mini_sdp v1 方案

1. 背景

WebRTC 中的 sdp 用文本字符串表示，比较冗长（5-10KB 左右），不利于快速、高效传输，直播场景下，会尤其影响首帧时间。

本文兼顾扩展性（能完全表示 sdp 所有信息）和压缩性精细设计 mini_sdp，把 sdp 的文本压缩成高效传输的二进制（压缩到 300B 左右）、使其只需一个 udp 包即可交互成功。

本方案的核心优势是：高压缩率、强扩展性、使用便利性（对原生 WebRTC 侵入低）。

2. sdp 表示

2.1 整体结构

mini_sdp 由 mini_sdp header、session header、和 n 个 media 三部分组成。其具体结构描述见 2.2-2.5 描述。

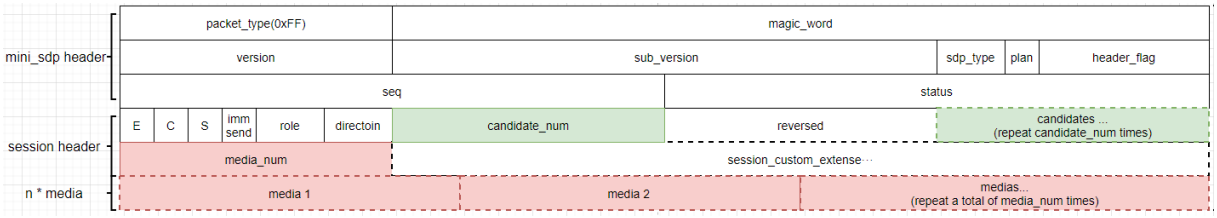

图 1

2.2 mini_sdp header 说明

如图 1 所示，mini_sdp header 为 mini_sdp 的头部，主要定义 mini_sdp 传输所需要的一些辅助信息及 sdp 的类型，各字段的长度及含义如下：

packet_type： 1B，固定为 0xFF，用于区分 rtp/rtcp/stun 包类型

magic_word： 3B，固定为“SDP”

version： 1B，版本号

sub_version： 2B，子版本号

sdp_type： 2b，0-offer, 1-answer

plan_type： 1b, 0-plan-b, 1-unifield-pan

header_flag： 5b, 保留位，如加压缩等标志

seq： 2B, 序列号，用于服务端去重

status： 2B，响应码，sdp_type 为 answer 时，status 表示不同的响应状态码

2.3 session header 说明

如图 1 所示，session header 主要定义 session 维度的一些信息，如是否加密，candidate，session 所包含的 media 数，及 session 扩展等，各字段的长度及含义如下：

