

# 10. - SIP SDP - Session Description Protocol

## SIP SDP

SIP SDP stands for Session Description Protocol and is a protocol used to describe multimedia sessions for the purposes of session initiation, control, and management. In the context of SIP, SDP is typically used in the body of SIP messages to describe the characteristics of media streams being offered or negotiated in a SIP session. This includes information such as the type of media, codec, bandwidth, and IP addresses and ports for the media streams. SDP is an important part of the SIP protocol because it allows endpoints to negotiate and establish media sessions for audio, video, and other types of multimedia content.

There are a lot of fields available in SDP and some are mandatory (shown here in blue). New fields can be added as needed. If an agent doesn't understand a field it simply ignores it.

<b>v=</b> (protocol version)	<b>t=</b> (time the session is active)
<b>o=</b> (owner/creator and session identifier).	<b>r=</b> (zero or more repeat times)
<b>s=</b> (session name)	<b>m=</b> (media name and transport address)
<b>i=</b> (session information)	
<b>u=</b> (URI of description)	
<b>e=</b> (email address)	
<b>p=</b> (phone number)	
<b>c=</b> (connection information)	
<b>b=</b> (bandwidth information)	
<b>a=</b> (zero or more media attribute lines)	
<b>z=</b> (time zone adjustments)	
<b>k=</b> (encryption key)	

# MIME

MIME stands for Multipurpose Internet Mail Extensions, which is a standard that extends the format of email messages to support text in character sets other than ASCII, as well as attachments of audio, video, images, and application programs. In the context of SIP SDP (Session Description Protocol), MIME is used to specify the media type and format of the SDP body in the SIP message. The MIME type for SDP is "application/sdp". The SDP body contains information about the media streams to be used for the session, such as the codecs, transport protocols, and media formats to be used. The MIME type allows the receiving endpoint to determine how to handle the SDP body in the SIP message.

```
INVITE sip:bob@biloxi.com SIP/2.0
Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bK776asdhds
Max-Forwards: 70
To: Bob <sip:bob@biloxi.com>
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710
CSeq: 314159 INVITE
Contact: <sip:alice@pc33.atlanta.com>
User-Agent: BriaX 3.5.5 build 81887
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, NOTIFY, MESSAGE, SUBSCRIBE, INFO
Allow-Events: presence, message-summary, refer, dialog
Session-Expires: 3600;refresher=uas
Content-Type: application/sdp
Content-Length: 283

v=0
o=- 123456 789012 IN IP4 pc33.atlanta.com
s=SIP Call
c=IN IP4 192.0.2.1
t=0 0
m=audio 49170 RTP/AVP 0 8 18 3 101
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
```

```
a=rtpmap:18 G729/8000
a=fmtp:18 annexb=no
a=rtpmap:3 GSM/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
m=video 49172 RTP/AVP 31 34
a=rtpmap:31 H261/90000
a=rtpmap:34 H263/90000
```

In this example, the header includes the following fields:

### Expand to view Header description

- **Via:** Indicates the transport protocol, IP address, and port number used for the request. In this case, it is a UDP transport protocol and the IP address is pc33.atlanta.com.
- **Max-Forwards:** Indicates the maximum number of times the request can be forwarded before it is discarded.
- **To:** Contains the display name and SIP URI of the destination user.
- **From:** Contains the display name and SIP URI of the source user, along with a unique identifier (tag) for the request.
- **Call-ID:** Unique identifier for the call.
- **CSeq:** Sequence number for the request.
- **Contact:** Indicates the SIP URI that can be used to reach the sender of the request.
- **User-Agent:** Indicates the user agent that generated the request.
- **Allow:** Lists the SIP methods that the user agent is able to handle.
- **Allow-Events:** Lists the types of events that the user agent can generate.
- **Session-Expires:** Indicates the duration of the session in seconds and the refresher type (in this case, uas).

