13

Negotiating Media Sessions

One of the most important uses of SIP is to negotiate the setup of sessions, as the name suggests. To do this, SIP uses another protocol, Session Description Protocol, to describe the actual parameters of the media session. This includes information such as media type, codec, bit rate, and the IP address and port numbers for the media session. In short, negotiating media sessions is all about exchanging the data necessary to begin the RTP media sessions described in Chapter 12 or SRTP media sessions described next in Chapter 14. This chapter will introduce the Session Description Protocol (SDP) and the Offer/Answer protocol, which is the way SIP uses SDP to negotiate sessions.

13.1 Session Description Protocol (SDP)

The Session Description Protocol, originally defined by RFC 2327 [1], was developed by the IETF MMUSIC working group. It is more of a description syntax than a protocol in that it does not provide a full-range media negotiation capability. The original purpose of SDP was to describe multicast sessions set up over the Internet's multicast backbone, the MBONE. The first application of SDP was by the experimental Session Announcement Protocol (SAP) [2] used to post and retrieve announcements of MBONE sessions. SAP messages carried an SDP message body, and were the template for SIP's use of SDP. Even though it was designed for multicast, SDP has been applied to the more general problem of describing general multimedia sessions established using SIP. SDP is current specified by RFC 4566 [3], which is mostly compatible with RFC 2327.

As seen in the examples of Chapter 2, SDP contains the following information about the media session:

- IP address (IPv4 or IPv6 address or host name);

- RTP profile (usually RTP/AVP although there are others such as RTP/SAVP);
- Port number (used by UDP or TCP for transport);
- Media type (audio, video, interactive whiteboard, and so forth);
- Media encoding scheme (PCM A-Law, MPEG II video, and so forth).

In addition, SDP contains information about the following:

- Subject of the session;
- Start and stop times;
- Contact information about the session.

Like SIP, SDP uses text coding. An SDP message is composed of a series of lines, called fields, whose names are abbreviated by a single lower-case letter, and are in a required order to simplify parsing. The set of SDP fields from RFC 4566 is shown in Table 13.1. The order in this table is the required order in SDP. Optional fields can be skipped, but must be in the correct order if present.

SDP was not designed to be easily extensible, and parsing rules are strict. The only way to extend or add new capabilities to SDP is to define a new attribute type. However, unknown attribute types can be silently ignored. An SDP

Table 13.1
SDP Fields

Field	Name	Mandatory/ Optional
`v=`	Protocol version number	m
`o=`	Owner/creator and session identifier	m
`s=`	Session name	m
`i=`	Session information	o
`u=`	Uniform Resource Identifier	o
`e=`	E-mail address	o
`p=`	Phone number	o
`c=`	Connection information	m
`b=`	Bandwidth information	o
`t=`	Timer session starts and stops	m
`r=`	Repeat times	o
`z=`	Time zone corrections	o
`k=`	Encryption key (deprecated)	o
`a=`	Attribute lines	o
`m=`	Media information	o
`a=`	Media attributes	o

parser must not ignore an unknown field, a missing mandatory field, or an out-of-sequence line. An example SDP message containing many of the optional fields is shown here:

```
v=0
o=johnston 2890844526 2890844526 IN IP4 43.32.1.5
s=IETF Update
i=This broadcast will cover the latest from the IETF
u=http://www.sipstation.com
e=Alan Johnston alan@avaya.com
p=+1-314-555-3333 (Daytime Only)
c=IN IP4 225.45.3.56/236
b=CT:144
t=2877631875 2879633673
m=audio 49172 RTP/AVP 0
a=rtpmap:0 PCMU/8000
m=video 23422 RTP/AVP 31
a=rtpmap:31 H261/90000
```

The general form of a SDP message is:

```
x=parameter1 parameter2 ... parameterN
```

The line begins with a single lower-case letter, for example, ***x***. There are never any spaces between the letter and the ***=***, and there is exactly one space between each parameter. Each field has a defined number of parameters. Each line ends with a CRLF. The individual fields will now be discussed in detail.

13.1.1 Protocol Version

The ***v=*** field contains the SDP version number. Because the current version of SDP is 0, a valid SDP message will always begin with ***v=0***.