E: 1b, 是否加密，0-不加密，1-加密

C： 1b，是否有 candidate，0-无，1-有

S: 1b，是否 bundle，0-无，1-有

immsend： 1b，是否打开 0-rtt 发包，0-打开，1-关闭

role： 2b，dtls role，0-actpass ，1-active ，2-passive

direction： 2b，session 方向如：0-sendonly ，1-receiveonly ，2-sendrecv

candidate_num：1B，candidate 个数

reversed： 1B，保留字

candidate： 子结构，candidate 描述

media_num: 1B, media 个数

session_custom_extense: 子结构（见 2.3.1），自定义长度 bitmap 及 key-value 映射

2.3.1 session_custom_extense 说明

如图 2 所示，扩展表采样类似 hpack 编码，包括位域静态码表和字符串动态码表，session_custom_extense 各字段的描述如下：

custom_ext_total_len: 1B，扩展总长度（含自身）

bit_map_size: 1B，自定义 bit_map 长度

custom_ext_str_len: 2B，key-value 扩展长度

custom_ext_str_id： 1B，key（uint8）

custom_ext_str: 自定义长度扩展 value（string）

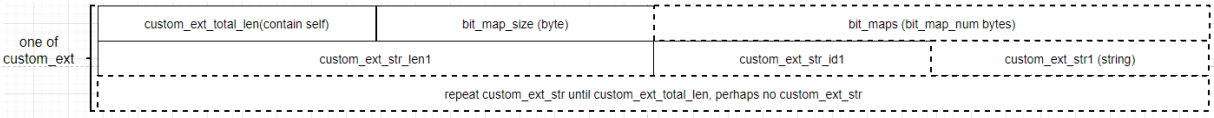

图 2

2.3.2 candidate 说明

ip_type: 1b, 0-ipv4，1-ipv6

candidate_flag: 7b, 保留位

candidate_port: 2B，端口

candidate_ip: 4B（ipv4）/16B（ipv6），

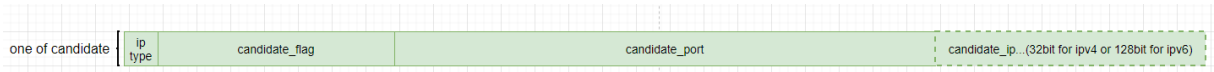

图 3

2.4 media 描述

如图 4 所示，media 描述了 sdp 流媒体信息，如媒体类型，codec、tracks、rtp 扩展等，各字段的描述如下：。

has_ext： 1b，是否有 media_custom_extense…扩展

track_num： 7b，track 数量

media_type： 2b，0-audio，1-video，2-datachannel

codec_num： 6b，codec 的数量

rtp_ext_num： 1B，rtp 扩展数量（id-url 的映射）

media_custom_extenses：子结构，自定义长度 bitmap 及 key-value 映射，结构同 **2.3.1**

tracks： 子结构（见 2.4.1），track 描述

codecs： 子结构（见 2.4.2），codecs 描述

rtp_extenses： 子结构（见 2.4.3），rtp 扩展描述

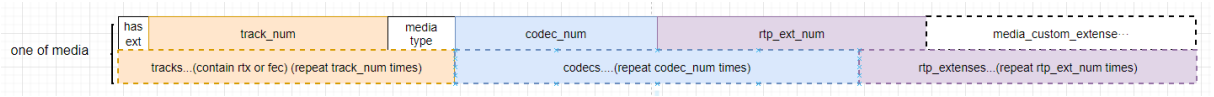

图 4

2.4.1 track 描述

ssrc: 4B

track_order: 1B

media_stream_id: 1B（id 对应的 string 在 custom_ext 中指明）

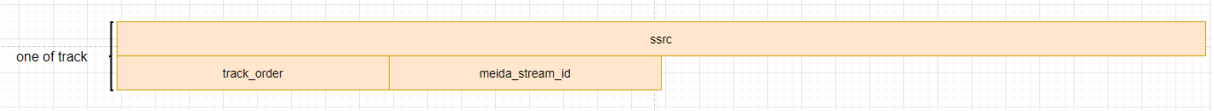

图 5

2.4.2 codecs 描述

如图 6 所示，codecs 结构对 codec 相关的描述进行了二进制压缩，并通过静态码表进行解析。

codec： 4b， 0-opus，1-MP4A-LATM，2-MP4A-ADTS，3-h264，4-h265

frequency: 4b， 0-44k，1-48k，2-90k

payload_type: 7b， 0-98，1-99， 2-100，3-101，4-102，5-103，6-104, 7-105, 8-106, 9-107...

has_ext: 1b，是否有 codec_custom_extense 扩展

channels： 2b，通道数

reversed： 6b，保留

codec_custom_extense 子结构（同 2.3.1 media_custom_extenses）

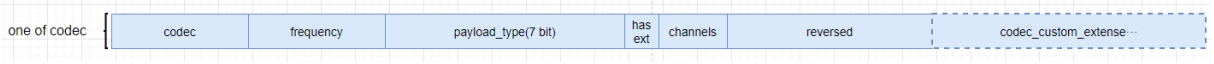

图 6

2.4.3 rtp_extenses 描述

rtp_ext_id: 1B

rtp_ext_url: 1B （具体 url 见码表）

图 7

2.5 扩展表

2.5.1 session 级别扩展

session 级别的扩展有两部分组成，位域静态扩展和字符串动态表扩展。位域扩展中用户可自定义每一位代表的含义；字符串扩展以 key-value 形式，用户可自定义变化的字符串，在 mini_sdp 结构中以 key 的 id 来代替，来进行去重。

bit_map:

str_map:

字段	扩展类型(uint8)	字段值类型	说明
ice_ufrag	0	string	--
ice_pwd	1	string	--
encrypt_key	2	string	加密 key
svr_sig	3	string	--
stream_url	4	string	--
auth_digest	5	string	--

2.5.2 media 级别扩展

media 扩展与 2.3.1 扩展结构相同，仅作用与本 media 范围内。

bit_map:

str_map:

