

Data	Notes	Author
2017-10-26	draft version	lichen
2017-10-31	remove redundant transport info for addTrack	lichen
2017-11-08	add suspend and resume	lichen

控制交互.....	4
常规格式.....	4
特殊格式.....	4
交互方式.....	4
请求对象.....	5
响应对象.....	5
通知对象.....	6
批量调用.....	6
示例.....	7
参数说明.....	9
消息详述.....	13
设置系统/会议全局参数消息(setParam).....	14
设置通道对象参数消息响应.....	14
通道运行状态控制(runctrl).....	15
通道运行状态控制响应.....	15
分配传输端口资源(addPort).....	16
分配传输端口资源响应.....	16
释放传输端口资源(delPort).....	17
释放传输端口资源响应.....	17
更新 Candidate(updateCand).....	18
更新本地 Candidate 响应.....	18
设置传输端口参数(setPortParam).....	19

设置传输端口参数响应.....	19
分配媒体轨道资源(addTrack).....	20
分配媒体轨道资源响应.....	21
释放媒体轨道资源(delTrack).....	22
释放媒体轨道资源响应.....	22
设置 mute/unmute(muteCtrl).....	23
设置 mute/unmute 响应.....	23
Dtmf 检测反馈(dtmfFB).....	24
Dtmf 检测反馈响应.....	24
重启 ICE 流程(restartICE).....	25
重启 ICE 流程响应.....	25
转发通道控制(sfuChanCtrl).....	26
转发通道控制响应.....	26
AVS&SignalService 基本交互.....	27

控制交互

常规格式

AVS 对外受控接口采用 HTTP+JSON (RFC 4627) 的方式，即 HTTP 的消息体中必须为一个有效的 JSON 结构。

其他约定规则：

- AVS 监听端口为 8088 或采用 stream UNIX Socket(/tmp/avsbinder)
- 采用长连接方式（因 avs 需要上报一些信息）
- HTTP 头部应包含 Connection: keep-alive。
- Content-Type 应包含 application/json
- Content-Length 应正确填写（不能省略）
- 采用 POST 方式
- url 采用 /com/grandstream/httpjson/avs

特殊格式

HTTP+binary data 的方式，具体使用待定。

交互方式

- 发送请求对象，并超时等待 AVS 反馈的响应对象（如果超时则认为执行失败）
- 发送通知对象，无需等待反馈（AVS 也不进行反馈）

默认都以请求对象的方式进行交互（即需要等待反馈）

请求对象

一个请求对象包含下列成员：

" message statusname "：

包含某请求对应消息名称的字符串

" message params "：

某消息所需要的结构化参数值

id：

某次请求的唯一标识 id, 值必须包含一个字符串。如果不包含该成员则被认定为是一个通知。

响应对象

响应表示为一个 JSON 对象，使用以下成员

error：

某请求执行成功与否及相关错误描述

(
code: 0:SUCCESS -1:FAILURE
message: 对该错误的简单描述字符串
)

resDescribe：

其他额外的结果反馈（根据具体的请求对象而异）

id:

该成员必须包含。

该成员值必须与请求对象中的 id 成员值一致。

通知对象

没有包含"id"成员的请求对象为通知， 作为通知的请求对象表明调用者对相应的响应对象并不感兴趣，本身也没有响应对象需要返回给调用者。AVS 不回复一个通知，包含那些批量请求中的通知对象。

由于通知没有返回的响应对象，所以通知不确定是否被执行。同样，调用者不会意识到任何错误（例如参数缺省，内部错误）。

批量调用

当需要同时发送多个请求对象时，可以发送一个包含所有请求对象的数组。

当批量调用的所有请求对象处理完成时，AVS 则需要返回一个包含相对应的响应对象数组。

每个响应对象都应对应每个请求对象，除非是通知的请求对象。调用者可以并发的，以任意顺序和任意宽度的并行性来处理这些批量调用。

这些相应的响应对象可以任意顺序的包含在返回的数组中，而调用者应该是基于各个响应对象中的 id 成员来匹配对应的对象请求。

若批量调用的请求操作本身为一个非法的请求（含无效 JSON），则 AVS 返回的将单单是一个响应对象而非数组。若批量调用没有需要返回的响应对象，则 AVS 不需要返回任何结果

