VoIP Protocols: SIP and H.323

By Kyzer Davis, Paul Giralt, Patrick Kinane, Gonzalo Salgueiro.
Sample Chapter is provided courtesy of Cisco Press.
Date: Nov 20, 2019.

Contents

Chapter Description

In this sample chapter from *CCNP Collaboration Call Control and Mobility CLACCM 300-815 Official Cert Guide*, you will review the function components of Session Initial Protocol (SIP), exam Session Description Protocol (SDP) fundamentals, and examine the H.323 communication protocol.

From the Book

Introduction to SDP

In the earlier days of Internet telephony, the process of setting up a call was very cumbersome and drawn out—very unlike the seamless call setup we experience today. To set up a communication session in the early days, participants had to do the following:

Step 1. The caller would bootstrap an audio or video application at a specific port number and IP address.

Step 2. The caller would inform the callee of the details of the port number and IP address over a PSTN line.

Step 3. The callee would fire up a local audio or video application and inform the caller of the IP address and port number on their end.

While this process was acceptable for the occasional calls made over packet networks for the purpose of research and demonstration, it clearly would not find acceptance if Internet telephony were to scale. Protocols were needed to set up, modify, and tear down communication sessions, and these protocols needed to provide enough information to allow participation within the communication session. SIP, as a call control protocol, is adept at setting up and tearing down communication sessions. However, it does not provide participants any information about the details of the communication session (for example, the media types supported and the IP/port pair for media). As a result, SIP relies on a peer protocol to facilitate advertisement and negotiation of media capabilities.

Session Description Protocol (SDP), originally defined in RFC 2327 (and later updated in RFC 4566), was designed to provide session details (such as the media types, media codec, and IP/port pair for media) and session metadata (such as the purpose of the session and the originator of the session) to participants. SDP is strictly a description protocol and it is leveraged by higher-level protocols such as Session Announcement Protocol (SAP), Session Initiation Protocol (SIP), Media Gateway Control Protocol (MGCP), Real Time Streaming Protocol (RTSP), Multipurpose Internet Mail Extensions (MIME), and Hypertext Transfer Protocol (HTTP).

SDP is completely textual and rigid in terms of formatting. Unlike H.323, it does not use binary encoding, such as ASN.1. This choice was made deliberately so that SDP could be leveraged by several protocols and to ensure that malformed SDP bodies could be easily identified and discarded. The formatting of SDP bodies is mostly in UTF-8. SDP bodies contain a number of textual lines that are each either classified as fields or as attributes. A field is separated from the next one by a carriage return/line feed (CRLF) sequence. The format of each field is as follows:

```
<type>=<value>[CRLF]
```

Attributes are the primary means of extending SDP. Over time, to enable several application use cases and to enable smooth interoperability between communicating entities, many SDP attributes were defined and standardized. (At the time of this writing, there are a few new SDP attributes in the process of being standardized by the IETF.) Attributes can be of two types:

■ **Property attributes:** These attributes are in the format a=<flag>. A property attribute conveys a simple Boolean meaning for media or the session.

■ **Value attributes:** These attributes are in the format a=<attribute>:<value>.

The primary purpose of an SDP body is to always ensure that the participants are provided sufficient information to join a communication session. Accordingly, SDP bodies are classified into three description levels:

■ Session description

■ Time description

■ Media description

The session description consists of a number of fields and optional attributes that provide details around the session, such as the name of the session, the originator of the session, and bandwidth constraints for the session. The session description can optionally contain attributes as well.

Communication sessions can either be unbounded or bounded in time. SDP time descriptions specify when communication sessions are active by using the timing (t=) field. The timing field has the following format:

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

This field is self-explanatory: *start-time* and *stop-time* simply encode the time when the session starts and ends, respectively. *start-time* and *stop-time* are expressed in decimal representations of Network Time Protocol (NTP) time values in seconds since 1900. The encoding of the *start-time* and *stop-time* determines whether the communication session is bounded, unbounded, or permanent. A bounded session has an explicit *start-time* and *stop-time*. An unbounded session does not have a *stop-time*, whereas a permanent session does not have a *start-time* or *stop-time*. The encoding of the timing field is useful for multicast communication sessions. For unicast sessions, the timing field must be encoded to specify a permanent session (t=0 0).

