

# 7 SDP协议分析

---

## 1.SDP的结构

## 2. SDP语法

### 2.1 必需字段

### 2.2 可选字段

### 2.3 字段顺序

### 2.4 子字段

## 3 SDP协议例子

### 例子2

### WebRTC：会话描述协议SDP

#### 什么是SDP

#### 协议格式说明

#### 常见属性

协议版本号：v=

会话发起者：o

会话名 s=

连接数据：c=

媒体描述：m=

附加属性：a=

时间：t=

#### WebRTC实例

#### 写在后面

#### 相关链接

#### 参考链接：

### 腾讯课堂 零声学院

FFmpeg/WebRTC/RTMP 音视频流媒体高级开发 <https://ke.qq.com/course/468797?tuin=137bb271>

SDP (Session Description Protocol) 完全是一种会话描述格式 — 它不属于传输协议 — 它只使用不同的适当的传输协议, 包括会话通知协议 (SAP)、会话初始协议 (SIP)、[实时流协议 \(RTSP\)](#)、MIME 扩展协议的电子邮件以及超文本传输协议 (HTTP)。SDP协议是也是基于文本的协议, 这样就能保证协议的可扩展性比较强, 这样就使其具有广泛的应用范围。SDP 不支持会话内容或媒体编码的协商, 所以在流媒体中只用来描述媒体信息。媒体协商这一块要用[RTSP](#)来实现。

### SDP协议格式

SDP描述由[许多文本行](#)组成, 文本行的格式为<类型>=<值>, <类型>是一个字母, <值>是结构化的文本串, 其格式依<类型>而定。

<type>=<value>[\[CRLF\]](#)。[SDP](#)在4566 中有详细描述。

## 1.SDP的结构

[SDP](#)仅仅提供了描述从会话信息到可能的会话参与者的格式。一个会话基本上有许多媒体流组成, 因此, 会话描述包含了许多与每个媒体流相关的参数说明, 此外还包括与会话整体相关的通用信息。所以, [SDP](#)消息中既包含了会话级参数又包括了媒体级参数。会话级参数包括如下信息, 如会话名称、会话发起者以及会话活动时间。媒体级参数包括媒体类型、端口号、传输协议以及媒体格式。下图描述了[SDP](#)消息的基本结构。



SDP 仅仅提供了对会话的描述，没有提供将会话和可能的参与者联系起来的方法，所以必须把SDP与其他协议（如SIP协议）联系起来使用。SDP协议同SIP协议一样基于文本的协议，需要使用特定的编码集。字段名称只能使用US-ASCII字符集，文本信息可以使用任何语言。由于SDP的ASCII编码比二进制编码占用带宽多，所以SDP采用紧凑格式提高带宽利用率。如v=version，s=session name等等。

SDP采用多文本行传递会话信息，每行使用“字段名=字段值”的格式。字段名只用一个字符表示（大小写敏感），字段值可以有多个信息块组成，用分号隔开，“=”左右不能有空格。

## 2. SDP语法

会话级别字段必须放在前面，然后才是媒体级参数，会话数据与媒体数据之间的界限就是第一个媒体描述字段（m=）的出现，之后的每个媒体描述字段的出现标志着这个会话中又一个媒体流数据的开始。

SDP字段包括必需字段与可选字段。

## 2.1 必需字段

- v= (协议版本号)，一个会话描述的开始，前一个会话结束标志。
- o= (会话源或者会话生成者，以及会话标识符)
- s= (会话名称) 这个字段是个文本字符串，可以显示给会话参与者。
- t= (会话时间) 这个字段指明会话开始时间与结束时间。
- m= (媒体) 该字段用来指明媒体类型、数据应该发送到的传输端口，传输协议 (例如RTP) 以及媒体格式 (例如RTP负载格式)

## 2.2 可选字段

SDP可选字段中，一些只能应用于会话级，一些只能应用于媒体级，还有一些可以应用于两种级别。对于应用于两种级别的字段，在某个具体媒体上，应用于媒体级的字段会覆盖应用于会话级的字段值。例如，某个字段其会话级的字段值为X,媒体1的该字段值为Y，那么值X应用于会话中除媒体类型1之外的所有媒体，而媒体1应用值Y。