给调用者。

示例

Syntax:

--> request

<-- response

一般调用：

```
--> {"runctrl": {"conf_id":"0","chan_id":"0","opt":"start"},"id": "1"}
<-- {"error":{"code":0}, "id": "1"}
```

包含无效 json 的调用：

```
--> {"runctrl": {"conf_id":"0","chan_id":"0","opt":"start"},"id": "1", "baz"]
<-- {"error": {"code": -1, "message": "Parse error"}, "id": "1"}
```

包含无效请求对象的调用：

```
--> {"abc": {"conf_id":"0","chan_id":"0","opt":"start"},"id": "1"}
<-- {"error": {"code": -1, "message": "Invalid Request"}, "id": "1"}
```

包含无效 json 的批量调用：

```
--> [
    {"runctrl": {"conf_id":"0","chan_id":"0","opt":"start"},"id": "1"},
    {"abc"}
]
<-- {"error": {"code": -1, "message": "Parse error"}, "id": "1"}
```

包含空数组的调用：

```
--> []
<-- {"error": {"code": -1, "message": "Invalid Request"}, "id": "1"}
```

非空且无效的批量调用：

```
--> [1]
<-- [
```

```

    { "error": { "code": -1, "message": "Invalid Request"}, "id": "1"}
  ]

```

无效的批量调用:

```

--> [1,2,3]
<-- [
  { "error": { "code": -1, "message": "Invalid Request"}, "id": "1"},
  { "error": { "code": -1, "message": "Invalid Request"}, "id": "1"},
  { "error": { "code": -1, "message": "Invalid Request"}, "id": "1"}
]

```

批量调用:

```

-->[
  {"runctrl": {"conf_id":"0","chan_id":"0","opt":"start"},"id": "1"},
  {"runctrl": {"conf_id":"0","chan_id":"1","opt":"start"},"id": "2"},
  {"runctrl": {"conf_id":"0","chan_id":"2","opt":"start"},"id": "3"},
  {"runctrl": {"conf_id":"0","chan_id":"3","opt":"start"},"id": "4"}
]
<-- [
  { "error":{"code":0}, "id": "1"},

  { "error":{"code":0}, "id": "2"},

  { "error":{"code":0}, "id": "3"},

  { "error":{"code":0}, "id": "4"},

]

```

所有都为通知的批量调用:

```

--> [
  {"method": "notify_sum", "params": [1,2,4]},
  {"method": "notify_hello", "params": [7]}
]

```

```

<-- //Nothing is returned for all notification batches

```


参数说明

token	token_value	on the fly	Tx/Rx	主要用途
MainCoder	"video/avc" ; <u>for H.264</u> "video/hevc" ; <u>for H.265</u> "video/vp8" ; <u>for VP8</u> "video/vp9" ; <u>for VP9</u> "audio/pcmu"; <u>for G.711 U-LAW</u> "audio/pcma"; <u>for G.711 A-LAW</u> "audio/ilbc" ; "audio/g722" ; <u>for G.722</u> "audio/adpcm32" ; <u>for G.726</u> "audio/g729" ; "audio/opus" ; "audio/g722.1" ; "audio/g722.1c" ; "audio/g723.1" ; "audio/gsm" ;	N	Tx	编码器类型
Codecs	"video/avc" ; <u>for H.264</u> "video/hevc" ; <u>for H.265</u> "video/vp8" ; <u>for VP8</u> "video/vp9" ; <u>for VP9</u> "audio/pcmu"; <u>for G.711 U-LAW</u> "audio/pcma"; <u>for G.711 A-LAW</u> "audio/ilbc" ; "audio/g722" ; <u>for G.722</u> "audio/adpcm32" ; <u>for G.726</u> "audio/g729" ; "audio/opus" ; "audio/g722.1" ; "audio/g722.1c" ; "audio/g723.1" ; "audio/gsm" ;	N	Rx	解码器类型
CN	cn = "0" <u>for switch off</u> "1" <u>for switch on</u>	N	Tx	开启 VAD/CNG/DTX 传输机制
Ptime	1*DIGIT <u>for time in ms</u>	N	Tx	音频编码 RTP 包时长
BitRate	rate (bps)	N	Tx	媒体编码码率