13.1.2 Origin

The ***o=*** field contains information about the originator of the session and session identifiers. This field is used to uniquely identify the session. The field contains:

```
o=username session-id version network-type address-type address
```

The ***username*** parameter contains the originator's login or host or - if none. The ***session-id*** parameter is a Network Time Protocol (NTP) [4] timestamp or a random number used to ensure uniqueness. The ***version*** is a numeric field that is increased for each change to the session, also recommended to be a NTP timestamp. The ***network-type*** is always ***IN*** for Internet. The ***address-type*** parameter is either ***IP4*** or ***IP6*** for IPv4 or IPv6 address either in dotted decimal form or a fully qualified host name.

13.1.3 Session Name and Information

The ***s=*** field contains a name for the session. It can contain any nonzero number of characters. The optional ***i=*** field contains information about the session. It can contain any number of characters.

13.1.4 URI

The optional ***u=*** field contains a uniform resource indicator (URI) with more information about the session.

13.1.5 E-Mail Address and Phone Number

The optional ***e=*** field contains an e-mail address of the host of the session. If a display name is used, the e-mail address is enclosed in **<>**. The optional ***p=*** field contains a phone number. The phone number should be given in globalized format, beginning with a +, then the country code, a space or −, then the local number. Either spaces or - are permitted as spacers in SDP. A comment may be present inside brackets ***( )***.

13.1.6 Connection Data

The ***c=*** field contains information about the media connection. The field contains:

```
c=network-type address-type connection-address
```

The network-type parameter is defined as ***IN*** for the Internet. The address type is defined as ***IP4*** for IPv4 addresses and ***IP6*** for IPv6 addresses. The connection-address is the IP address or host that will be sending the media packets, which could be either multicast or unicast. If multicast, the connection-address field contains:

```
connection-address=base-multicast-address/ttl/number-of-addresses
```

where ***ttl*** is the time-to-live value, and ***number-of-addresses*** indicates how many contiguous multicast addresses are included starting with the ***base-multicast-address***.

13.1.7 Bandwidth

The optional **`b=`** field contains information about the bandwidth required. It is of the form:

```
b=modifier:bandwidth-value
```

The **`modifier`** is either **`CT`** for conference total or **`AS`** for application specific. **`CT`** is used for multicast session to specify the total bandwidth that can be used by all participants in the session. **`AS`** is used to specify the bandwidth of a single site. The **`bandwidth-value`** parameter is the specified number of kilobytes per second.

13.1.8 Time, Repeat Times, and Time Zones

The **`t=`** field contains the start time and stop time of the session.

```
t=start-time stop-time
```

The times are specified using NTP [4] timestamps. For a scheduled session, a **`stop-time`** of zero indicates that the session goes on indefinitely. A **`start-time`** and **`stop-time`** of zero for a scheduled session indicates that it is permanent. The optional **`r=`** field contains information about the repeat times that can be specified in either in NTP or in days (**`d`**), hours (**`h`**), or minutes (**`m`**). The optional **`z=`** field contains information about the time zone offsets. This field is used if a reoccurring session spans a change from daylight savings to standard time, or vice versa.

13.1.9 Encryption Keys

The optional **`k=`** field was used to carry encryption keys. However, its use is no longer recommended and was included in RFC 4566 for parser compatibility reasons. Instead, **`a=crypto`** or **`a=key-mgt`** should be used, whose use is described in Chapter 14.

13.1.10 Media Announcements

The optional **`m=`** field contains information about the type of media session. The field contains:

```
m=media port transport format-list
```

The **`media`** parameter is either **`audio`**, **`video`**, **`text`**, **`application`**, **`message`**, **`image`**, or **`control`**. The **`port`** parameter contains the port number. The **`transport`** parameter contains the transport protocol or the RTP profile used. The set of defined RTP profiles is in Table 15.3. The ***format-list*** contains more information about the media. Usually, it contains media payload types defined in RTP audio video profiles. More than one media payload type can be listed, allowing multiple alternative codecs for the media session. For example, the following media field lists three codecs:

```
m=audio 49430 RTP/AVP 0 6 8 99
```

One of these three codecs can be used for the audio media session. If the intention is to establish three audio channels, three separate media fields would be used. For non-RTP media, Internet media types should be listed in the ***format-list***. For example,

```
m=application 52341 udp wb
```

could be used to specify the ***application/wb*** media type. Common SDP media types are listed in Table 13.2.