The SDP message in the body of the request describes the media capabilities of the sender and includes the following information:

- **v** (Protocol version): Indicates the protocol version of the session description. In this case, it is 0.
- **o** (Origin): Specifies the originator of the session and a unique identifier for the session. The fields include username, session identifier, session version number, and network type (IN for Internet) and address type (IP4 for IPv6).  
O=<username> <sess-id> <sess-version> <nettype> <addrtype> <unicast-address>

The nettype and addrtype fields can take various values depending on the network protocol being used. Some of the options for the nettype field include:

- IN: Internet
- AT: Atmosphere
- FI: FIP Snooping
- Token-Ring: IBM Token Ring Network

Some of the options for the addrtype field include:

- IP4: IPv4 address
- IP6: IPv6 address
- MAC: MAC address
- NSAP: Network Service Access Point
- E.164: E.164 telephone number
- **s** (Session name): Specifies a human-readable session name.
- **c** (Connection information): Specifies the connection information for the session. In this case, it is the IP address 192.0.2.1 - This field specifies the network address type IP4 or IP6 and the connection address IP Address or hostname of the media stream RTP, there are different types of connection information fields:
  - c=IN IP4 192.0.2.1: This specifies that the media stream will use IPv4 and the IP address is 192.0.2.1
  - c=IN IP6 ::1: This specifies that the media stream will use IPv6 and the IP address is ::1 (loopback address).
  - c=IN IP4 192.168.1.100/255: This specifies that the media stream will use IPv4 and the IP address is 192.168.1.100, with a subnet mask of 255.255.255.0.
- **t** (Timing): Specifies the start and stop times of the session. In this case, it is set to 0, which means the session is not limited by time.
- **m** (Media): Specifies the media type (audio), the port number (1234), and the transport protocol (RTP/AVP). There can be multiple media fields in an SDP message.
  - m=audio 49170 RTP/AVP 0 8: This specifies an audio stream on port 49170 using RTP/AVP protocol, with two codecs: PCMU (payload type 0) and PCMA (payload type 8).
  - m=video 51372 RTP/AVP 31 34: This specifies a video stream on port 51372 using RTP/AVP protocol, with two codecs: H.263 (payload type 31) and H.264 (payload type 34).
- **a** (Attribute): Specifies additional attributes for the session. In this case, it specifies the RTP payload type for each codec (0 for PCMU, 8 for PCMA, and 101 for telephone-event), the codec names (PCMU, PCMA, and telephone-event), and the sampling rate (8000). The last line specifies the allowable range of event codes for the telephone-event codec. there can be multiple attribute fields in an SDP message
- a=rtpmap:0 PCMU/8000: This specifies the codec for payload type 0 as PCMU with a sampling rate of 8000 Hz.
- a=rtpmap:31 H263/90000: This specifies the codec for payload type 31 as H.263 with a clock rate of 90000 Hz.
- a=sendrecv: This specifies that the media stream is bi-directional (send and receive).
- a=rtcp-mux: This specifies that the RTP and RTCP packets are multiplexed on the same port.

# RTPMAP

In the context of SDP, RTPMAP (RTP Mapping) is an attribute used to map a particular codec to an RTP payload type number. RTP is the Real-time Transport Protocol, which is used to transmit audio and video over IP networks. RTP uses payload types to identify the format of the data being transmitted.

The RTPMAP attribute provides a way for the sender to signal the receiver about the payload type numbers and the corresponding codecs being used. The attribute specifies the encoding name, the clock rate of the codec, and the number of audio channels being transmitted.

The syntax for the RTPMAP attribute is as follows:

```
a=rtptime:<payload type> <encoding name>/<clock rate>[/<encoding parameters>]
```

where:

- `<payload type>` is an integer value representing the payload type number.
- `<encoding name>` is a string that identifies the codec being used, such as "PCMU" for G.711 mu-law audio or "H264" for H.264 video.
- `<clock rate>` is an integer value indicating the clock rate of the codec, in Hz.
- `<encoding parameters>` is an optional string that specifies additional parameters for the codec, such as packetization mode or frame size.

Here is an example of an RTPMAP attribute for G.711 mu-law audio:

```
a=rtptime:0 PCMU/8000
```

This specifies that payload type 0 is used for G.711 mu-law audio, with a clock rate of 8000 Hz.

Overall, RTPMAP is an important attribute in SDP as it allows the sender and receiver to negotiate and agree upon the codec to be used for transmitting the media.