可选字段如下：

- i= (会话信息) 对字段的文本描述，提供了比会话名称更多的信息。该字段既可以用于会话级也可以用于媒体级。
- u= (描述的URI地址) URI信息，通过这个地址可以获取更多会话相关信息。例如，一个会议可能公布在WEB页面上，所以需要该WEB的URI。每个会话只能提供一个URI
- E= (E-mail地址) 负责会话个体的E-mail地址，可以有多个。只能用于会话级别。
- p= (电话号码) 同email一样，多个，会话级别。
- c= (连接信息) 该字段提供连接数据，包括连接类型、网络类型和连接地址。可应用于会话级也可以用于媒体级。
- b= (带宽信息) 指明带宽需求，单位kbit/s, 可用于两个级别
- r= (重复次数) 如果是有规律的日程安排活动，这个字段用来指明会话重复频次和时间。
- z= (时区调整) 用于按日程安排的有规律活动会话。会话可能会夸时区，避免时区变更造成的混乱。
- k= (加密密钥) 为了对媒体加密、解密，该字段提供了一个加密密钥或者规定了一个获取密钥的机制。可用于两个级别。
- a= (属性) 用于描述会话或者某个媒体的额外属性。

可选字段中，连接信息字段是特别与条件相关的。因为该字段定义了数据应发送到的网络地址，所以必须被包含进会话描述中的某个地方。

## 2.3 字段顺序

因为有些字段既可以应用于会话级又可以用于媒体级，所以字段顺序对于避免含义模糊特别重要。SDP所要求的字段顺序如下：

## 会话级

协议版本号 (v)

会话源 (o)

会话名称 (s)

会话信息 (i) (可选)

URI (u) (可选)

E-mail地址 (e) (可选)

电话号码 (p) (可选)

连接信息 (c) (可选)

带宽信息 (b) (可选)

时间描述 (t)

重复信息 (r) (可选)

时区调整 (z) (可选)

加密密钥 (k) (可选)

属性 (a) (可选)

## 媒体级

媒体描述 (m)

媒体信息 (i) (可选)

连接信息 (c) (会话级进行了规定，这里可选)

带宽信息 (b) (可选)

加密密钥 (k) (可选)

属性 (a) (可选)

## 2.4 子字段

在SDP中许多字段采用多个子字段的形式，此时，这些字段值由多个以空格符间隔的多个值组成。格式如下：

字段 名称=<子字段1的值> <子字段2的值> <子字段3的值>

下面举例说明比较重要的字段

- 会话源 (o)

有6个子字段：用户名、会话ID、版本、网络类型、地址类型、地址。

- 连接信息

连接数据有3个子字段：网络类型、地址类型和连接地址。同会话源中含义不同，他们表示需要接收媒体数据的网络 and 地址，而不是生成会话的网络和地址。

- 媒体信息

媒体信息 (m) 有四个子字段：媒体类型、端口、传输协议、格式。

m描述行：

格式： m=(媒体) (端口) (传送层) (格式列表)

媒体类型： 音频 (audio), 视频(video), 应用, 数据和控制

端口： 媒体传送层端口

传送层： ip4上大多基于rtp/udp上传送 (RTP/AVP) IETF RTP协议, 在udp上传输

格式列表： 对应对应的音频负载类型 (PT)

m=video 0 RTP/AVP 96

```
> Session Attribute (a): tool:libavformat 58.29.100
> Media Description, name and address (m): video 0 RTP/AVP 96
> Media Attribute (a): fmp:96 packetization-mode=1; sprop-parameter-sets=Z2QAHqzZQLAx5ahAAAA
> Media Attribute (a): rtpmap:96 H264/90000
> Media Attribute (a): control:streamid=0
> Media Description, name and address (m): audio 0 RTP/AVP 97
```

a是对m描述行 的格式列表的 PT 做一步的描述。

媒体类型可以是音频、视频、应用程序、数据或控制，如果是语音，媒体类型就是音频。

端口指明媒体要发送到的端口号，端口号与所有的连接类型和传输协议有关。例如VoIP，媒体通常在UDP传输协议上采用RTP承载，这样端口号将是1024和65535之间的一个偶数值。相应奇数对应RTCP端口号。