The media description section of SDP bodies provides sufficient detail about the media and transport characteristics of the communication session. Participants use this information to join a multicast session or negotiate common capabilities for unicast sessions. The media description section includes the following information:

■ The media types (for example, audio, video, application, image)

■ The transport protocol (for example, RTP)

■ The media formats for different media types (for example, G.711, H.264)

■ Optionally, the IP address and port pair for media

Fields and attributes in SDP bodies can be either mandatory or optional. In either case, they must follow the rigid ordering structure shown in Table 2-5.

Table 2-5 Fields and Attributes in SDP Bodies

Field/Attribute Description		Mandatory or Optional?
Session Description		
v=	Protocol version	Mandatory
o=	Originator and session identifier	Mandatory
s=	Session name	Mandatory
i=	Session information	Optional
u=	URI of description	Optional
e=	Email address	Optional
p=	Phone number	Optional
c=	Connection information; not required if included in all media	Optional
b=	Zero or more bandwidth information lines	Optional
z=	Time zone adjustments	Optional
k=	Encryption key	Optional
a=	Zero or more session attribute lines	Optional
Time Description		
t=	Time the session is active	Mandatory
r=	Zero or more repeat times	Optional
Media Description (If Present)		
m=	Media name and transport address	Mandatory
i=	Media title	Optional
c=	Connection information; optional if included at the session level	Optional
b=	Zero or more bandwidth information lines	Optional
k=	Encryption key	Optional
a=	Zero or more media attribute lines	Optional

Field/Attribute Description	Mandatory or Optional?
Session Description	

SDP fields and attributes can appear at two levels:

■ Session level

■ Media level

The session-level section of SDP bodies provides default values for various fields that are to be used and interpreted. For example, if a user agent wants to use the same media connection IP address for all media streams within the session, it can encode an SDP body with a session-level description of media connection information. However, if further granularity is required on a per-media-stream basis, the user agent can encode an SDP body with one or several media-level descriptions. Example 2-3 is a snippet of an SDP body carried within a SIP message. (The actual SIP message is omitted from this example for brevity.)

Example 2-3 ***SDP Body Carried Within a SIP Message***

```
v=0
o=CiscoSystemsSIP-GW-UserAgent 1597 5834 IN IP4 10.94.64.12
s=SIP Call
c=IN IP4 10.1.1.1
t=0 0
a=recvonly
m=audio 16590 RTP/AVP 8 101
a=rtpmap:8 PCMA/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15
a=ptime:20
m=video 51372 RTP/AVP 126
a=rtpmap:126 H264/90000
```

The session-level description starts with the v= line and continues until the first media-level section. Every media-level section is identified by an m= line and continues until the next m= line or until the end of the SDP body. As shown in Example 2-3, the media connection IP address (c=IN IP4 10.1.1.1) has only a session-level description, and it spans the audio and video stream. In addition to having a media connection information field, the direction attribute (a=recvonly) is specified for both the audio and video media streams. Session-level descriptions serve as default values to be interpreted and used if no corresponding media-level description(s) is available.

Example 2-4 is a snippet of an SDP body where the direction attribute has a session-level description and a media-level description. You should be aware that certain SDP fields and attributes can be present concurrently at different levels of the SDP body. When this occurs, the media-level field or attribute overrides the session-level field or attribute. So, in the case of the direction attribute appearing twice in Example 2-4, the media-level description of the direction attribute is given higher precedence.