13.1.11 Attributes

The optional ***a=*** field contains attributes of the preceding media session. This field can be used to extend SDP to provide more information about the media. If not fully understood by a SDP user, the attribute field can be ignored. There can be one or more attribute fields for each media payload type listed in the media

Table 13.2
Common SDP Media Types

Example	Type	Specificiation
m=audio 49122 RTP/AVP 0	Audio media, also used for telephone-events (DTMF)	RFC 3551
m=video 52134 RTP/SAVP 24	Video media	RFC 3551
m=text 11000 RTP/AVP 98	Real-time text (T.140)	RFC 4103
m=application 12454 wb udp *m=application 3422 TCP/TLS/BFCP ** *m=application 554 TCP/RTSP rtsp*	Application media, used for white board (wb), BFCP, RTSP, and others	RFC 4566
*m=message 12763 TCP/MSRP **	Message media for MSRP	RFC 4975
m=image 54111 TCP t38	Fax (T.38) Note: Fax can also use a ***m=audio*** media type	RFC 3362 RFC 4612
m=control	Control media	RFC 2327

field. For the media line example in Section 13.1.9, the following three attribute fields could follow the media field:

```
a=rtpmap:0 PCMU/8000
a=rtpmap:6 DVI4/16000
a=rtpmap:8 PCMA/8000
a=rtpmap:99 iLBC
```

Other attributes are shown in Table 13.3. Full details of the use of these attributes are in the standard document [2]. The details of the iLBC (Internet low bit rate) codec are in [5].

Attributes can be either session level or media level in SDP. Session level means that the attribute is listed before the first media line in the SDP. If this is the case, the attribute applies to all the media lines below it. Media level means it is listed after a media line. In this case, the attribute only applies to this particular media stream. SDP can include both session level and media level attributes. If the same attribute appears as both, the media level attribute overrides the session level attribute for that particular media stream.

Note that the connection data field can also be session level or media level. There are three possibilities:

Table 13.3
SDP Attribute Values Defined in RFC 4566

Attribute	Name
`a=rtpmap:`	RTP/AVP list.
`a=fmtp:`	Format transport.
`a=ptime:`	Length of time in milliseconds for each packet.
`a=maxptime:`	Maximum ptime.
`a=cat:`	Category of the session.
`a=keywds:`	Keywords of session.
`a=tool:`	Name of tool used to create SDP.
`a=orient:`	Orientation for whiteboard sessions.
`a=type:`	Type of conference.
`a=charset:`	Character set used for subject and information fields.
`a=sdplang:`	Language for the session description.
`a=lang:`	Default language for the session.
`a=framerate:`	Maximum video frame rate in frames per second.
`a=quality:`	Suggests quality of encoding.
`a=direction:`	Direction for symmetric media.
`a=inactive`	Inactive mode.
`a=recvonly`	Receive only mode.
`a=sendrecv`	Send and receive mode.
`a=sendonly`	Send only mode.

- A single `c=` field at the session level. This is the mos t common case.
- A session level `c=` field and some media level `c=` fields.
- Each media level field with no session level stream.

The same rules for attributes apply when both session and media level `c=` fields are present; the media field overrides the session level for that particular media stream.

13.2 SDP Extensions

There are a number of SDP extensions that have been defined. Common ones are summarized in Table 13.4.

The RTCP IP address and port attribute, `a=rtcp` [6] is covered in Chapter 10. The `a=setup` and `a=connection` attributes are used for connection oriented media, such as TCP. Section 8.5.2 shows the use of these attributes in establishing MSRP sessions. Another example is shown below of Binary Floor Control

Table 13.4
Common SDP Extensions

Attribute	Name	Reference
`a=rtcp`	Port and IP address for RTCP [6]	RFC 3605
`a=mid` `a=group`	Media session identifier and grouping of media streams [7]	RFC 3388
`a=setup` `a=connection`	Connection-oriented media using as TCP transport [8]	RFC 4145
`a=key-mgt`	Key management for MIKEY [9]	RFC 4567
`a=crypto`	Key management for SRTP [10]	RFC 4568