格式子字段列出了所支持的所有不同类型的媒体格式。优先使用靠前的格式。

如某个系统准备在端口45678接收语音，并且只能处理G.711 u编码的语音，那么RTP负载类型为0，媒体信息如下：

m=audio 45678 RTP/AVP 0

如果某个系统准备在端口45678可以处理几种编码的语音：G.728编码格式（负载类型为15）、GSM编码格式（负载类型为3）、G.711 u编码格式（负载类型为0），而且系统优先采用G.728格式，则相应的媒体信息如下所示：

m=audio 45678 RTP/AVP 15 3 0

- 属性 (a)

属性可以描述额外信息，可应用于会话级，也可用于媒体级或者两者兼有。

a描述行（大部分情况是必须）：

格式：a=rtpmap: (净荷类型) (编码名) / (时钟速率) 【/(编码参数)】

a=control: (音/视频连接信息)

a=control:rtsp://192.168.1.197/h264stream0/trackID=0

a=rtpmap:96 H264/90000

属性有两种形式，第一种是特征属性，第二种属于值属性。SDP描述了多个建议属性。

例如a=sendonly

表明会话描述的发送者只希望发送数据而不打算接收数据，端口号无意义，可以置为0。

例如[a=recvonly](#)

表明这个会话描述的发送者只想接收数据而不打算发送数据。

例如[a=rtpmp:<负载类型><编码名称>/<时钟速率>\[<编码参数>\]](#)

[rtpmap](#)属性提供了一个在VoIP应用中的重要属性使用方法，该属性可用于媒体流，在媒体格式不是静态的RTP负载类型时特别有用。

严格来说，“[rtpmap](#)”只在使用动态负载类型情况下才是必须的，例如标准的G.711语音是静态RTP负载类型，采用如下方法就可以对它完整描述：

[m=audio 45678 RTP/AVP 0](#)

而对动态负载来说需要指定更多信息才能使远端完全识别到媒体编码，例如16位线性编码16kHz取样的立体声音就是一个动态RTP负载类型，如果我们采用动态负载类型98表示这个媒体流，那么SDP格式如下：

[m=audio 45678 RTP/AVP 98](#)

[a=rtpmap 98 L16/16000/2](#)

SIP建议无论是静态负载还是动态负载，尽量都要采用该属性。

### 3 SDP协议例子

下面是一个helix 流媒体服务器的RTSP协议中的SDP协议：

[v=0](#) //SDP version

// o field定义的源的一些信息。其格式为：o=<username> <sess-id> <sess-version> <nettype>  
<addrtype> <unicast-address>

[o=- 1271659412 1271659412 IN IP4 10.56.136.37 s=<No title>](#)

[i=<No author> <No copyright>](#) //session的信息

[c=IN IP4 0.0.0.0](#) //connect 的信息，分别描述了：网络协议，地址的类型，连接地址。

[c=IN IP4 0.0.0.0](#)

[t=0 0](#) //时间信息，分别表示开始的时间和结束的时间，一般在流媒体的直播的时移中见的比较多。

[a=SdpplnVersion:1610641560](#) //描述性的信息

[a=StreamCount:integer;2](#) //用来描述媒体流的信息，表示有两个媒体流。integer表示信息的格式为整数。

[a=control:\\*](#)

[a=DefaultLicenseValue:integer;0](#) //License信息

[a=FileType:string;"MPEG4"](#) ////用来描述媒体流的信息说明当前协商的文件是mpeg4格式的文件

[a=LicenseKey:string;"license.Summary.Datatypes.RealMPEG4.Enabled"](#)

