网易云 | 通信与视频

网易NRTC支持WebRTC的工业级实践

2017年10月21日

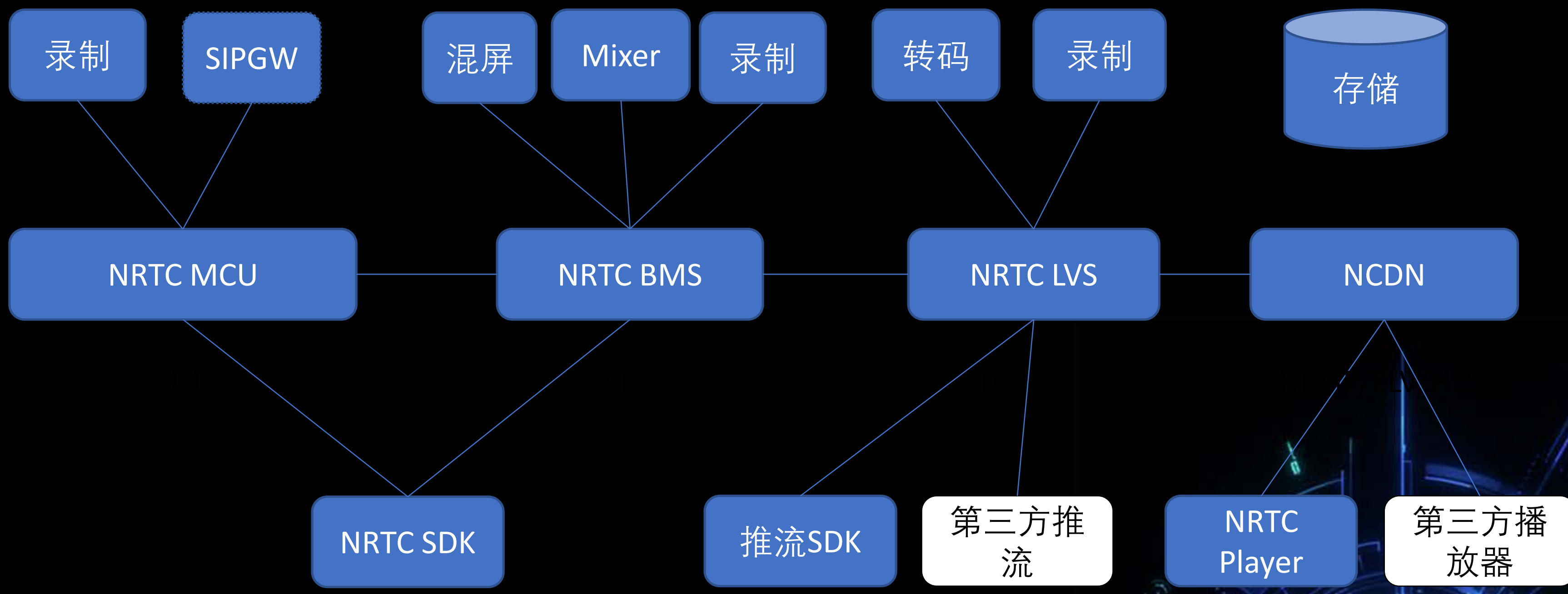
赵加雨Rain

LiveVideoStackCon

NRTC

- NetEase RTC
- 工业级的功能完整的音视频技术方案

NRTC Architecture



NRTC Capabilities

- 实时音视频通话
- 直播
- 互动直播
- 点播
- 短视频

音视频技术栈

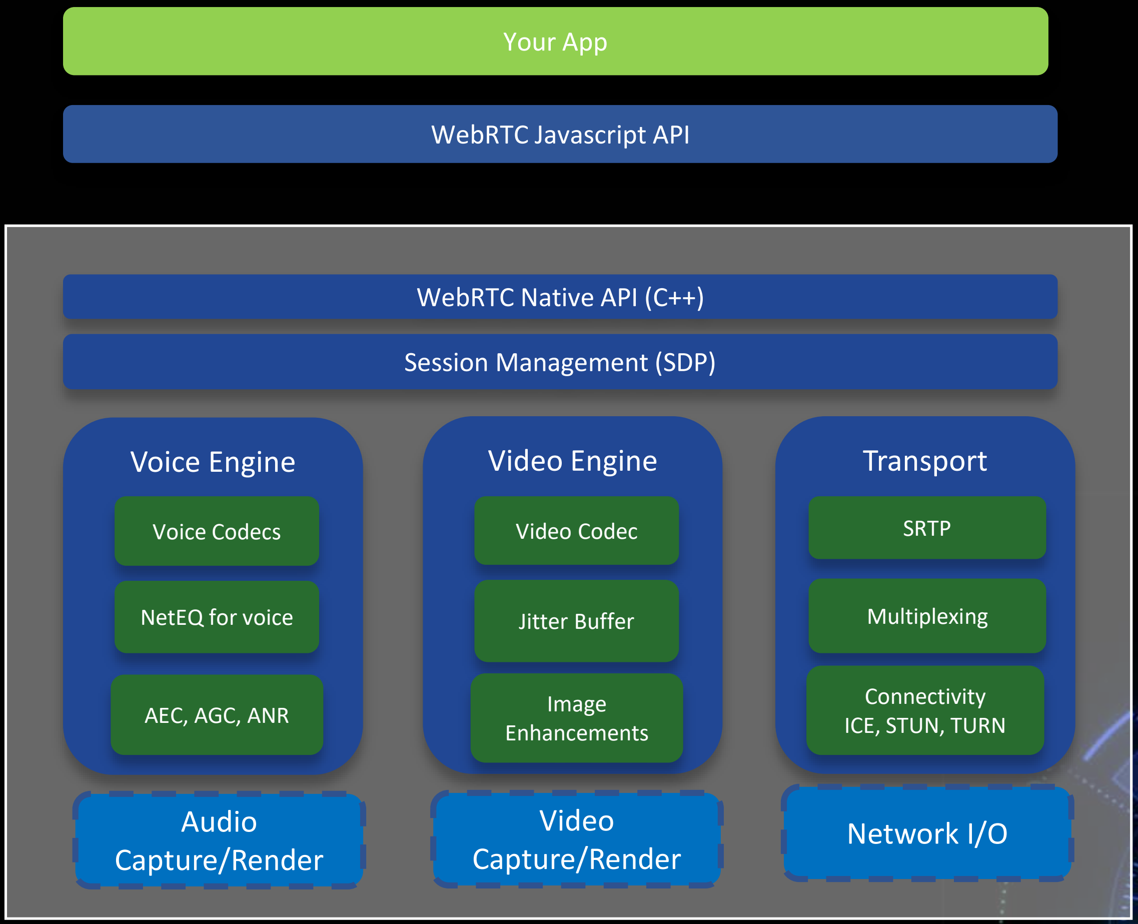
- 信令：SDP, JSEP, SIP, Jingle, ROAP
- 传输：RTP, RTCP, DTLS, RTMP, FLV, HLS
- P2P：ICE, STUN, TURN, NAT
- 网络：UDP, TCP
- 音频：Opus, G711, AAC, Speex, 3A
- 视频：H264, VP8
- QoS：FEC, NACK, BWE
- Server：SFU, MCU
- 端：Capture, Render, 各种适配

WebRTC introduction

- The specification in browser
  - Web Real-Time Communication
  - A specification that is being standardized by W3C and IETF
  - Enables web browsers with audio, video and sharing capabilities via simple JavaScript APIs
  - Zero install
  - Peer 2 peer, also capable for conference
  - Interoperability with existing voice and video systems
- An open source project
  - Contributed by Google
  - In C++ and it is cross platform
  - From Google's acquisition of Global IP Solutions
  - A fully implemented VoIP client framework

WebRTC Architecture

网易云 | 通信与视频



LiveVideoStackCon

WebRTC limitations

- JavaScript APIs, run in browser
- Signaling not defined
- No SFU/MCU
- Follow standards
- 依赖browser的实现