`a=floorctrl`
`a=confid`
`a=userid`
`a=floorid`
	Binary Floor Control Protocol (BFCP) information [11]	RFC 4583

`a=fingerprint`	Connection-oriented media using TLS [12]	RFC 4572
`a=label`	Media label [13]	RFC 4574

`a=accept-types`
`a=accept-wrapped-types`
`a=max-size`
`a=path`
	Message Session Relay Protocol (MSRP) information [14]	RFC 4975

`a=ice-pwd`
`a=ice-ufrag`
`a=ice-lite`
`a=ice-mismatch`
`a=ice-options`
	Interactive connectivity establishment (ICE) [15]	[15]

`a=chatroom`	Chat room name for MSRP	[16]

Protocol (BFCP) [11] session establishment, which shows the use of many of these SDP attributes. The first **m=** media line is for a BFCP stream running over TLS over TCP. The **a=connection:new** indicates that a new TCP connection needs to be opened and that this endpoint will do a passive open (the other endpoint will do the active open). The **a=fingerprint** contains a fingerprint of the certificate to be exchanged during the TLS handshake, as described in Section 14.6 the **a=confid** and **a=userid** attributes contain the conference ID and user ID of the user. The **a=floorid** attributes indicate that floor 1 is associated with **a=label:1**, which is associated with the **m=audio** stream while floor 2 is associated with **a=label:2,** which is associated with the **m=video** stream.

```
v=0
o=bob 2808844564 2808844564 IN IP4 130.43.2.1
s=
t=0 0
c=
m=application 54052 TCP/TLS/BFCP *
a=setup:passive
a=connection:new
a=fingerprint:SHA-1 AD:9B:B1:3F:72:18:00:3B:54:02:12:DF:3E:5F:49
:1B:19:E5:DC:AB
a=floorctrl:s-only
a=confid:38921838776
a=userid:bob
a=floorid:1 m-stream:1
a=floorid:2 m-stream:2
m=audio 54026 RTP/AVP 0
a=label:1
m=video 54042 RTP/AVP 31
a=label:2
```

13.3 The Offer Answer Model

The use of SDP with SIP is given in the SDP offer answer RFC 3264 [17]. The default message body type in SIP is **application/sdp**. The calling party lists the media capabilities that they are willing to receive in SDP, usually in either an **INVITE** or in an **ACK**. The called party usually lists their media capabilities in the **200 OK** response to the **INVITE**. More generally, offers or answers may be in **INVITE**s, **PRACK**s, or **UPDATE**s or in reliably sent **18x** or **200** responses to these methods.

Because SDP was developed with scheduled multicast sessions in mind, many of the fields have little or no meaning in the context of dynamic sessions established using SIP. In order to maintain compatibility with the SDP protocol, however, all required fields are included. A typical SIP use of SDP includes the version, origin, subject, time, connection, and one or more media and attribute fields is shown in Table 13.1. The subject and time fields are not used by SIP but are included for compatibility. In the SDP standard, the subject field is a required field and must contain at least one character, suggested to be **s=-** if there is no subject. The time field is usually set to **t=0 0**.

SIP uses the connection, media, and attribute fields to set up sessions between UAs. The origin field has limited use with SIP. Usually, the **session-id** is kept constant throughout a SIP session and the **version** is incremented each time the SDP is changed. If the SDP being sent is unchanged from that sent previously, the **version** is kept the same.

Because the type of media session and codec to be used are part of the connection negotiation, SIP can use SDP to specify multiple alternative media types and to selectively accept or decline those media types. The offer answer specification, RFC 3264 [17], recommends that an attribute containing **a=rtpmap:** be used for each media field. A media stream is declined by setting the port number to zero for the corresponding media field in the SDP response. In the following example, the caller Tesla wants to set up an audio and video call with two possible audio codecs and a video codec in the SDP carried in the initial **INVITE:**

```
v=0
o=Tesla 2890844526 2890844526 IN IP4 lab.high-voltage.org
s=-
c=IN IP4 100.101.102.103
t=0 0
m=audio 49170 RTP/AVP 0 8
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
m=video 49172 RTP/AVP 32
a=rtpmap:32 MPV/90000
```