## All RTPMAP codecs

Payload type (PT)	Name	Type	No. of channels	Clock rate (Hz) <a href="#">[note 1]</a>	Frame size (byte)	Default packet interval (ms)	Description	References
-------------------	------	------	-----------------	---	-------------------	------------------------------	-------------	------------

0	PCMU	audio	1	8000	any	20	ITU-T <u>G.711</u> PCM $\mu$ - <u>Law</u> audio 64 kbit/s	RFC 3551
1	reserved (previously <u>FS-1016</u> <u>CELP</u> )	audio	1	8000			reserved, previously <u>FS-1016</u> <u>CELP</u> audio 4.8 kbit/s	RFC 3551, previously RFC 1890
2	reserved (previously G721 or G726-32)	audio	1	8000			reserved, previously ITU-T <u>G.721</u> <u>ADPCM</u> audio 32 kbit/s or ITU-T <u>G.726</u> audio 32 kbit/s	RFC 3551, previously RFC 1890
3	GSM	audio	1	8000	20	20	European <u>GSM Full</u> <u>Rate</u> audio 13 kbit/s (GSM 06.10)	RFC 3551
4	G723	audio	1	8000	30	30	ITU-T <u>G.723.1</u> audio	RFC 3551
5	DVI4	audio	1	8000	any	20	<u>IMA</u> <u>ADPCM</u> audio 32 kbit/s	RFC 3551
6	DVI4	audio	1	16000	any	20	<u>IMA</u> <u>ADPCM</u> audio 64 kbit/s	RFC 3551

7	LPC	audio	1	8000	any	20	Experimental <u>Linear Predictive Coding</u> audio 5.6 kbit/s	RFC 3551
8	PCMA	audio	1	8000	any	20	ITU-T G.711 PCM <u>A-Law</u> audio 64 kbit/s	RFC 3551
9	G722	audio	1	8000 <sup>[note 2]</sup>	any	20	ITU-T <u>G.722</u> audio 64 kbit/s	<u>RFC 3551 - Page 14</u>
10	L16	audio	2	44100	any	20	<u>Linear PCM</u> 16-bit Stereo audio 1411.2 kbit/s, <sup>[2][3]</sup> <sup>[4]</sup> uncompressed	<u>RFC 3551, Page 27</u>
11	L16	audio	1	44100	any	20	<u>Linear PCM</u> 16-bit audio 705.6 kbit/s, uncompressed	<u>RFC 3551, Page 27</u>
12	QCELP	audio	1	8000	20	20	<u>Qualcomm Code Excited Linear Prediction</u>	RFC 2658, <u>RFC 3551</u>

13	CN	audio	1	8000			<u>Comfort noise.</u> Payload type used with audio codecs that do not support comfort noise as part of the codec itself such as <u>G.711</u> , <u>G.722.1</u> , <u>G.722</u> , <u>G.726</u> , <u>G.727</u> , <u>G.728</u> , <u>GSM 06.10</u> , <u>Siren</u> , and <u>RTAudio</u> .	RFC 3389
14	MPA	audio	1, 2	90000	8-72		<u>MPEG-1</u> or <u>MPEG-2</u> audio only	RFC 3551, RFC 2250
15	G728	audio	1	8000	2.5	20	ITU-T <u>G.728</u> audio 16 kbit/s	RFC 3551
16	DVI4	audio	1	11025	any	20	<u>IMA ADPCM</u> audio 44.1 kbit/s	RFC 3551
17	DVI4	audio	1	22050	any	20	IMA ADPCM audio 88.2 kbit/s	RFC 3551

18	G729	audio	1	8000	10	20	ITU-T <u>G.729</u> and G.729a audio 8 kbit/s; Annex B is implied unless the <code>annexb=no</code> parameter is used	<u>RFC 3551</u> , <u>Page 20</u> , <u>RFC 3555</u> , <u>Page 15</u>
19	reserved (previousl y CN)	audio					reserved, previously <u>comfort</u> <u>noise</u>	RFC 3551
25	CELLB	video		90000			<u>Sun</u> CellB video[5]	RFC 2029
26	JPEG	video		90000			<u>JPEG</u> video	RFC 2435
28	nv	video		90000			<u>Xerox</u> <u>PARC's</u> Network Video (nv) [6][7]	<u>RFC 3551</u> , <u>Page 32</u>
31	H261	video		90000			ITU-T <u>H.261</u> video	RFC 4587
32	MPV	video		90000			MPEG-1 and MPEG-2 video	RFC 2250
33	MP2T	audio/vid eo		90000			MPEG-2 <u>transport</u> <u>stream</u>	RFC 2250
34	H263	video		90000			<u>H.263</u> video, first version (1996)	RFC 3551, RFC 2190