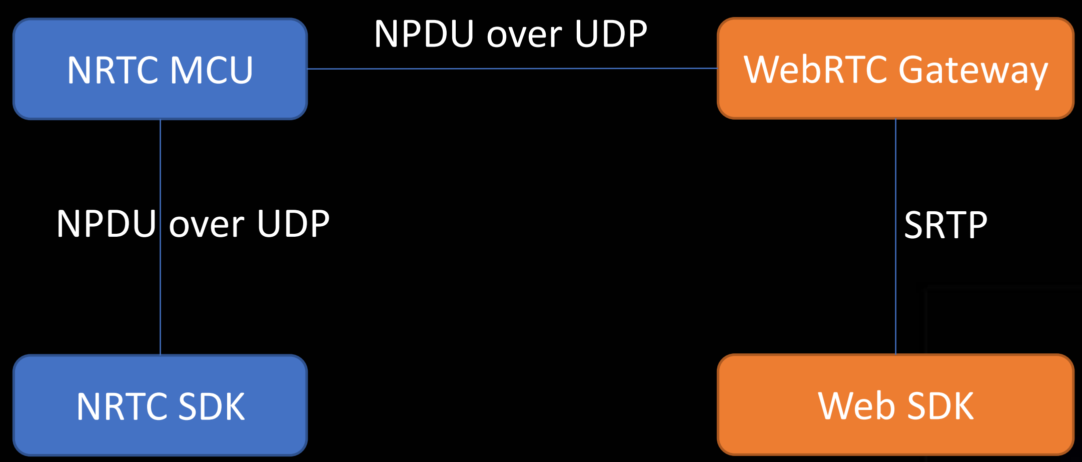
使用WebRTC

- JavaScript APIs
- libwebrtc
- Compatible with WebRTC

NRTC vs WebRTC