The codecs are referenced by the RTP/AVP profile numbers 0, 8, and 32. The called party Marconi answers the call, chooses the second codec for the first media field, and declines the second media field, only wanting a PCM A-Law audio session.

```
v=0
o=Marconi 2890844526 2890844526 IN IP4 tower.radio.org
s=-
c=IN IP4 200.201.202.203
t=0 0
m=audio 60000 RTP/AVP 8
a=rtpmap:8 PCMA/8000
m=video 0 RTP/AVP 32
```

If this audio-only call is not acceptable, then Tesla would send an **ACK** then a **BYE** to cancel the call. Otherwise, the audio session would be established and RTP packets exchanged. As this example illustrates, unless the number and order of media fields is maintained, the calling party would not know for certain which media sessions were being accepted and declined by the called party.

The offer/answer rules are summarized in the following sections.

13.3.1 Rules for Generating an Offer

An SDP offer must include all required SDP fields (this includes `v=`, `o=`, `s=`, `c=`, and `t=`). It usually includes a media field (`m=`) but it does not have to. The media lines contain all codecs listed in preference order. The only exception to this is if the endpoint supports a huge number of codecs, the most likely to be accepted or most preferred should be listed. Different media types include audio, video, text, MSRP, BFCP, and so forth.

13.3.2 Rules for Generating an Answer

An SDP answer to an offer must be constructed according to these rules. The answer must have the same number of `m=` lines in the same order as the answer. Individual media streams can be declined by setting the port number to 0. Streams are accepted by sending a nonzero port number. The listed payloads for each media type must be a subset of the payloads listed in the offer. Note that for dynamic payloads, the same dynamic payload number does not need to be used in each direction. Usually, only a single payload is selected. More than one may be selected, but endpoints doing this must be capable of dynamically switching between them without signaling. Since many simple endpoints can only have one codec running at a time, this should be avoided. One common exception is to accept a media codec and also telephone-events (Section 12.7). This allows the codec to be used except when a DTMF key is pressed when a telephone-events payload is used.

13.3.3 Rules for Modifying a Session

Either party can initiate another offer/answer exchange to modify the session. When a session is modified, the following rules must be followed. The origin (`o=`) line version number must either be the same as the last one sent, which indicates that this SDP is identical to the previous exchange, or it may be incremented by one, which indicates new SDP that must be parsed. The offer must include all existing media lines and they must be sent in the same order. Additional media streams can be added to the end of the `m=` line list. An existing media stream can be deleted by setting the port number to 0. This media line must remain in the SDP in this and all future offer/answer exchanges for this session. For an existing media stream, any aspect can be changed.

13.3.4 Special Case—Call Hold

One party in a call can temporarily place the other on hold (i.e., suspending the media packet sending). This is done by sending an `INVITE` with an identical SDP to that of the original `INVITE` but with `a=sendonly` attribute present. The call is made active again by sending another **INVITE** with the **a=sendrecv** attribute present. (Note that older RFC 2543 compliant UAs may initiate hold using **c=0.0.0.0**.) For further examples of SDP use with SIP, see the SDP offer answer examples document [18].

13.4 Static and Dynamic Payloads

The payload type (PT) is used to identify the media codec in the media line of SDP as described in Section 13.1.10. This same payload type is also carried in individual RTP media packets sent during the media session. RFC 3551 defines some *static* payload types. These payloads are considered static because a given payload number defined in the specification always refers to that particular codec. For example, PT 0 for audio always means G.711 PCM codec. The use of **a=rtpmap** attribute for static payloads is optional, although it is considered good practice to include it. However, static payloads are no longer allocated by the IETF. Instead, all new codecs must make use of *dynamic* payload types. Dynamic payload types are in the range of 96–127. Payloads in this range do not refer to a particular codec; instead the required **a=rtpmap** attribute must be used to indicate the payload. There are a number of rules associated with the use of dynamic payloads in the SDP offer answer exchange. They are:

- Dynamic payloads must be negotiated with SDP.
- The **a=rtpmap** attribute is mandatory.
- Dynamic payload numbers cannot be redefined within a session.
- Dynamic payload numbers do not need to be the same in both directions of a bidirectional session.

The last rule means that it is possible that payload 97 means one codec in one direction but another codec in a different direction.