Example 2-4 ***SDP Body with Session- and Media-Level Definitions for the Direction Attribute***

```
v=0
o=CiscoSystemsSIP-GW-UserAgent 1597 5834 IN IP4 10.94.64.12
s=SIP Call
c=IN IP4 10.1.1.1
t=0 0
a=recvonly
m=audio 16590 RTP/AVP 8 101
a=rtpmap:8 PCMA/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15
a=sendrecv
a=ptime:20
```

The Offer/Answer Framework

SDP was originally conceived as a way to describe multicast sessions over Mbone (short for *multicast backbone*). SDP scales really well for multicast, as there is a unified view of the session for all participants. For example, for multicast communication sessions, each participant requires a single media address and port to join a communication session. While SDP has the capability of describing unicast communication sessions, it is a slightly

more challenging proposition than describing multicast sessions. For a unicast session between two participants, each participant has a localized view of the session; each participant has its own media IP address and port pair, its own set of supported media types, and its own set of supported codecs per media type. To obtain a complete view of a unicast session, the participants must exchange information elements and agree on a common set of parameters. The SDP offer/answer model, defined in RFC 3264, provides such a framework for information exchange and parameter negotiation. To get a better understanding of the offer/answer framework, it is important to understand certain terms that are frequently referenced in subsequent sections:

- **Agent:** An entity involved in an offer/answer exchange
- **Answerer:** An agent that receives a session description that describes aspects of a plausible media communication session and responds with its own session description
- **Answer:** An SDP message sent from an answerer to an offerer
- **Offerer:** An agent that generates a session description to create or modify a session
- **Offer:** An SDP message sent by an offerer
- **Media Stream:** A single media instance in a communication session

Operation of the Offer/Answer Framework

The offer/answer exchange requires the existence of a stateful, higher-level protocol such as SIP that is capable of exchanging SDP bodies during call setup and/or modification. The protocol has to be stateful to maintain context around the exchange between an offerer and an answerer, as there may be several SDP exchanges during the course of a call. It is important for the higher-level protocol to accurately map requests and responses.

Generating the SDP Offer and Answer

The SDP offer/answer model begins with one of the user agents constructing an SDP body according to the guidelines of RFC 4566. You should realize that the initiator of a communication session (the user agent that sends the SIP INVITE) need not always be the one constructing the SDP offer. For example, for SIP delayed offer calls, the user agent being invited to a communication session is the one that constructs the SDP offer. Figure 2-4 shows the SIP three-way handshake for a delayed offer call versus the same handshake for an early offer call.