72-76	reserved						reserved because RTCP packet types 200-204 would otherwise be indistinguishable from RTP payload types 72-76 with the marker bit set	RFC 3550, RFC 3551
77-95	unassigned						note that RTCP packet type 207 (XR, Extended Reports) would be indistinguishable from RTP payload types 79 with the marker bit set	RFC 3551, RFC 3611
dynamic	H263-1998	video		90000			<u>H.263</u> video, second version (1998)	RFC 3551, RFC 4629, RFC 2190
dynamic	H263-2000	video		90000			<u>H.263</u> video, third version (2000)	RFC 4629
dynamic (or profile)	H264 AVC	video		90000			<u>H.264</u> video (MPEG-4 Part 10)	RFC 6184, previously RFC 3984

dynamic (or profile)	H264 SVC	video		90000			<u>H.264</u> video	RFC 6190
dynamic (or profile)	H265	video		90000			<u>H.265</u> video (HEVC)	RFC 7798
dynamic (or profile)	theora	video		90000			<u>Theora</u> video	<u>draft-</u> <u>barbato-</u> <u>avt-rtp-</u> <u>theora</u>
dynamic	iLBC	audio	1	8000	20, 30	20, 30	<u>Internet</u> <u>low</u> <u>Bitrate</u> <u>Codec</u> 13.33 or 15.2 kbit/ s	RFC 3952
dynamic	PCMA-WB	audio	1	16000	5		ITU-T <u>G.711.1</u> A-law	RFC 5391
dynamic	PCMU-WB	audio	1	16000	5		ITU-T <u>G.711.1</u> $\mu$ -law	RFC 5391
dynamic	G718	audio		32000 (placeholder)	20		ITU-T <u>G.718</u>	<u>draft-ietf-</u> <u>payload-</u> <u>rtp-g718</u>
dynamic	G719	audio	(various)	48000	20		ITU-T <u>G.719</u>	RFC 5404
dynamic	G7221	audio		16000, 32000	20		ITU-T <u>G.722.1</u> and G.722.1 Annex C	RFC 5577
dynamic	G726-16	audio	1	8000	any	20	ITU-T <u>G.726</u> audio 16 kbit/s	RFC 3551
dynamic	G726-24	audio	1	8000	any	20	ITU-T G.726 audio 24 kbit/s	RFC 3551

dynamic	G726-32	audio	1	8000	any	20	ITU-T G.726 audio 32 kbit/s	RFC 3551
dynamic	G726-40	audio	1	8000	any	20	ITU-T G.726 audio 40 kbit/s	RFC 3551
dynamic	G729D	audio	1	8000	10	20	ITU-T <u>G.729</u> Annex D	RFC 3551
dynamic	G729E	audio	1	8000	10	20	ITU-T <u>G.729</u> Annex E	RFC 3551
dynamic	G7291	audio		16000	20		ITU-T <u>G.729.1</u>	RFC 4749
dynamic	GSM-EFR	audio	1	8000	20	20	ITU-T <u>GSM-EFR</u> (GSM 06.60)	RFC 3551
dynamic	GSM-HR-08	audio	1	8000	20		ITU-T <u>GSM-HR</u> (GSM 06.20)	RFC 5993
dynamic (or profile)	AMR	audio	(various)	8000	20		<u>Adaptive Multi-Rate</u> audio	RFC 4867
dynamic (or profile)	AMR-WB	audio	(various)	16000	20		<u>Adaptive Multi-Rate Wideband</u> audio (ITU-T G.722.2)	RFC 4867
dynamic (or profile)	AMR-WB+	audio	1, 2 or omit	72000	13.3-40		<u>Extended Adaptive Multi Rate</u> = <u>WideBand</u> audio	RFC 4352
dynamic (or profile)	vorbis	audio	(various)	(various)			<u>Vorbis</u> audio	RFC 5215