FrameRate	1*DIGIT	N	Tx/Rx	视频帧率
GopLength	Framerate~65535 or -1	N	Tx	-1: 无周期性的 IDR
ImageSize	1*DIGIT "x" 1*DIGIT	N	Tx/Rx	视频尺寸
BindPort	port = 1*DIGIT	N	Rx	提供解码数据到达端口
TargetAddr	ip ":" port	N	Tx	提供编码数据发送的目标地址和端口
QoS	qos = ("audio/" / "video/") 2*2HEXDIG	N	-	传输的 QoS 设置: 两位 16 进制
SymRTP	"0" for switch off "1" for switch on	N	-	对称 RTP 开关
RtcpMux	"0" no-mux "1" mux	N	-	rtp,rtcp 是否共用端口
TransMode	"sendRecv" "sendOnly" "recvOnly"	N	-	传输模式
PacketizationMode	"0" for Single NAL Unit Mode "1" for Non-Interleaved Mode "2" for Interleaved Mode	N	-	AVC 的 NAL 分包方式
SrtpMode	"2" for AES256 CM SHA1 80 "3" for AES256 CM SHA1 32 "4" for AES128 CM SHA1 80 "5" for AES128 CM SHA1 32	N	-	Srtp 加密方式
SrtpSendKey	hexadecimal string	N	Tx	加密所需 Key
SrtpRecvKey	hexadecimal string	N	Rx	解码所需 Key
MuteVoip	mute_val = "0" for un-mute "1" for mute	Y	-	会议成员静音开关
candidate	["candidate-attribute0", "candidate-attribute1", ...]	Y	Tx/Rx	候选地址信息 RFC5245
fingerprint	fingerprint-attribute = "fingerprint" ":" hash-func SP fingerprint	N	Tx/Rx	DTLS 指纹信息 RFC4572
ice_ufrag	contain at least 128 bits of randomness(at least 4 characters long)	N	Tx/Rx	Ice credentials
ice_pwd	contain at least 128 bits of	N	Tx/Rx	Ice credentials