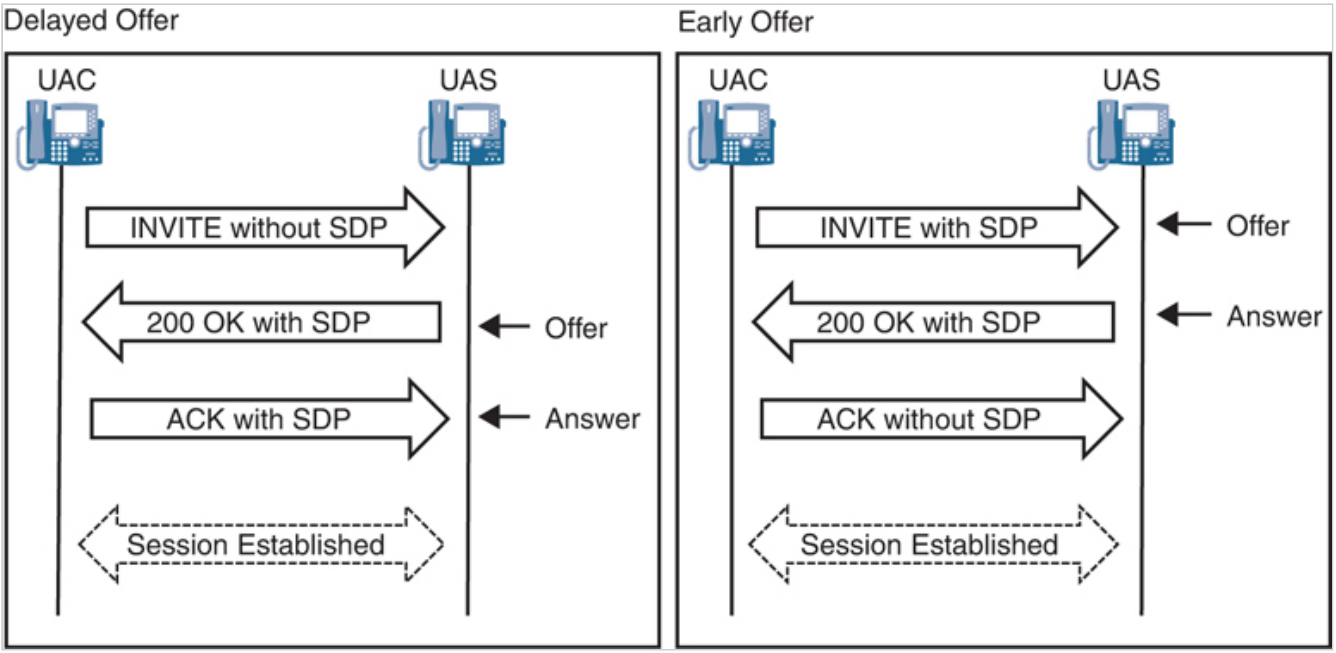

Figure 2-4 ***SIP Delayed Offer Versus SIP Early Offer***

Regardless of which user agent constructs the offer, the SDP body must consist of a session description, a time description, and a media description section. This strict encoding format ensures that the peer user agent is provided sufficient information to

participate in the communication session. The session description section contains all the mandatory fields (for example, v, o, s) as well as optional attribute values. For unicast sessions, the time description section must contain a timing field to indicate a permanent session (t=0 0). The media description section of the SDP offer can contain several media lines (m=), such that each media line corresponds to different media types or they correspond to the same media type or a combination of the two.

Each media line in the SDP body must encode sufficient information about the media stream to convey the following:

- The media type of the stream (for example, audio, video, image)
- The transport port and IP address of the media stream
- The list of media formats per media stream

The format of any media line within an SDP body is as follows:

```
m=<media> <port> <proto> <fmt list>
```

The <media> subfield indicates the media type, such as audio, video, image, and so on. The possible set of media types that can be advertised in SDP bodies is maintained in the Media Type registry of the Internet Assigned Numbers Authority.

The <port> subfield is used to advertise the port number on which media reception is expected. It is a common misconception that the port number signifies the port number from which media is sourced. Although most implementations utilize symmetric RTP, which does source RTP from the same port advertised by SDP for RTP reception, some implementations may utilize asymmetric RTP, which may use another source port. For media transport protocols such as RTP, the peer protocol Real-Time Transport Control Protocol (RTCP) allows participants to provide real-time media reception quality feedback. By default, RTCP is exchanged on the next higher port number following the RTP port number. If for some reason an application does not want to exchange RTCP on the next higher port number following RTP, it can explicitly indicate this by using the a=rtcp attribute. Example 2-5 demonstrates the use of the a=rtcp attribute in an SDP body.

Example 2-5 ***SDP Body Demonstrating the Use of the a=rtcp Attribute***

```
v=0
o=CiscoSystemsSIP-GW-UserAgent 1597 5834 IN IP4 10.94.64.12
s=SIP Call
c=IN IP4 10.1.1.1
t=0 0
a=recvonly
m=audio 16590 RTP/AVP 8 101
a=rtcp:53020
a=rtpmap:8 PCMA/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15
a=sendrecv
a=ptime:20
```

The <port> subfield is useful only when interpreted and used in conjunction with the connection data (c=) field. Without the connection data field, the remote user agent is only aware of a port number and has no information about the remote IP address. The connection data field can be scoped to include a session-level or media-level definition.

The <proto> subfield identifies the transport protocol for media. For media encapsulated over RTP, this subfield is set to RTP/AVP or, optionally, to RTP/SAVP for Secure RTP (SRTP). AVP stands for audio/video profile, and SAVP stands for secure audio/video profile.

The <fmt list> subfield specifies the media formats supported by the user agent generating the SDP body. The encoding of this subfield depends on the value of the <proto> subfield, which is either set to RTP/AVP or RTP/SAVP and includes a list of payload numbers (or sometimes only one payload number). For applications to discern the media format to which a given payload number corresponds, there is a list of payload number-to-media format mappings defined in the RTP audio/video profile. Table 2-6 lists a selection of these mappings for common audio codecs.

Table 2-6 Mapping Between Payload Numbers and Media Formats for Common Audio Codecs

Payload Type	Encoding Name	Clock Rate (Hz)
0	PCMU	8000

4	G723	8000
8	PCMA	8000
9	G722	8000
15	G728	8000
18	G729	8000

NOTE

A comprehensive list of the mappings of all payload numbers to media formats is maintained in an Internet Assigned Numbers Authority registry at https://www.iana.org/assignments/rtp-parameters/rtp-parameters.xhtml.