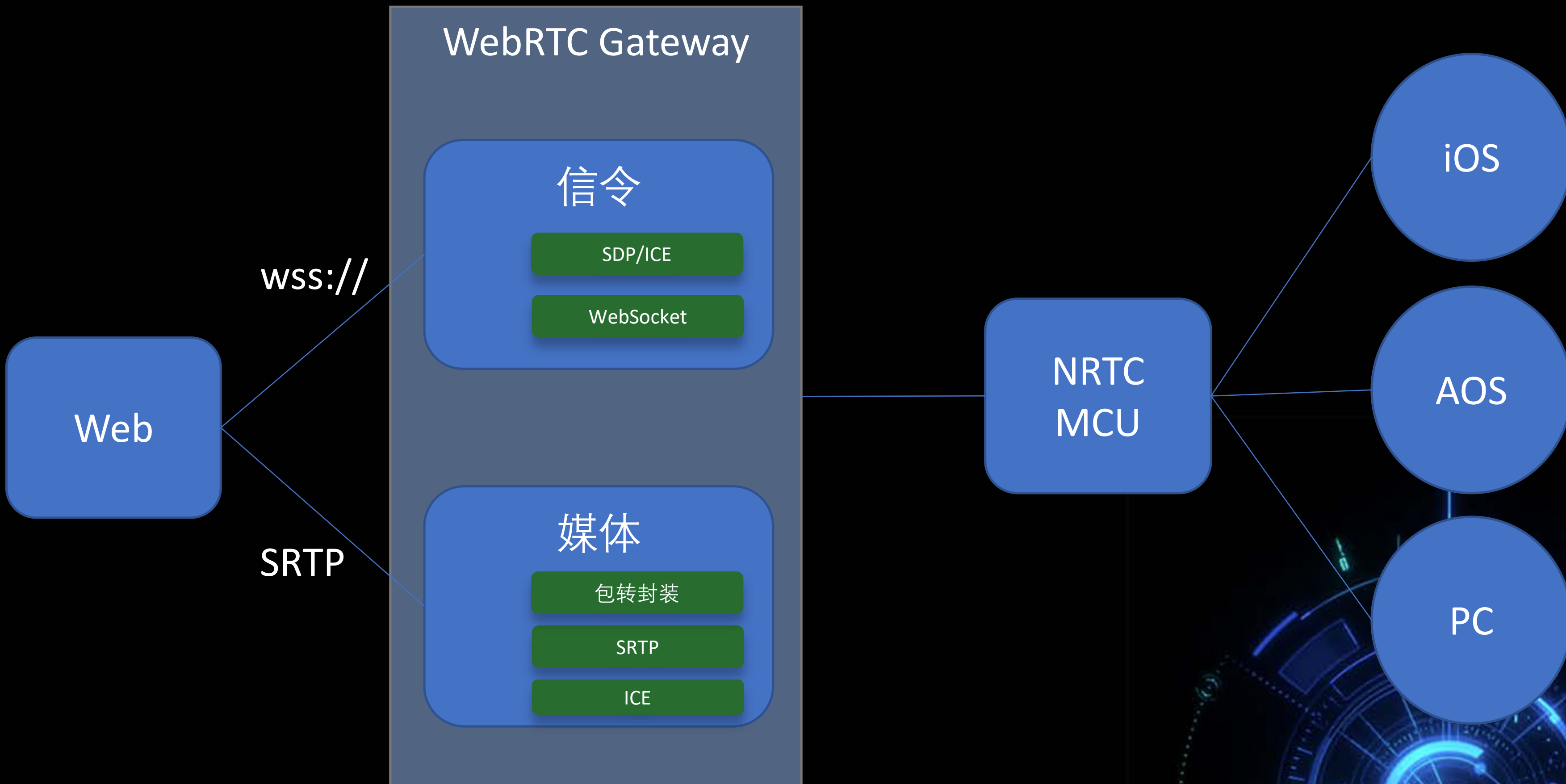
- NRTC早于WebRTC
- NRTC是VoIP的完整方案, WebRTC ≈ NRTC SDK
- NRTC的实现更灵活
- NRTC是工业级的实现