[a=range:npt=0-72.080000](#) //用来表示媒体流的长度

[m=audio 0 RTP/AVP 96](#) //做为媒体描述信息的重要组成部分描述了媒体信息的详细内容：表示session的audio是通过RTP来格式传送的，其payload值为96传送的端口还没有定。

[b=as:24](#) //audio 的bitrate

[b=RR:1800](#)

```
b=RS:600
a=control:streamid=1 //通过媒体流1来发送音频
a=range:npt=0-72.080000 //说明媒体流的长度。
a=length:npt=72.080000
a=rtpmap:96 MPEG4-GENERIC/32000/2 //rtpmap的信息，表示音频为AAC的其sample为32000
a=fmtp:96 profile-level-id=15;mode=AAC-
hbr;sizelength=13;indexlength=3;indexdeltalength=3;config=1210 //config为AAC的详细格式信息
a=mimetype:string;"audio/MPEG4-GENERIC"
a=Helix-Adaptation-Support:1
a=AvgBitRate:integer;48000
a=HasOutOfOrderTS:integer;1
a=MaxBitRate:integer;48000
a=Preroll:integer;1000
a=OpaqueData:buffer;"A4CAgCIAAAAEgICAfEAVABgAAAC7gAAAu4AFgICAfKIBoCAgAEC"
a=StreamName:string;"Audio Track"
```

下面是video的信息基本和audio的信息相对称，这里就不再说了。

```
m=video 0 RTP/AVP 97
b=as:150
b=RR:11250
b=RS:3750
a=control:streamid=2
a=range:npt=0-72.080000
a=length:npt=72.080000
a=rtpmap:97 MP4V-ES/2500
a=fmtp:97 profile-level-id=1;
a=mimetype:string;"video/MP4V-ES"
a=Helix-Adaptation-Support:1
a=AvgBitRate:integer;300000
a=HasOutOfOrderTS:integer;1
a=Height:integer;240 //影片的长度
a=MaxBitRate:integer;300000
a=MaxPacketSize:integer;1400
a=Preroll:integer;1000
a=Width:integer;320 //影片的宽度
a=OpaqueData:buffer;"AzcAAB8ELyARAbd0AAST4AAEk+AFIAAAAbDzAAABtQ7gQMDPAAABAAA
AASAAhED6KFAG8KIfBgEC"
a=StreamName:string;"Video Track"
```



## 例子2

```
1 v=0
2 o=StreamingServer 3677033027 1437537780000 IN IP4 192.168.1.44
3 s=\demo.mp4
4 u=http:///
5 e=admin@
6 c=IN IP4 0.0.0.0
7 b=AS:1398
8 t=0 0
9 a=control:*
10 a=x-copyright: MP4/3GP File hinted with GPAC 0.5.0-rev4065 (C)200
    0-2005 - http://gpac.sourceforge.net
11 a=range:npt=0- 216.52167
12
13 m=video 0 RTP/AVP 96
14 b=AS:1242
15 a=3GPP-Adaptation-Support:1
16 a=rtpmap:96 H264/90000
17 a=control:trackID=65536
18 a=fmtp:96 profile-level-id=42000A; packetization-mode=1; sprop-pa
    rameter-sets=Z0IACpZUBQHogA==,aM44gA==
19 a=framesize:96 640-480
20
21 m=audio 0 RTP/AVP 97
22 b=AS:156
23 a=3GPP-Adaptation-Support:1
24 a=rtpmap:97 mpeg4-generic/48000/1
25 a=control:trackID=65537
26 a=fmtp:97 profile-level-id=41; config=1188; streamType=5; mode=AA
    C-hbr; objectType=64; constantDuration=1024; sizeLength=13; index
    Length=3; indexDeltaLength=3
```

视频”a=fmtp”字段的解析 参考 RFC3984的8.2节

音频config描述符的解析 参考 RFC 3016

# WebRTC：会话描述协议SDP

## 什么是SDP

SDP (Session Description Protocol) 是一种通用的会话描述协议，主要用来描述多媒体会话，用途包括会话声明、会话邀请、会话初始化等。