In the case of dynamic payload numbers (payload numbers between 96 and 127), there has to be an explicit mapping specified in the SDP body, using the a=rtpmap attribute. While it is not required to use the a=rtpmap attribute for static assignments already specified in the RTP audio/video profile, it seems to be the preferred formatting choice for most vendors.

To better understand this concept, see the sample SDP body provided in Example 2-6. The media line (m=) lists three static payload numbers: 0, 8, and 18. For a user agent that receives this SDP body, the interpretation of the static payload numbers 0, 8, and 18 is provided by the RTP audio/video profile and translates to PCMU, PCMA, and G729, respectively (refer to Table 2-6). Providing a mapping between these static payload numbers and their corresponding media formats via the a=rtpmap attribute is a redundant but nonetheless well adopted practice. For dynamic payload numbers, such as the OPUS codec utilizing payload type 114, the a=rtpmap attribute is required to explicitly provide a binding to the media format.

Example 2-6 ***SDP Body Demonstrating the Use of the a=rtpmap Attribute***

```
v=0
o=CiscoSystemsCCM-SIP 2828060 1 IN IP4 10.1.1.1
s=SIP Call
c=IN IP4 10.1.1.1
b=TIAS:64000
b=AS:64
t=0 0
m=audio 17236 RTP/AVP 0 8 18 114
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:18 G729/8000
a=rtpmap:114 opus/48000/2
a=fmtp:114  maxplaybackrate=16000;sprop-maxcapturerate=16000;
;stereo=0;sprop-stereo=0;usedtx=0
```

The a=fmtp attribute shown in Example 2-6 is used to encode media format–specific parameters. For example, when using the OPUS codec, many attributes can be utilized. One example defined here is the maximum average bitrate for the session via the maxaveragebitrate attribute. As the example shows, many attributes can be applied to a given media format on a single line by using the semicolon (;) as a separator. These attributes vary per media format, so refer to the applicable media format standard for a=fmtp usage.

Media lines have their interpretations tightly coupled with the SDP direction attribute. A direction attribute can have a session-level scope or a media-level scope. For unicast streams, the offerer can specify the directionality of a media stream by using the SDP direction attribute. Accordingly, a stream can be marked as sendonly, recvonly, inactive, or sendrecv. Table 2-7 summarizes the meaning of each Direction Attribute.

Table 2-7 SDP Direction Attributes Description

Direction	Direction of Media

Attribute	
sendonly	The sender wishes to only send media to its peer.
recvonly	The sender wishes to only receive media from its peer.
sendrecv	The sender wishes to send and receive media.
inactive	The sender wishes to set up the session but not transmit or receive media.

When the direction attribute has a sendrecv or recvonly value, it signifies the IP address and port number on which the sender would expect to receive media (RTP) from its peer. If the direction attribute is marked as sendonly, it indirectly signifies the IP address and port on which the sender (of the SDP) expects to receive RTCP but not RTP.

As mentioned previously, the IP address and port listed in the SDP offer does not signify the source address and port for RTP packets. Instead, it signifies the address and port on which the offerer expects to receive media.

If a user agent sets the direction attribute to inactive, it means that the user agent wants to simply establish a communication session without transmitting or receiving media. However, at a later time, the user agent can initiate a new SDP offer/answer exchange to update the direction attribute. Regardless of the value of the direction attribute, there is a continuous passage of RTCP traffic between communicating entities. While constructing an SDP body, if the user agent does not specify an explicit value for the direction attribute, it always defaults to sendrecv.

As mentioned earlier, the user agent constructing the SDP offer can include one or more media lines such that the media lines can correspond to the same media type, different media types, or a combination of the two. Conventionally, the offerer must use a valid, nonzero port number for each media line within the offer. This is because the use of port zero for a media line(s) within the offer has no useful semantics.