字段	扩展类型(uint8)	字段值类型	说明
bitrate	0	uint32_t	码率控制

静态 rtp_ext_url 码表

rtp_ext_url	rtp_ext_url 对应 url
0	http://www.webrtc.org/experiments/rtp-hdrext/abs-send-time
1	http://www.webrtc.org/experiments/rtp-hdrext/playout-delay
2	http://www.ietf.org/id/draft-holmer-rmcat-transport-wide-cc-extensions-01
3	http://www.webrtc.org/experiments/rtp-hdrext/meta-data-01
4	http://www.webrtc.org/experiments/rtp-hdrext/meta-data-02
5	http://www.webrtc.org/experiments/rtp-hdrext/meta-data-03
6	http://www.webrtc.org/experiments/rtp-hdrext/decoding-timestamp
7	http://www.webrtc.org/experiments/rtp-hdrext/video-composition-time
8	http://www.webrtc.org/experiments/rtp-hdrext/video-frame-type

2.5.3 codec 级别扩展

codec 级扩展与 2.3.1 扩展结构相同，仅作用与本 media 范围内。

bit_map:

0=nack：是否支持 nack ，0-不支持，1-支持

1=flex_fec: 是否支持 flex_fec，0-不支持，1-支持

2=transport_cc: 是否支持 transport_cc，0-不支持，1-支持

3=remb：是否支持 remb，0-不支持，1-支持

4=bframe_enable: 是否支持 bframe_enable，0-不支持，1-支持

5=ps_enable: 音频 ftmp 选项，0-不支持，1-支持

6=sbr_enable: 音频 ftmp 选项，0-不支持，1-支持

7=stereo：音频 ftmp 选项，0-非立体声，1-立体声

8=cpresent: 音频 ftmp 选项，0-不支持，1-支持

9=useinbandfec: 音频 ftmp 选项

str_map:

字段	扩展类型(uint8)	字段值类型	说明
object	1	uint32_t	音频 ftmp 中 object 的值
config	2	string	aac config

3. 交互流程

sdp 转换流程：

标准 sdp——>mini_sdp——网络传输——mini_sdp——>标准 sdp

3.1 1-rtt 方案（中缓存）

1-rtt sdp（中缓存）交互方案如图 8 所示，首先将文本 sdp 压缩成 mini_sdp，然后通过多个冗余的 sdp 包发送给对端，对端收到 mimi_sdp 后，将 mini_sdp 还原为文本 sdp，生成文本 answer sdp，再将文本 answer 压缩成 mini_sdp，并通过 udp 返回给对端。sdp 交互完以后再进行 stun 握手校验，stun 交互完以后，向对端发送 rtp 数据包。同时通过 rtcp，将认证串（认证串包含客户端和服务端信息，某一认证串只能由该客户端使用，其他客户端使用无效）返回给对端。

对端第二次信令交互即可走 3.3 0-rtt 方案。

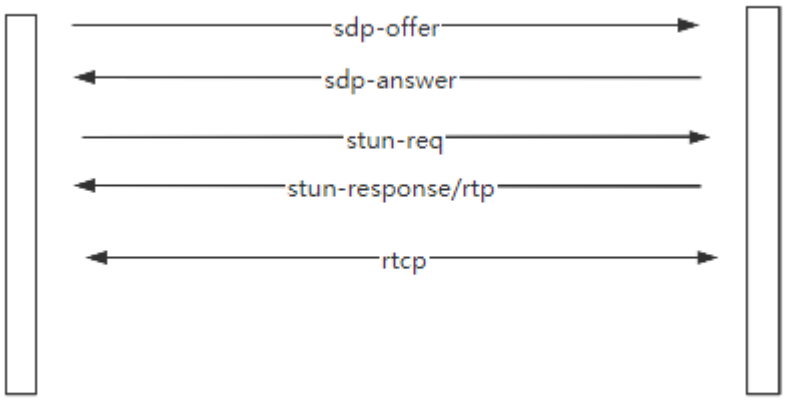

图 8

3.2 1-rtt 方案（不中缓存）

1-rtt sdp（不中缓存-异步回源）交互方案如图 9 所示，在收到 sdp-offer 及认证串校验成功时，立即异步回源，同时回 answer，并再次异步校验 stun，在收到回源数据时，通过 rtcp 更新 aac config。然后发送 rtp/rtcp。