13.5 SIP Offer Answer Exchanges

The main offer answer exchanges with SIP are in the **INVITE/200 OK** exchange or in the **200 OK/ACK** exchange, if the **INVITE** did not contain an offer. There are other offer/answer modes, summarized in Table 13.5, which is taken from [19]. Support of the specification listed implies that the user agent supports this additional offer/answer exchange mode.

Table 13.5
SIP Offer/Answer Exchange Modes

Offer	Answer	Specification
`INVITE`	`2xx` to `INVITE`	RFC 3261
`2xx` to `INVITE`	`ACK`	RFC 3261
`INVITE`	Reliable `1xx` to `INVITE`	RFC 3262
Reliable `1xx` to `INVITE`	`PRACK`	RFC 3262
`PRACK`	`200` to `PRACK`	RFC 3262
`UPDATE`	`2xx` to `UPDATE`	RFC 3311

Source: [19].

13.6 Conclusion

This chapter has covered the use of SDP in the Offer/Answer Protocol to negotiate the establishment and modification of media sessions. Core SDP and the Offer/Answer Protocol allow basic media sessions to be established. Some SDP extensions are required for more advanced media setup and control.

13.7 Questions

Q13.1 Create an SDP offer for Bob offering audio and video with the following audio codecs: iLBC, GSM and video codecs: MPV, and H.261. Bob wants to receive audio media on port 60322, video on port 60324, and RTCP on port 60326. Bob would prefer a packetization time of 30 ms for audio.

Q13.2 Create an SDP answer for Alice to Bob's offer from the previous question, accepting video but declining audio. You can choose whichever ports and codecs you like.

Q13.3 Find the three syntax errors in this SDP example.

```
v=0
o=alice 289084526 28904529 IP4 231.3.43.1
s=-
c=IN IP4 231.3.43.1
m=audio 49170 RTP/AVP 0 97 98
a=rtpmap:97 iLBC/8000
```

Q13.4 Create an SDP offer by Alice that could have resulted in the following SDP answer.

```
v=0
o=bob 2808844564 2808844564 IN IP4 130.43.2.1
s=-
t=0 0
```

```
m=audio 49174 RTP/AVP 0
c=IN IP4 130.43.2.1
a=rtpmap:0 PCMU/8000
a=recvonly
m=text 49176 RTP/AVP 96
c=IN IP4 130.43.2.2
a=rtpmap:96 t140/1000
```

Q13.5 Indicate the IP address and port number associated with each of the three media streams.

```
v=0
o=Tesla 2890844526 2890844526 IN IP4 lab.high-voltage.org
s=-
c=IN IP4 100.101.102.103
t=0 0
m=audio 49170 RTP/AVP 0 8
c=IN IP4 101.102.103.106
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
m=video 49172 RTP/AVP 34
a=rtpmap:34 H263/90000
m=video 53132 RTP/AVP 26
c=IN IP4 100.102.103.4
a=rtpmap:26 JPEG/90000
```

Q13.6 Describe in words the offer and answer in the SDP below.

Offer:

```
v=0
o=alice 2890844526 2890844526 IN IP4 host.atlanta.example.com
s=
c=IN IP4 host.atlanta.example.com
t=0 0
m=audio 49170 RTP/AVP 0 8 97
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:97 iLBC/8000
m=video 51372 RTP/AVP 31 32
a=rtpmap:31 H261/90000
a=rtpmap:32 MPV/90000
```

Answer:

```
v=0
o=bob 2808844564 2808844564 IN IP4 host.biloxi.example.com
s=
c=IN IP4 host.biloxi.example.com
t=0 0
m=audio 49174 RTP/AVP 0
a=rtpmap:0 PCMU/8000
m=video 49172 RTP/AVP 32
c=IN IP4 otherhost.biloxi.example.com
a=rtpmap:32 MPV/90000
```

Q13.7 Find two errors in the offer/answer exchange:

Offer:

```
v=0
o=Tesla 2890844526 2890844526 IN IP4 lab.high-voltage.org
s=-
c=IN IP4 100.101.102.103
t=0 0
m=audio 49170 RTP/AVP 0 8
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
m=video 49172 RTP/AVP 32
a=rtpmap:32 MPV/90000
```

Answer:

```
v=0
o=Marconi 2890844526 2890844526 IN IP4 tower.radio.org
s=-
c=IN IP4 200.201.202.203
t=0 0
m=audio 60000 RTP/AVP 98
a=rtpmap:98 iLBC/8000
```

Q13.8 Is it permissible to define payload 98 as iLBC codec in one direction and payload 97 as iLBC in the other direction?