dynamic (or profile)	opus	audio	1, 2	48000 <a href="#">[note 3]</a>	2.5–60	20	<u>Opus</u> audio	RFC 7587
dynamic (or profile)	speex	audio	1	8000, 16000, 32000	20		<u>Speex</u> audio	RFC 5574
dynamic	mpa-robust	audio	1, 2	90000	24–72		Loss-Tolerant <u>MP3</u> audio	RFC 5219 (previously RFC 3119)
dynamic (or profile)	MP4A-LATM	audio		90000 or others			<u>MPEG-4 Audio</u> (includes <u>AAC</u> )	RFC 6416 (previously RFC 3016)
dynamic (or profile)	MP4V-ES	video		90000 or others			<u>MPEG-4 Visual</u>	RFC 6416 (previously RFC 3016)
dynamic (or profile)	mpeg4-generic	audio/video		90000 or other			<u>MPEG-4</u> Elementary Streams	RFC 3640
dynamic	VP8	video		90000			<u>VP8</u> video	RFC 7741
dynamic	VP9	video		90000			<u>VP9</u> video	<a href="#">draft-ietf-payload-vp9</a>
dynamic	L8	audio	(various)	(various)	any	20	<u>Linear PCM</u> 8-bit audio with 128 offset	RFC 3551 Section 4.5.10 and Table 5
dynamic	DAT12	audio	(various)	(various)	any	20 (by analogy with L16)	IEC 61119 12-bit nonlinear audio	RFC 3190 Section 3
dynamic	L16	audio	(various)	(various)	any	20	<u>Linear PCM</u> 16-bit audio	RFC 3551 Section 4.5.11, RFC 2586
dynamic	L20	audio	(various)	(various)	any	20 (by analogy with L16)	<u>Linear PCM</u> 20-bit audio	RFC 3190 Section 4

dynamic	L24	audio	(various)	(various)	any	20 (by analogy with L16)	<u>Linear PCM</u> 24-bit audio	RFC 3190 Section 4
dynamic	raw	video		90000			Uncompressed Video	RFC 4175
dynamic	ac3	audio	(various)	32000, 44100, 48000			<u>Dolby AC-3</u> audio	RFC 4184
dynamic	eac3	audio	(various)	32000, 44100, 48000			<u>Enhanced AC-3</u> audio	RFC 4598
dynamic	t140	text		1000			<u>Text over IP</u>	RFC 4103
dynamic	EVRC EVRC0 EVRC1	audio		8000			<u>EVRC</u> audio	RFC 4788
dynamic	EVRCB EVRCB0 EVRCB1	audio		8000			<u>EVRC-B</u> audio	RFC 4788
dynamic	EVRCWB EVRCWB0 EVRCWB1	audio		16000			<u>EVRC-WB</u> audio	RFC 5188
dynamic	jpeg2000	video		90000			<u>JPEG 2000</u> video	RFC 5371
dynamic	UEMCLIP	audio		8000, 16000			<u>UEMCLIP</u> audio	RFC 5686
dynamic	ATRAC3	audio		44100			<u>ATRAC3</u> audio	RFC 5584
dynamic	ATRAC-X	audio		44100, 48000			<u>ATRAC3+</u> audio	RFC 5584
dynamic	ATRAC-ADVANCE D-LOSSLESS	audio		(various)			<u>ATRAC</u> Advanced Lossless audio	RFC 5584
dynamic	DV	video		90000			<u>DV</u> video	RFC 6469 (previously RFC 3189)

dynamic	BT656	video					<u>ITU-R</u> <u>BT.656</u> video	RFC 3555
dynamic	BMPEG	video					Bundled MPEG-2 video	RFC 2343
dynamic	SMPTE29 2M	video					<u>SMPTE</u> <u>292M</u> video	RFC 3497
dynamic	RED	audio					Redundan t Audio Data	RFC 2198
dynamic	VDVI	audio					Variable- rate DVI4 audio	RFC 3551
dynamic	MP1S	video					MPEG-1 Systems Streams video	RFC 2250
dynamic	MP2P	video					MPEG-2 Program Streams video	RFC 2250
dynamic	tone	audio		8000 (default)			tone	RFC 4733
dynamic	telephone -event	audio		8000 (default)			<u>DTMF</u> tone	RFC 4733
dynamic	aptx	audio	2 - 6	(equal to sampling rate)	4000 ÷ sample rate	4[ <u>note 4</u> ]	<u>aptX</u> audio	RFC 7310
dynamic	jxsv	video		90000			<u>JPEG XS</u> video	RFC 9134

