTC cloud of cloud game. The control info includes user login, audio and video client-cloud coordination operation control, heart-beating, etc.

③ The audio and video media streaming of the user produced by the Web browser is pushed to RTC cloud via WebRTC SDK. And the audio streaming is pulled from RTC cloud to the Web client of the user.

④ The user video streaming is pulled from the RTC cloud by WebRTC SDK and rendered together with game screen.

⑤ The rendered game streaming is pushed to RTC cloud of cloud game and pulled by Cloud Game SDK to the user.



RTC cloud for real-time interactive live e-commerce

Requirement 1: relative to stats of frame freeze

Requirement

① An API is expected to added in WebRTC specification for setting the duration of frame freeze which can be used to calculate *freezeCount* and *totalFreezesDuration* in Identifiers for WebRTC's Statistics API

② The existing *totalInterruptionDuration* and *interruptionCount* in *audio_coding_module_typedefs.h* for audio frame interruption statistics are expected to be added to WebRTC specification in Identifiers for WebRTC's Statistics API.

```
129    // number of audio interruptions
130    int32_t interruptionCount;
131    // total duration of audio interruptions
132    int32_t totalInterruptionDurationMs;
```

Gap analysis

freezeCount of type unsigned long

Count the total number of video freezes experienced by this receiver. It is a freeze if frame duration, which is time interval between two consecutively rendered frames, is equal or exceeds Max(3 * avg_frame_duration_ms, avg_frame_duration_ms + 150), where avg_frame_duration_ms is linear average of durations of last 30 rendered frames.

- The statistics of frame freeze is important to WebRTC network problem diagnosis and network optimization
- In the existing *freezeCount* algorithm, frame duration is used for calculation:

It is a freeze if frame duration, which is time interval between two consecutively rendered frames, is equal or exceeds Max(3 * avg_frame_duration_ms, avg_frame_duration_ms + 150), where avg_frame_duration_ms is linear average of durations of last 30 rendered frames.

- The algorithm is ok when the RTC network situation is not too bad.
- The algorithm cannot fully meet the needs in practice when the network condition is bad, e.g., the linear average of durationsof last 30 rendered frames (avg_frame_duration_ms) is 200ms, it is not taken as a freeze until when the frame duration exceeds 600ms.

—— In practice, the video is freezed on the receiver when the consecutive frame duration exceeds 200ms.

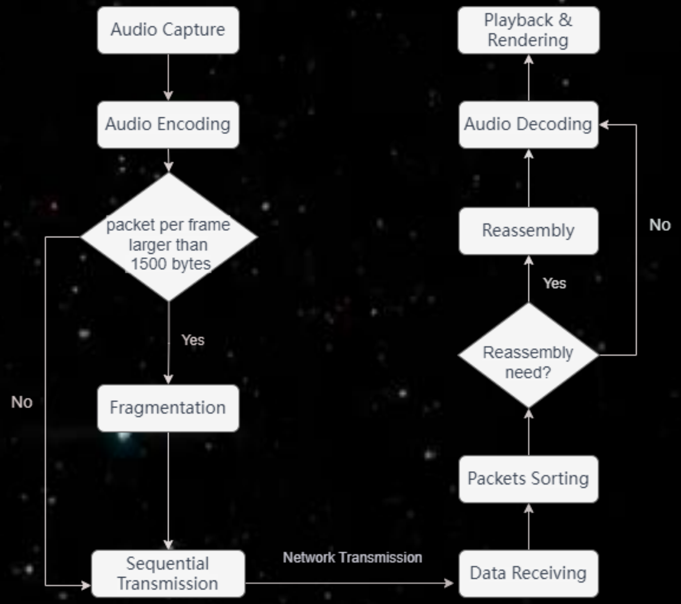
Requirement 2: relative to audio transmission

Requirement

It is expected to support the transmission of audio data packet, the length of which bigger than IP packet MTU (1500 bytes), by splitting audio frame into multiple packets at the sender and reassembling at the receiver.

Gap analysis

- Currently an audio frame is encapsulated in one RTP packet when it is transmitted via WebRTC which not support to split one audio frame into several packets. When the length of the audio packet is bigger than the IP packet MTU limit (typically 1500 bytes), the audio packet transmission fails.
- Some audio codecs, such as Dolby Surround 7.1, may have a total data length of more than 1500 bytes per data packet
- WebRTC supports to handle fragmentation and reassembly for video transmission, not for audio.



High-level fragmentation and assembly workflow

Issue —— Failed to start the audio playback device when the iOS app in the background

Issue Reproduction Steps:

① Two users A and B (iOS device) join the WebRTC conference, user A turn on the microphone and user B not.

② User B switches iOS application to the background, and user A not switched to the background.

③ Turn off the microphone of the conference application of user A.

④ Turn on the microphone of the conference application of user A.

⑤ At this time, the conference application of user B cannot hear the voice of user A.

⑥ Bug issue link: https://bugs.chromium.org/p/webrtc/issues/detail?id=13406

Root Cause:

- The issue only exists in iOS devices.
- In WebRTC, when the incoming audio data packets reaches zero, the audio playback device is closed. When the audio data stream resumes, the device is reopened.
- WebRTC does not differentiate between foreground and background operation.
- iOS restricts audio device activation in the background, resulting in restarting the audio device failure.

```
void AudioState::RemoveReceivingStream(
    webrtc::AudioReceiveStreamInterface* stream) {
  RTC_DCHECK_RUN_ON(&thread_checker_);
  auto count = receiving_streams_.erase(stream);
  RTC_DCHECK_EQ(1, count);
  config_.audio_mixer->RemoveSource(
      static_cast<AudioReceiveStreamImpl*>(stream));
  UpdateNullAudioPollerState();
  if (receiving_streams_.empty()) {
    config_.audio_device_module->StopPlayout();
  }
}
```

audio_state.cc

Next step

- Continue to refine WebRTC interactive scenarios, use cases and requirements in a TF of WNIG or new CG for "Uses Cases and Rrequirements: Low latency transmission for Live Streaming Multimedia Ecosystem"
- Create official Github repository for the TF or CG

Current Draft: https://webrtc-live-streaming-tf.github.io/webrtc-use-cases/

TABLE OF CONTENTS

WebRTC Use Cases and Requirements for Live Streaming Multimedia Ecosystem

10 September 2023

▼More details about this document

Latest published version:
https://www.w3.org/webrtc-live-streaming-use-cases/

Latest editor's draft:
https://webrtc-live-streaming-tf.github.io/webrtc-use-cases/

History:
Commit history

Editors:
Lin Li (China Mobile)
Cheng Chen (ByteDance)
Song Xu (China Mobile)
Fuqiao Xue (W3C)

Feedback:
GitHub webrtc-live-streaming-tf/webrtc-use-cases (pull requests, new issue, open issues)

Web Real-Time Communication Lead the Web to interactive and immersive potential