Q13.9 A user agent supports RFC 3261 and RFC 3262, but does not support RFC 3311. Which offer/answer modes does this user agent support? Which is likely to be the most commonly used?

Q13.10 For the offer/answer exchange below, generate a new offer answer exchange between Marconi and Tesla where Marconi puts the audio stream on hold.

Offer:

```
v=0
o=Tesla 2890844526 2890844526 IN IP4 lab.high-voltage.org
s=-
c=IN IP4 100.101.102.103
t=0 0
m=audio 49170 RTP/AVP 0 8
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
m=video 49172 RTP/AVP 32
a=rtpmap:32 MPV/90000
```

Answer:

```
v=0
o=Marconi 2890844526 2890844526 IN IP4 tower.radio.org
s=-
c=IN IP4 200.201.202.203
t=0 0
m=audio 60000 RTP/AVP 8
a=rtpmap:8 PCMA/8000
m=video 0 RTP/AVP 32
```

References

[1] Handley, M., and V. Jacobson, "SDP: Session Description Protocol," RFC 2327, April 1998.

[2] Handley, M., C. Perkins, and E. Whelan, "Session Announcement Protocol," RFC 2974, October 2000.

[3] Handley, M., V. Jacobson, and C. Perkins, "SDP: Session Description Protocol," RFC 4566, July 2006.

[4] Mills, D., "Network Time Protocol (Version 3): Specification, Implementation, and Analysis," RFC 1305, March 1992.

[5] Duric, A., and S. Andersen, "Real-Time Transport Protocol (RTP) Payload Format for Internet Low Bit Rate Codec (iLBC) Speech," RFC 3952, December 2004.

[6] Huitema, C., "Real Time Control Protocol (RTCP) Attribute in Session Description Protocol (SDP)," RFC 3605, October 2003.

[7] Camarillo, G., et al., "Grouping of Media Lines in the Session Description Protocol (SDP)," RFC 3388, December 2002.

[8] Yon, D., and G. Camarillo, "TCP-Based Media Transport in the Session Description Protocol (SDP)," RFC 4145, September 2005.

[9] Arkko, J., et al., "Key Management Extensions for Session Description Protocol (SDP) and Real Time Streaming Protocol (RTSP)," RFC 4567, July 2006.

[10] Andreasen, F., M. Baugher, and D. Wing, "Session Description Protocol (SDP) Security Descriptions for Media Streams," RFC 4568, July 2006.

[11] Camarillo, G., "Session Description Protocol (SDP) Format for Binary Floor Control Protocol (BFCP) Streams," RFC 4583, November 2006.

[12] Lennox, J., "Connection-Oriented Media Transport over the Transport Layer Security (TLS) Protocol in the Session Description Protocol (SDP)," RFC 4572, July 2006.

[13] Levin, O., and G. Camarillo, "The Session Description Protocol (SDP) Label Attribute," RFC 4574, August 2006.

[14] Campbell, B., R. Mahy, and C. Jennings, "The Message Session Relay Protocol (MSRP)," RFC 4975, September 2007.

[15] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols," draft-ietf-mmusic-ice-19 (work in progress), October 2007.

[16] Niemi, A., M. Garcia-Martin, and G. Sandbakken, "Multi-Party Chat Using the Message Session Relay Protocol (MSRP)," draft-ietf-simple-chat-03 (work in progress), October 2008.

[17] Rosenberg, J., and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)," RFC 3264, June 2002.

[18] Johnston, A., and R. Sparks, "Session Description Protocol (SDP) Offer/Answer Examples," RFC 4317, December 2005.

[19] Sawada, T., and P. Kyzivat, "SIP (Session Initiation Protocol) Usage of the Offer/Answer Model," Internet-Draft, draft-ietf-sipping-sip-offeranswer-10 (work in progress), January 2009.