## FMTP

The "fmtp" (format parameters) attribute in SDP (Session Description Protocol) is used to describe the format parameters of a media stream, specifically for codecs that support dynamic negotiation of parameters. It specifies the media format parameters as a set of parameter name-value pairs, separated by semicolons.

The format of the "fmtp" attribute is as follows:

```
a=fmtp:<format> <parameter_name>=<value>;<parameter_name>=<value>;...
```

- `<format>`: specifies the media format to which the parameters apply.
- `<parameter_name>`: specifies the name of the parameter.
- `<value>`: specifies the value of the parameter.

For example, the "fmtp" attribute for the H.264 video codec may look like:

```
a=fmtp:120 profile-level-id=42801E; packetization-mode=1; sprop-parameter-sets=Z0IACpZTBYmI,aMljiA==
```

In this example, the `<format>` value is 120 (which is the format number for H.264). The `<parameter_name>` values include "profile-level-id", "packetization-mode", and "sprop-parameter-sets", and their corresponding `<value>`s are "42801E", "1", and "Z0IACpZTBYmI,aMljiA==", respectively. These parameters provide information such as the profile and level of the codec, the packetization mode, and the parameter sets required for decoding the video stream.

The "fmtp" attribute is typically used in conjunction with the "rtpmap" attribute to describe media formats that use RTP (Real-time Transport Protocol) for transmission.

This is a SIP response to an INVITE request shown above. It has a status code of 200 OK, indicating that the request was successful. The Via header indicates the path that the request followed. The To header contains the tag that identifies the current dialog. The From header contains the tag that identifies the previous dialog. The Call-ID header is used to identify the call. The CSeq header contains the sequence number and method of the request. The Contact header specifies the address where the response should be sent. The Content-Type header indicates that the body of the message is in SDP format. The SDP message body includes connection information, timing information, and media information for both audio and video. In this response, some codecs have been removed from both audio and video streams.

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bK776asdhds
To: Bob <sip:bob@biloxi.com>;tag=2482893830n
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710
CSeq: 314159 INVITE
Contact: <sip:bob@192.0.2.4:5060>
Content-Type: application/sdp
Content-Length: 204