NRTC with WebRTC



WebRTC Gateway

网易云 | 通信与视频



LiveVideoStackCon

工业化

- Browser的兼容性
- Lite ICE
- RTCP feedbacks
- Reliable connections
- Congestion Control

Browser的各种坑

- browser兼容性 - adapter.js
- Video resolution
- MediaStream lifecycle
- getUserMedia call success, no media

ICE

- NAT
- STUN - RFC 5389
- TURN – RFC 5766
- ICE – RFC 5245
- TCP

Lite ICE

- 当通信一方是Server时，有公网IP
- Host candidates only
- Full peer发起连通检查，2步连通检查

网络监测

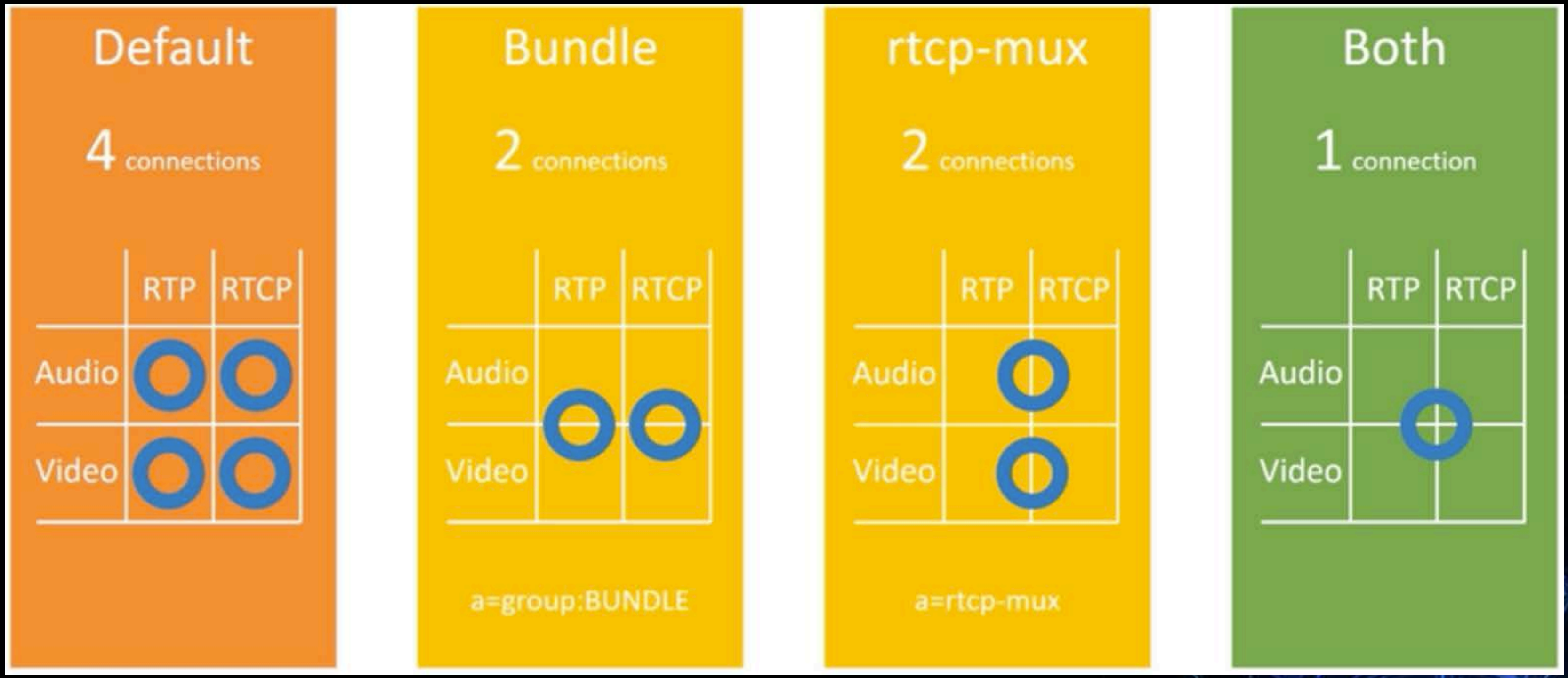
- WebSocket disconnect event
- RTCPeerConnection disconnect events
  - oniceconnectionstatechange
  - onsignalingstatechange
- keepalive over signaling channel

断开重连

- Start over
  - Detach stream, 销毁现有连接等
  - 信令连接、鉴权、媒体连接
- ICE restart

```
pc.createOffer({iceRestart: true})
.then(function(offer) {
    return pc.setLocalDescription(offer);
})
```