WebRTC主要在连接建立阶段用到SDP，连接双方通过信令服务交换会话信息，包括音视频编解码器(codec)、主机候选地址、网络传输协议等。

下面先简单介绍下SDP的格式、常用属性，然后通过WebRTC连接建立过程生成的SDP实例进行进一步讲解。

## 协议格式说明

SDP的格式非常简单，由多个行组成，每个行都是如下格式。

```
1 <type>=<value>
```

其中：

- `<type>`：大小写敏感的一个字符，代表特定的属性，比如 `v` 代表版本；
- `<value>`：结构化文本，格式与属性类型有关，UTF8编码；
- `=` 两边不允许存在空格；
- `=*` 表示是可选的；

## 常见属性

以下面的SDP为例：

```
1 v=0
2 o=alice 2890844526 2890844526 IN IP4 host.anywhere.com
3 s=
4 c=IN IP4 host.anywhere.com
5 t=0 0
6 m=audio 49170 RTP/AVP 0
7 a=rtpmap:0 PCMU/8000
8 m=video 51372 RTP/AVP 31
9 a=rtpmap:31 H261/90000
10 m=video 53000 RTP/AVP 32
11 a=rtpmap:32 MPV/90000
```

协议版本号： `v=`

格式如下，注意，没有子版本号。

```
1 v=0
```

## 会话发起者：o

格式如下，其中，username、session-id、nettype、addrtype、unicast-address 一起，唯一标识一个会话。

```
1 o=<username> <sess-id> <sess-version> <nettype> <addrtype> <unicas  
t-address>
```

各字段含义如下：

- username：发起者的用户名，不允许存在空格，如果应用不支持用户名，则为-。
- sess-id：会话id，由应用自行定义，规范的建议是NTP(Network Time Protocol)时间戳。
- sess-version：会话版本，用途由应用自行定义，只要会话数据发生变化时（比如编码），sess-version随着递增就行。同样的，规范的建议是NTP时间戳。
- nettype：网络类型，比如IN表示Internet。
- addrtype：地址类型，比如IP4、IP6。
- unicast-address：域名，或者IP地址。

## 会话名 s=

必选，有且仅有一个s=字段，且不能为空。如果实在没有有意义的会话名，可以赋一个空格，即s=。

```
1 s=<session name>
```

## 连接数据：c=

格式如下：

```
1 c=<nettype> <addrtype> <connection-address>
```

每个SDP至少需要包含一个会话级别的c=字段，或者在每个媒体描述后面各包含一个c=字段。（媒体描述后的c=会覆盖会话级别的c=）

- nettype：网络类型，比如IN，表示Internet。
- addrtype：地址类型，比如IP4、IP6。
- connection-address：如果是广播，则为广播地址组；如果是单播，则为单播地址；

举例01：

```
1 c=IN IP4 224.2.36.42/127
```

举例02：

```
1 c=IN IP4 host.anywhere.com
```

## 媒体描述： `m=`

SDP可能同时包含多个媒体描述。格式如下：

```
1 m=<media> <port> <proto> <fmt> ...
```

其中：

- media：媒体类型。包括 video、audio、text、application、message等。
- port：传输媒体流的端口，具体含义取决于使用的网络类型（在 `c=` 中声明）和使用的协议(proto，在 `m=` 中声明)。
- proto：传输协议，具体含义取决于 `c=` 中定义的地址类型，比如 `c=` 是IP4，那么这里的传输协议运行在IP4之上。比如：
  - UDP：传输层协议是UDP。
  - RTP/AVP：针对视频、音频的RTP协议，跑在UDP之上。
  - RTP/SAVP：针对视频、音频的SRTP协议，跑在UDP之上。
- fmt：媒体格式的描述，可能有多个。根据 proto 的不同，fmt 的含义也不同。比如 proto 为 RTP/SAVP 时，fmt 表示 RTP payload 的类型。如果有多个，表示在这次会话中，多种payload类型可能会用到，且第一个为默认的payload类型。