v=0
o=- 123456 789012 IN IP4 192.0.2.4
```

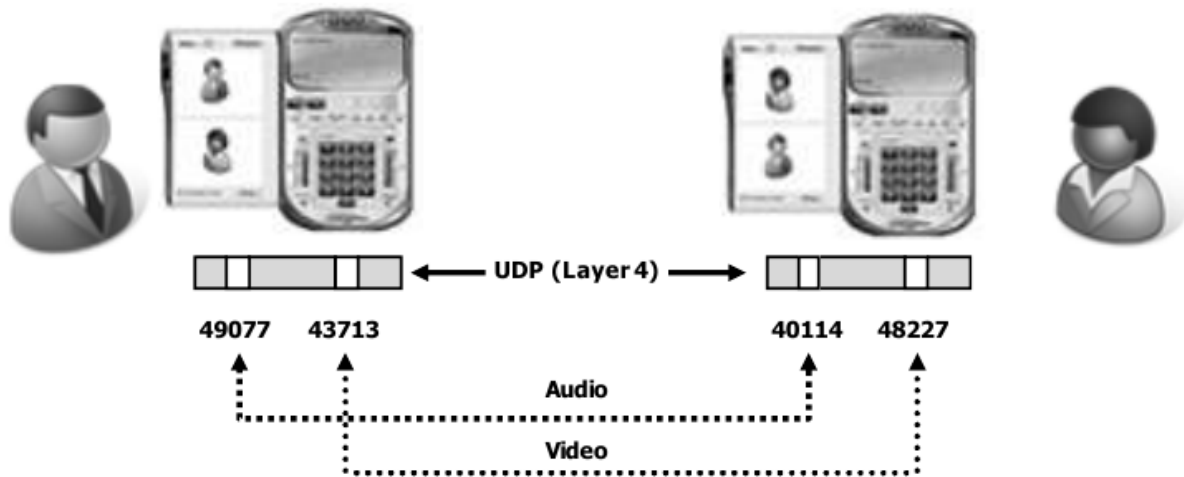
```
s=SIP Call
c=IN IP4 192.0.2.1
t=0 0
m=audio 49170 RTP/AVP 0 18
a=rtpmap:0 PCMU/8000
a=rtpmap:18 G729/8000
a=fmtp:18 annexb=no
m=video 49172 RTP/AVP 31
a=rtpmap:31 H261/90000
```

- `v=0`: The protocol version being used is 0.
- `o=- 123456 789012 IN IP4 pc33.atlanta.com`: This is the origin field. The `-` indicates that the session is not valid for further communication. `123456` is the session ID, which is chosen arbitrarily. `789012` is the session version number. `IN` stands for "Internet", and `IP4 pc33.atlanta.com` (192.0.2.4) is the address of the host that originated the session.
- `s=SIP Call`: The session name, which is an optional field. In this case, it is "SIP Call".
- `c=IN IP4 192.0.2.1`: The connection information for the session. This field specifies the network type, address type, and address information. In this case, it indicates that the session is using the Internet network, IPv4 address, and the address is `192.0.2.1`.
- `t=0 0`: The timing information for the session. The first `0` indicates the start time of the session, which is "now". The second `0` indicates the session will not terminate automatically.
- `m=audio 49170 RTP/AVP 0 18`: This is the media description for the audio stream. `audio` indicates that this is an audio stream. `49170` is the port number being used for the RTP stream. `RTP/AVP` specifies the protocol being used (RTP) and the payload format for the audio data (AVP). `0` and `18` are the RTP payload types being used for this stream.
- `a=rtpmap:0 PCMU/8000`: This attribute maps the payload type `0` to the audio codec PCMU and indicates that the audio is sampled at a rate of 8000 Hz.
- `a=rtpmap:18 G729/8000`: This attribute maps the payload type `18` to the audio codec G.729 and indicates that the audio is sampled at a rate of 8000 Hz.
- `a=fmtp:18 annexb=no`: This is an optional attribute that provides additional parameters for the G.729 codec. In this case, it specifies that the codec is not using the Annex B format.
- `m=video 49172 RTP/AVP 31`: This is the media description for the video stream. `video` indicates that this is a video stream. `49172` is the port number being used for the RTP stream. `RTP/AVP` specifies the protocol being used (RTP) and the payload format for the video data (AVP). `31` is the RTP payload type being used for this stream.
- `a=rtpmap:31 H261/90000`: This attribute maps the payload type `31` to the video codec H.261 and indicates that the video is sampled at a rate of 90000 Hz.

## Multiple M Lines

In SIP SDP, multiple "m=" lines can be included in a single SDP message to provide information about multiple media streams within a session. Each "m=" line corresponds to a different type of media, such as audio, video, or data. The "m=" line specifies the transport protocol and port number to be used for that particular media stream, as well as the format of the media data being transmitted. Each "m=" line can also have its own set of media-level attributes, such as the RTP payload types being used for different codecs, or the clock rate of the media data. By including multiple "m=" lines in a single SDP message, a SIP session can support multiple media streams simultaneously.

## Multiple 'm' lines



INVITE (SDP Element)

```
m=audio 49077 RTP/AVP 0 8 97
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
m=video 43713 RTP/AVP 31 32
a=rtpmap:31 H261/90000
a=rtpmap:32 MPV/90000
```

200 OK (SDP Element)

```
m=audio 40114 RTP/AVP 0
a=rtpmap:0 PCMU/8000
m=video 48227 RTP/AVP 32
a=rtpmap:32 MPV/90000
```

Revision #4

Created 6 May 2023 05:30:53 by Cesar Gzz

Updated 6 May 2023 17:25:28 by Cesar Gzz