	randomness(at least 22 characters long)			
Nack	0: 表示不启用 nack; 1:表示启用 nack	N	Tx/Rx	
PayloadType	pt =status 1*DIGIT	N	Tx/Rx	RTP 码流载荷类型
SSRC	10 进制, 32bits 的整数	N	Tx/Rx	媒体流对应的 SSRC -1: 随机生成或自检测或无效
FecPT	pt = 1*DIGIT "-1"禁用 FEC "0~127"FEC 有效 payload 值	N	Tx/Rx	FEC 载荷类型
RedPT	pt = 1*DIGIT "-1"禁用 RED "0~127"RED 有效 payload 值 一路 RED/FEC/Media pt 不能重复	N	Tx/Rx	RED 载荷类型
FecSSRC	10 进制, 32bits 的整数		Tx/Rx	FEC 流对应的 SSRC -1: 随机生成或自检测或无效
FecType	"0" rs_fec "1" ulpfec	N	Tx/Rx	FEC 类型识别
RtxPT	10 进制, 不超过 127 的整数	N	Tx/Rx	RTX 载荷类型
RtxSSRC	10 进制, 32bits 的整数	N	Tx/Rx	RTX 流对应的 SSRC -1: 随机生成或自检测或无效
mohFile	对应文件的完整路径	N	-	
ssrc_audio_level	"0" enable "1" disable	N	Rx	Client-to-Mixer Audio Level Indication RFC6464
IceRole	"0" controlling "1" controlled	N	-	
SslRole	"0" ssl client "1" ssl server	N	-	

```

candidate-attribute      = "candidate" ":" foundation SP component-id SP
                           transport SP
                           priority SP
                           connection-address SP      ;from RFC 4566
                           port                        ;port from RFC 4566
                           SP cand-type
                           [SP rel-addr]
                           [SP rel-port]
                           *(SP extension-att-name SP
                              extension-att-value)

foundation                = 1*32ice-char
component-id              = 1*5DIGIT
transport                 = "UDP" / transport-extension
transport-extension       = token                      ; from RFC 3261
priority                  = 1*10DIGIT
cand-type                 = "typ" SP candidate-types
candidate-types           = "host" / "srflx" / "prflx" / "relay" / token
rel-addr                  = "raddr" SP connection-address
rel-port                  = "rport" SP port
extension-att-name        = byte-string                ;from RFC 4566
extension-att-value       = byte-string
ice-char                  = ALPHA / DIGIT / "+" / "/"

```

消息详述

message name	message params	usage	comment
setParam	STURN/TURN Server Info MOH File(ConfID)等	Request/Response	全局系统参数/会议全局 参数设置
runctrl	conf_id/chan_id/opt 等	Request/Response	通道运行状态控制
addPort	conf_id/chan_id/ICE/DTLS 等	Request/Response	分配传输端口资源
delPort	conf_id/chan_id/port_id 等	Request/Response	释放传输端口资源
updateCand	port_id/candidate	Request/Response	更新 Candidate
setPortParam	conf_id/chan_id/port_id/IceRole 等	Request/Response	设置传输端口参数
addTrack	conf_id/chan_id/track_id 等	Request/Response	分配媒体轨道资源
delTrack	conf_id/chan_id/track_id 等	Request/Response	释放媒体轨道资源
muteCtrl	conf_id/chan_id/MuteVoip 等	Request/Response	设置 mute/unmute
dtmfFB	conf_id/chan_id/dtmf 等	Request/Response	Dtmf 检测反馈
restartICE	conf_id/chan_id/port_id 等	Request/Response	重启 ICE 流程
sfuChanCtrl	conf_id/chan_id/src_chan_id 等	Request/Response	转发通道控制

设置系统/会议全局参数消息(setParam)

key	M/O	value	sub key	M/O	comment
stunserver	O	Object(list)	address	M	
			port	M	
turnserver	O	Object(list)	address	M	
			port	M	
			username	M	
			password	M	
			protocoltype	O	
MOHInfo	O	Object	conf_id	M	
			mohFile	M	

例如：

```
{
  "setParam": {
    "stunserver": [ {
      "address": "192.168.12.11",
      "port": "1234"
    } ],
  },
  "id": "1"
}
```

设置通道对象参数消息响应

key	M/O	value	sub key	M/O	comment
error	M	Object	code	M	错误代号
			message	O	错误描述

例如：

```
{"error": {
  "code": 0,
  "message": null},
  "id": "1"}
```

通道运行状态控制(runctrl)

key	M/O	value	comment
conf_id	M	唯一字符串	会议编号
chan_id	M	唯一字符串	通道编号
opt	M	"start" or "reset" or "suspend" or "resume" or	"start":开启通道 "reset":释放通道 "suspend":暂停通道 "resume":恢复通道
mediaType	O	"audio" or "video" or "all"	只在"suspend" or "resume" 时有效

例如：

```
{ "runctrl": {
  "conf_id": "0",
  "chan_id": "0",
  "opt": "start"},
  "id": "1" }
```