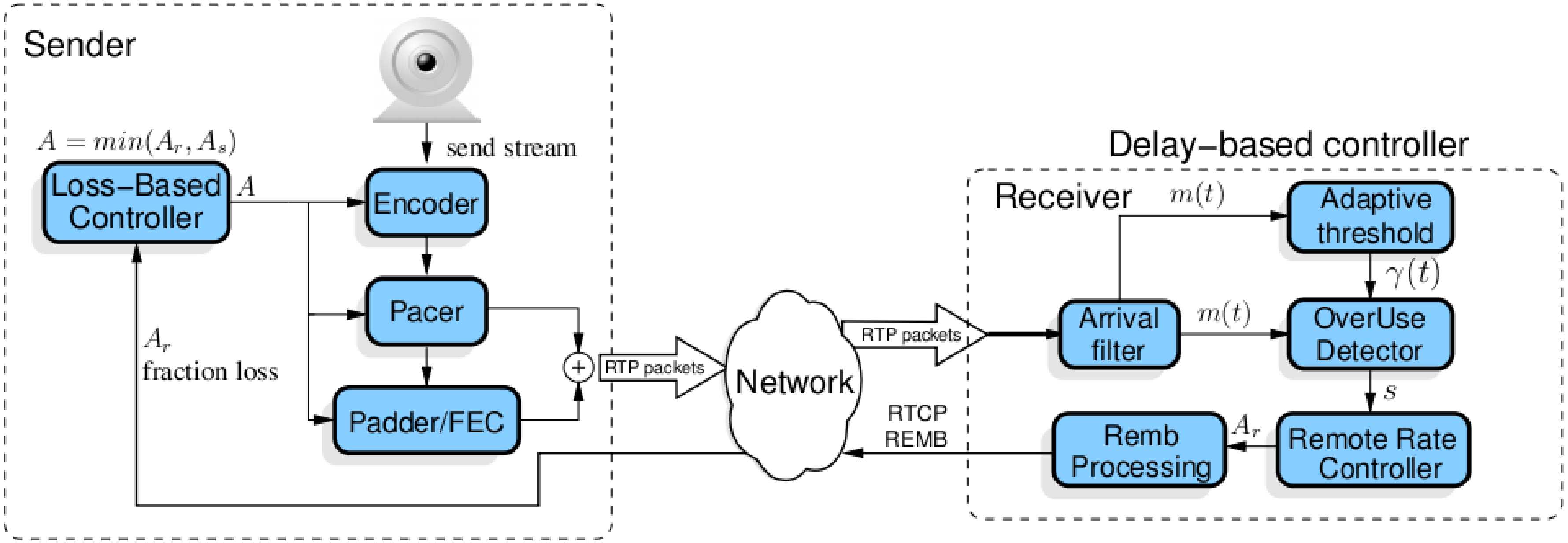
Multiplexing and bundle



FIR and PLI

- 请求关键帧

GCC


Sender
$A = min(A_r, A_s)$
Loss–Based Controller
A
send stream
Encoder
Pacer
Padder/FEC
A_r
fraction loss
RTP packets
Network
RTP packets
RTCP
REMB
Delay–based controller
Receiver
$m(t)$
Adaptive threshold
$\gamma(t)$
Arrival filter
$m(t)$
OverUse Detector
s
Remb Processing
A_r
Remote Rate Controller

REMB

- 接收端最大接收码率估测
- SDP:
  - a=rtcp-fb:107 goog-remb

GCC feedbacks

- Delay-based controller
  - Transport cc
- Loss-based controller
  - RTCP SR/RR

NACK

- RTCP feedback, RFC 4585
- Bi-direction retransmission
- SDP:
  - m=video 1234 RTP/SAVPF 107
  - a=rtpmap:107 H264/90000
  - a=rtcp-fb:107 nack
  - a=rtcp-fb:107 nack pli

一个SDP例子


```
v=0
o=nrtc 0 2 IN IP4 127.0.0.1
a=group:BUNDLE audio video
a=ice-ufrag:193055970146817/378951
a=ice-pwd:nfklsa
m=audio 1 RTP/SAVPF 111
c=IN IP4 0.0.0.0
a=rtcp:1 IN IP4 0.0.0.0
a=sendrecv
a=mid:audio
a=rtcp-mux
a=rtpmap:111 opus/48000/2
a=fmtp:111 maxplaybackrate=16000; sprop-maxcapturerate=16000;
a=fmtp:111 minptime=60;useinbandfec=1
a=setup:passive
a=maxptime:60
a=ptime:60
m=video 1 RTP/SAVPF 107
c=IN IP4 0.0.0.0
a=rtcp:1 IN IP4 0.0.0.0
a=sendrecv
a=mid:video
a=rtcp-mux
a=rtpmap:107 H264/90000
a=rtcp-fb:107 ccm fir
a=rtcp-fb:107 nack
a=rtcp-fb:107 nack pli
a=rtcp-fb:107 goog-remb
a=rtcp-fb:107 transport-cc
a=fmtp:107 level-asymmetry-allowed=1;packetization-mode=1;profile-level-id=42e01f
a=extmap:5 http://www.ietf.org/id/draft-holmer-rmcat-transport-wide-cc-extensions-01
a=setup:passive
```

Thank you