举例，下面表示媒体类型是视频，采用SRTP传输流媒体数据，且RTP包的类型可能是122、102...119，默认是122。

```
1 m=video 9 UDP/TLS/RTP/SAVPF 122 102 100 101 124 120 123 119
```

对于 RTP/SAVP，需要注意的是，payload type 又分两种类型：

1. 静态类型：参考 [RTP/AVP audio and video payload types](#)。
2. 动态类型：在 `a=fmt` 中进行定义。（`a=` 为附加属性，见后面小节）

举例，下面的SDP中：

1. 对于audio，111 是动态类型，表示 `opus/48000/2`。
2. 对于video，122 是动态类型，表示 `H264/90000`。

```
1 m=audio 9 UDP/TLS/RTP/SAVPF 111 103 104 9 0 8 126
2 a=rtpmap:111 opus/48000/2
3 m=video 9 UDP/TLS/RTP/SAVPF 122 102 100 101 124 120 123 119
4 a=rtpmap:122 H264/90000
```

## 附加属性： `a=`

作用：用于扩展SDP。

有两种作用范围：会话级别(session-level)、媒体级别 (media-level) 。

1. 媒体级别：媒体描述 (`m=`) 后面可以跟任意数量的 `a=` 字段，对媒体描述进行扩展。
2. 会话级别：在第一个媒体字段(media field)前，添加的 `a=` 字段是会话级别的。

有如下两种格式：

```
1 a=<attribute>
2 a=<attribute>:<value>
```

格式1举例：

```
1 a=recvonly
```

格式2举例：

```
1 a=rtpmap:0 PCMU/8000
```

**时间：** **t=**

作用：声明会话的开始、结束时间。

格式如下：

```
1 t=<start-time> <stop-time>
```

如果 `<stop-time>` 是0，表示会话没有结束的边界，但是需要在 `<start-time>` 之后会话才是活跃 (active) 的。如果 `<start-time>` 是0，表示会话是永久的。

举例：

```
1 t=0 0
```

## WebRTC实例

下面例子来自腾讯云WebRTC服务的远端offer。

```
1 // sdp版本号为0
2 v=0
3 // o=<username> <sess-id> <sess-version> <nettype> <addrtype> <u
  nicast-address>
4 // 用户名为空，会话id是8100750360520823155，会话版本是2（后面如果有类似改
  变编码的操作，sess-version加1），地址类型为IP4，地址为127.0.0.1（这里可以
  忽略）
5 o=- 7595655801978680453 2 IN IP4 112.90.139.105
6 // 会话名为空
7 s=-
8 // 会话的起始时间，都为0表示没有限制
9 t=0 0
10 a=ice-lite
```

```

11 // 音频、视频的传输采取多路复用，通过同一个RTP通道传输音频、视频，可以参考 https://tools.ietf.org/html/draft-ietf-mmusic-sdp-bundle-negotiation-54
12 a=group:BUNDLE 0 1
13 // WMS是WebRTC Media Stream的缩写，这里给Media Stream定义了一个唯一的标识符。一个Media Stream可以有多个track (video track、audio track)，这些track就是通过这个唯一标识符关联起来的，具体见下面的媒体行(m=)以及它对应的附加属性(a=ssrc:)
14 // 可以参考这里 http://tools.ietf.org/html/draft-ietf-mmusic-msid
15 a=msid-semantic: WMS 5Y2wZK8nANNAoVw6dSAHVjNxrD10bBM2kBPV
16 // m=<media> <port> <proto> <fmt> ...
17 // 本次会话有音频，端口为9（可忽略，端口9为Discard Protocol专用），采用UDP传输加密的RTP包，并使用基于SRTCP的音视频反馈机制来提升传输质量，111、103、104等是audio可能采用的编码（参见前面m=的说明）
18 m=audio 9 UDP/TLS/RTP/SAVPF 111 103 104 9 0 8 126
19 // 音频发送者的IP4地址，WebRTC采用ICE，这里的 0.0.0.0 可直接忽略
20 c=IN IP4 0.0.0.0
21 // RTCP采用的端口、IP地址（可忽略）
22 a=rtcp:9 IN IP4 0.0.0.0
23 // ice-ufrag、ice-pwd 分别为ICE协商用到的认证信息
24 a=ice-ufrag:58142170598604946
25 a=ice-pwd:71696ad0528c4adb02bb40e1
26 // DTLS协商过程的指纹信息
27 a=fingerprint:sha-256 7F:98:08:AC:17:6A:34:DB:CF:3B:EC:93:ED:57:3F:5A:9E:1F:4A:F3:DB:D5:BF:66:EE:17:58:E0:57:EC:1B:19
28 // 当前客户端在DTLS协商过程中，既可以作为客户端，也可以作为服务端，具体可参考 RFC4572
29 a=setup:actpass
30 // 当前媒体行的标识符（在a=group:BUNDLE 0 1 这行里面用到，这里0表示audio）
31 a=mid:0
32 // RTP允许扩展首部，这里表示采用了RFC6464定义的针对audio的扩展首部，用来调节音量，比如在大型会议中，有多个音频流，就可以用这个来调整音频混流的策略
33 // 这里没有vad=1，表示不启用这个音量控制
34 a=extmap:1 urn:ietf:params:rtp-hdrext:ssrc-audio-level
35 // 表示既可以发送音频，也可以接收音频
36 a=sendrecv
37 // 表示启用多路复用，RTP、RTCP共用同个通道
38 a=rtcp-mux
39 // 下面几行都是对audio媒体行的补充说明（针对111），包括rtpmap、rtcp-fb、f