通道运行状态控制响应

key	M/O	value	sub key	M/O
error	M	Object	code	M
			message	O

分配传输端口资源(addPort)

key	M/O	value	comment
conf_id	M	唯一字符串	
chan_id	M	唯一字符串	
ICE	M	"0" enable "1" disable	是否启用 ICE 流程
DTLS	M	"0" enable "1" disable	是否启用 DTLS

例如：

```
{
  "addPort": {
    "conf_id": "0",
    "chan_id": "0",
    "ICE": "1",
    "DTLS": "1"
  },
  "id": "1"
}
```

分配传输端口资源响应

key	M/O	value	sub key	M/O	comment
error	M	Object	code	M	
			message	O	
port_id	M	唯一字符串	-	-	
InfoICE	O	Object	candidate	M	List 类型
			fingerprint	M	
			ice_ufrag	M	
			ice_pwd	M	
InfoPort		Object	rtp_port	M	
			rtcp_port	M	
			fingerprint	O	

释放传输端口资源(delPort)

key	M/O	value	comment
conf_id	M	唯一字符串	会议编号
chan_id	M	唯一字符串	通道编号
port_id	M	唯一字符串	传输端口资源编号

例如：

```
{
  "delPort": {
    "conf_id": "0",
    "chan_id": "0",
    "port_id": "0"
  },
  "id": "1"
}
```

释放传输端口资源响应

key	M/O	value	sub key	M/O
error	M	Object	code	M
			message	O

更新 Candidate(updateCand)

key	M/O	value	comment
conf_id	M	唯一字符串	会议编号
chan_id	M	唯一字符串	通道编号
port_id	M	唯一字符串	输出端口资源
type	M	"0" local "1" remote	
candidate	M		List 类型

例如：

```
{
  "updateLCand": {
    "conf_id": "0",
    "chan_id": "0",
    "port_id": "0",
    "type": "0",
    "candidate": [
      "candidate:1552544681 1 udp 2122194687 192.168.127.33 47390 typ host generation 0 ufrag ILZw network-id 1 network-cost 50",
      "candidate:1552544681 1 udp 2122194687 192.168.127.33 43077 typ host generation 0 ufrag ILZw network-id 1 network-cost 50"
    ]
  },
  "id": "1"
}
```

更新本地 Candidate 响应

key	M/O	value	sub key	M/O
error	M	Object	code	M
			message	O

设置传输端口参数(setPortParam)

key	M/O	value	sub key	M/O	comment
conf_id	M	唯一字符串			
chan_id	M	唯一字符串			
port_id	M	唯一字符串			
InfoICE	O	Object	IceRole	O	
			SslRole	O	
			fingerprint	O	
			ice_ufrag	O	
			ice_pwd	O	
			candidate	O	List 类型
InfoPort	O	Object	targetAddr	O	
			RtcpMux	O	
			SymRTP	O	
			Qos	O	
			srtpMode	O	
			srtpSendKey	O	
			srtpRecvKey	O	
			fingerprint	O	

例如：

设置传输端口参数响应

key	M/O	value	sub key	M/O
error	M	Object	code	M
			message	O

分配媒体轨道资源(addTrack)

key	M/O	value	sub key	M/O
conf_id	M	唯一字符串		
conf_id	M	唯一字符串		
port_id	M	唯一字符串		
track_id	M	唯一字符串		
mediaType	M	"audio" or "video" or "content"		
audio_tx_param	O	Object	MainCoder	M
			BitRate	O
			PayloadType	M
			SSRC	O
			Ptime	M
			CN	O
			FecPT	O
			FecSSRC	O
audio_rx_param	O	Object	Codecs	M
			PayloadType	M
			SSRC	O
			FecPT	O
			FecSSRC	O
			ssrc_audio_level	O
audio_transport	O	Object	TransMode	O
video_tx_param	O	Object	MainCoder	M
			ImageSize	O
			FrameRate	O
			Profile	O
			BitRate	O
			PacketizationMode	O
			PayloadType	M
			SSRC	O
			FecPT	O
			RedPT	O