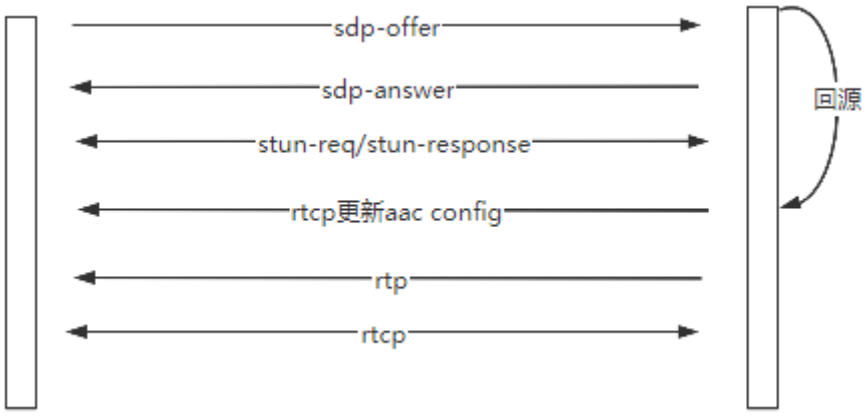

图 9

1-rtt sdp（不中缓存-同步回源）交互方案如图 9 所示，在收到 sdp-offer 及认证串校验失败时，则先回 answer，然后等待 stun 握手校验完成，再发起回源，在收到回源数据时，再次更新 sdp。然后发送 rtp/rtcp。

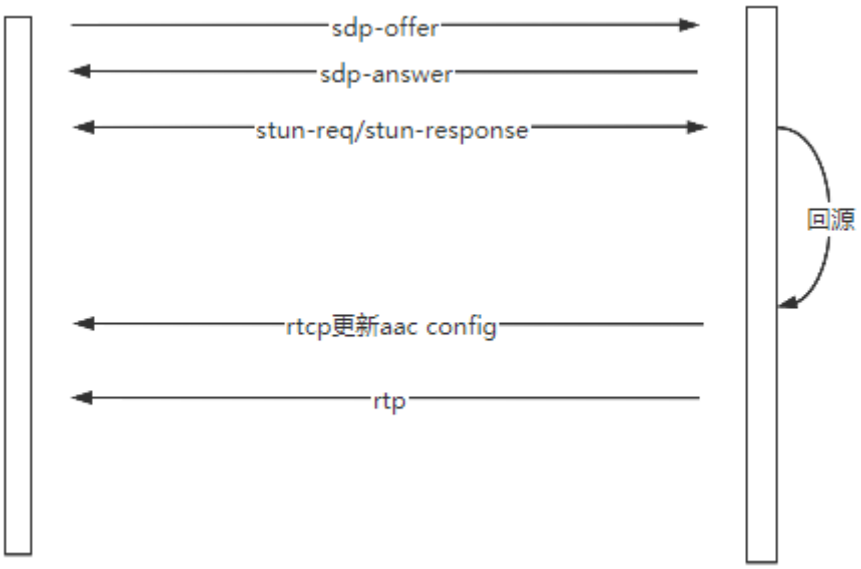

图 10

3.3 0-rtt 方案（中缓存）

0-rtt sdp 交互方案如图 11 所示，首先将文本 sdp 压缩成 mini_sdp，mimi_sdp 带上认证串，然后通过多个冗余的 sdp 包发送给对端，对端收到 mimi_sdp 后，将 mini_sdp 还原为文本 sdp，生成文本 answer sdp，再将文本 answer 压缩成 mini_sdp，并通过 udp 返回给对端，认证串校验成功则同时发送 rtp 数据包；同时异步进行 stun 校验，在一定时间内 stun 异常，则断掉 peerconnection，回收该 peerconnection 所有资源。

若认证串校验失败，则退化为 1-rtt 方案，等待 stun 握手校验完毕，再向对端发送 rtp 包

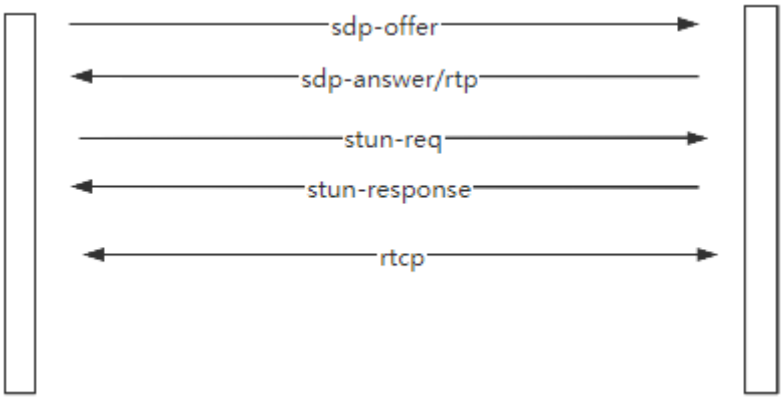

图 11