```

```

mtp
40 // rtpmap: 编解码器为opus, 采样率是48000, 2声道
41 a=rtpmap:111 opus/48000/2
42 // rtcp-fb: 基于RTCP的反馈控制机制, 可以参考 https://tools.ietf.org/html/rfc5124、https://webrtc.org/experiments/rtp-hdrext/transport-wide-cc-02/
43 a=rtcp-fb:111 transport-cc
44 a=rtcp-fb:111 nack
45 // 最小的音频打包时间
46 a=fmtp:111 minptime=20
47 // 跟前面的rtpmap类似
48 a=rtpmap:126 telephone-event/8000
49 // ssrc用来对媒体进行描述, 格式为a=ssrc:<ssrc-id> <attribute>:<value>, 具体可参考 RFC5576
50 // cname用来唯一标识媒体的数据源
51 a=ssrc:16864608 cname:YZcxBwerFFm6GH69
52 // msid后面带两个id, 第一个是MediaStream的id, 第二个是audio track的id (跟后面的mslabel、label对应)
53 a=ssrc:16864608 msid:5Y2wZK8nANNAoVw6dSAHVjNxrD10bBM2kBPV 128f4fa0-81dd-4c3a-bbcd-22e71e29d178
54 a=ssrc:16864608 mslabel:5Y2wZK8nANNAoVw6dSAHVjNxrD10bBM2kBPV
55 a=ssrc:16864608 label:128f4fa0-81dd-4c3a-bbcd-22e71e29d178
56 // 跟audio类似, 不赘述
57 m=video 9 UDP/TLS/RTP/SAVPF 122 102 125 107 124 120 123 119
58 c=IN IP4 0.0.0.0
59 a=rtcp:9 IN IP4 0.0.0.0
60 a=ice-ufrag:58142170598604946
61 a=ice-pwd:71696ad0528c4adb02bb40e1
62 a=fingerprint:sha-256 7F:98:08:AC:17:6A:34:DB:CF:3B:EC:93:ED:57:3F:5A:9E:1F:4A:F3:DB:D5:BF:66:EE:17:58:E0:57:EC:1B:19
63 a=setup:actpass
64 a=mid:1
65 a=extmap:2 urn:ietf:params:rtp-hdrext:toffset
66 a=extmap:3 http://www.webrtc.org/experiments/rtp-hdrext/abs-send-time
67 a=extmap:4 urn:3gpp:video-orientation
68 a=extmap:5 http://www.ietf.org/id/draft-holmer-rmcat-transport-wide-cc-extensions-01
69 a=extmap:6 http://www.webrtc.org/experiments/rtp-hdrext/playout-delay