			FecType	O
			FecSSRC	O
			RtxPT	O
			RtxSSRC	O
			Nack	O
video_rx_param	O	Object	Codecs	M
			ImageSize	O
			FrameRate	O
			PayloadType	M
			SSRC	O
			FecPT	O
			RedPT	O
			FecType	O
			FecSSRC	O
			RtxPT	O
			RtxSSRC	O
			RtxSSRC	O
			WaitKeyframe	O
			Nack	O
video_transport	O	Object	TransMode	O

例如：

```
{
  "addTrack": {
    "conf_id": "0",
  },
  "id": "1"
}
```

分配媒体轨道资源响应

key	M/O	value	sub key	M/O
error	M	Object	code	M
			message	O

释放媒体轨道资源(delTrack)

key	M/O	value	comment
conf_id	M	唯一字符串	会议编号
conf_id	M	唯一字符串	会议编号
track_id	M	唯一字符串	媒体轨道编号

例如：

```
{
  "delTrack": {
    "conf_id": "0"
  },
  "id": "1"
}
```

释放媒体轨道资源响应

key	M/O	value	sub key	M/O
error	M	Object	code	M
			message	O

设置 mute/unmute(muteCtrl)

key	M/O	value	comment
conf_id	M	唯一字符串	会议编号
chan_id	M	唯一字符串	通道编号
MuteVoip	M	"0" for un-mute "1" for mute	

```

{"muteCtrl":{
  "conf_id":"0",
  "chan_id":"0",
  "MuteVoip":"1"
},
"id":"2"}

```

设置 mute/unmute 响应

key	M/O	value	sub key	M/O
error	M	Object	code	M
			message	O

Dtmf 检测反馈(dtmfFB)

key	M/O	value	comment
conf_id	M	唯一字符串	会议编号
chan_id	M	唯一字符串	通道编号
dtmf	M		

```
{
  "conf_id": "0",
  "chan_id": "0",
  "dtmf": "123"
},
{id": "2"}
```

Dtmf 检测反馈响应

key	M/O	value	sub key	M/O
error	M	Object	code	M
			message	O

重启 ICE 流程(restartICE)

key	M/O	value	sub key	M/O	comment
conf_id	M	唯一字符串	-	-	
chan_id	M	唯一字符串	-	-	
port_id	M	唯一字符串	-	-	
type	M	"0" local trigger "1" peer trigger	-	-	
InfoICE	O	Object	candidate	O	List 类型
			ice_ufrag	M	
			ice_pwd	M	

```
{
  "conf_id": "0",
  "chan_id": "0",
  "port_id": "0",
  "type": "0",
  "id": "2"
}
```

重启 ICE 流程响应

key	M/O	value	sub key	M/O	comment
error	M	Object	code	M	
			message	O	
port_id	M	唯一字符串	-	-	
InfoICE	O	Object	candidate	O	List 类型
			ice_ufrag	M	
			ice_pwd	M	

转发通道控制(sfuChanCtrl)

key	M/O	value	sub key	sub value	comment
conf_id	M	唯一字符串	-	-	
chan_id	M	唯一字符串	-	-	
ope	M	"suspend" or "resume" or "reset" or	-	-	suspend:暂停转发 resume:恢复转发 reset:删除转发
sfuChan	M	Object(list)	src_chan_id	唯一字符串	待转发的通道编号
			mediaType	"video" or "content" or "all"	video:主视频 content:演示视频 all:所有视频

```
{
  "sfuChanCtrl": {
    "conf_id": "0",
    "chan_id": "0",
    "ope": "reset",
    "sfuChan": [
      {
        "src_chan_id": "chan8",
        "mediaType": "all"
      }
    ]
  },
  "id": "2"
}
```

转发通道控制响应

key	M/O	value	sub key	M/O
error	M	Object	code	M
			message	O

AVS&SignalService 基本交互