```

```
70 a=sendrecv
71 a=rtcp-mux
72 a=rtcp-rsize
73 a=rtpmap:122 H264/90000
74 a=rtcp-fb:122 ccm fir
75 a=rtcp-fb:122 nack
76 a=rtcp-fb:122 nack pli
77 a=rtcp-fb:122 goog-remb
78 a=rtcp-fb:122 transport-cc
79 a=fmtp:122 level-asymmetry-allowed=1;packetization-mode=1;profile-level-id=42001f
80 a=rtpmap:102 rtx/90000
81 a=fmtp:102 apt=122
82 a=rtpmap:125 H264/90000
83 a=rtcp-fb:125 ccm fir
84 a=rtcp-fb:125 nack
85 a=rtcp-fb:125 nack pli
86 a=rtcp-fb:125 goog-remb
87 a=rtcp-fb:125 transport-cc
88 a=fmtp:125 level-asymmetry-allowed=1;packetization-mode=1;profile-level-id=42e01f
89 a=rtpmap:107 rtx/90000
90 a=fmtp:107 apt=125
91 a=rtpmap:124 H264/90000
92 a=rtcp-fb:124 ccm fir
93 a=rtcp-fb:124 nack
94 a=rtcp-fb:124 nack pli
95 a=rtcp-fb:124 goog-remb
96 a=rtcp-fb:124 transport-cc
97 a=fmtp:124 level-asymmetry-allowed=1;packetization-mode=1;profile-level-id=4d0032
98 a=rtpmap:120 rtx/90000
99 a=fmtp:120 apt=124
100 a=rtpmap:123 H264/90000
101 a=rtcp-fb:123 ccm fir
102 a=rtcp-fb:123 nack
103 a=rtcp-fb:123 nack pli
104 a=rtcp-fb:123 goog-remb
105 a=rtcp-fb:123 transport-cc
106 a=fmtp:123 level-asymmetry-allowed=1;packetization-mode=1;profile-level-id=4d0032
```



```
e-level-id=640032
107 a=rtpmap:119 rtx/90000
108 a=fmtp:119 apt=123
109 a=ssrc-group:FID 33718809 50483271
110 a=ssrc:33718809 cname:ovaCctnHP9Ascii9c
111 a=ssrc:33718809 msid:5Y2wZK8nANNAoVw6dSAHVjNxrD10bBM2kBPV 1d7fc3
    00-9889-4f94-9f35-c0bcc77a260d
112 a=ssrc:33718809 mslabel:5Y2wZK8nANNAoVw6dSAHVjNxrD10bBM2kBPV
113 a=ssrc:33718809 label:1d7fc300-9889-4f94-9f35-c0bcc77a260d
114 a=ssrc:50483271 cname:ovaCctnHP9Ascii9c
115 a=ssrc:50483271 msid:5Y2wZK8nANNAoVw6dSAHVjNxrD10bBM2kBPV 1d7fc3
    00-9889-4f94-9f35-c0bcc77a260d
116 a=ssrc:50483271 mslabel:5Y2wZK8nANNAoVw6dSAHVjNxrD10bBM2kBPV
117 a=ssrc:50483271 label:1d7fc300-9889-4f94-9f35-c0bcc77a260d
```

## 写在后面

SDP协议格式本身很简单，难点一般在于应用层在不同场景下扩展出来的属性，以及不同扩展属性对应的含义。比如上面举的例子，扩展属性、属性值的说明分散在数十个RFC里，查找、理解都费了一番功夫。如有错漏，敬请指出。

## 相关链接

[SDP: Session Description Protocol](#)

[Annotated Example SDP for WebRTC](#)

## 参考链接：

sdp协议介绍

<https://www.cnblogs.com/idignew/p/7249056.html>

sdp协议

<https://www.jianshu.com/p/94b118b8fd97>

SDP协议详细总结

<https://blog.csdn.net/jobbofhe/article/details/78477407>