On receiving the offer, the answerer must construct an SDP body following the guidelines of RFC 4566: It must include a session description, a time description, and a media description. Even if there is absolute parity between the offer and the answer in terms of the media streams and media formats per stream, it is reasonable to assume that the answer will differ from the offer on certain aspects such as the IP address and port pair for media, support for SDP extensions, and so on. In such instances, the origin line (o=) of the answer must be different from that of the offer. The timing field in the answer must mirror the timing field in the offer. With regard to the media description, the constructed answer must follow several rules that are discussed in more detail in the following paragraphs.

While constructing the answer, the answerer must generate a response to each media line listed in the offer, and the number of media lines in the offer and answer must always be the same. If a given media type in the offer is not supported by the answerer, the answerer must reject the corresponding media line by setting the port number to zero. If an answerer rejects a media stream, there is no RTCP traffic exchanged for that media stream. Example 2-7 demonstrates an offer/answer exchange where the video media type is rejected by the answerer.

Example 2-7 ***SDP Offer/Answer Exchange for Disabling Video***

```
v=0
o=CiscoSystemsCCM-SIP 279932 1 IN IP4 172.18.110.91
s=SIP Call
c=IN IP4 14.50.214.107
b=AS:384
t=0 0
m=audio 49172 RTP/AVP 8
a=rtpmap:8 PCMA/8000
m=video 0 RTP/AVP 126
a=rtpmap:126 H264/90000
```

NOTE

If there are no media formats in common for all streams, the entire offer session is rejected.

As discussed previously, an offerer can set the direction attribute (at the session level or media level) to sendrecv, sendonly, recvonly, or inactive. Table 2-8 highlights the different ways in which the direction attribute can be set in an answer for unicast media sessions.

Table 2-8 Different Ways of Setting the Direction Attribute in an Answer

Direction Attribute in Offer	Direction Attribute in an Answer
sendonly	recvonly/inactive
recvonly	sendonly/inactive
sendrecv	sendrecv/sendonly/recvonly/inactive
inactive	inactive

For streams that are marked as recvonly in the answer, the answer must contain at least one media format that was listed in the offer. In addition, the answerer may include media formats not listed in the offer that the answerer is willing to receive. This is useful in scenarios where the offerer proceeds to modify the communication session at a later stage and includes an updated media format list.

For streams that the answerer marked as sendonly, the answer must contain at least one media format that was listed in the offer. For streams marked as sendrecv in the answer, the answer must list at least one media format that it is willing to use for both sending and receiving media. In such a situation, the answer might also list media formats that were not a part of the offer. Again, this is useful in scenarios where the offerer proceeds to modify the communication session at a later stage and includes an updated media format list. For streams marked as inactive in the answer, the media format list in the answer mirrors that in the offer.

NOTE

Media formats in the offer and answer are always listed in decreasing order of precedence, from left to right.

It is required for the answerer to use the a=rtpmap attribute for each media format to provide a payload number to the media format binding—regardless of whether the answer contains static or dynamic payload numbers. If a media format in the offer is described using the a=fmtp attribute, and that media format is echoed in the answer, the answerer must ensure that the same fmtp parameters are listed.

NOTE

The offer and answer can optionally include bandwidth and packetization interval attributes. For more on packetization intervals, see Chapter 3.

Media lines marked as sendonly and recvonly by the offerer have a reverse interpretation when accepted by the answerer. For example, consider an offer where the audio media line is marked as sendonly. When accepted by the answerer, the same audio media line has to be marked as recvonly. This ensures that the offer/answer exchange concludes with both user agents converging on a unified view of the communication session. In this

case, the offerer only transmits media packets, while the answerer only receives media packets. Of course, this assumes that the answerer does not set the stream as inactive.

Modifying a Session

During the course of a communication session, it is not uncommon for application interactions to require modification of session characteristics. These modifications could include changing the media formats, changing the value of the direction attribute, adding new media streams, and removing existing media streams, for instance. Some examples of scenarios where modifications occur are during supplementary service actions such as call hold, resume, transfer, and even conferencing. Nearly all aspects of a communication session can be modified. To effect a change or modification of session characteristics, the two user agents must engage in a new SDP offer/answer exchange. The high-level flow of modifying a communication session is depicted in Figure 2-5.

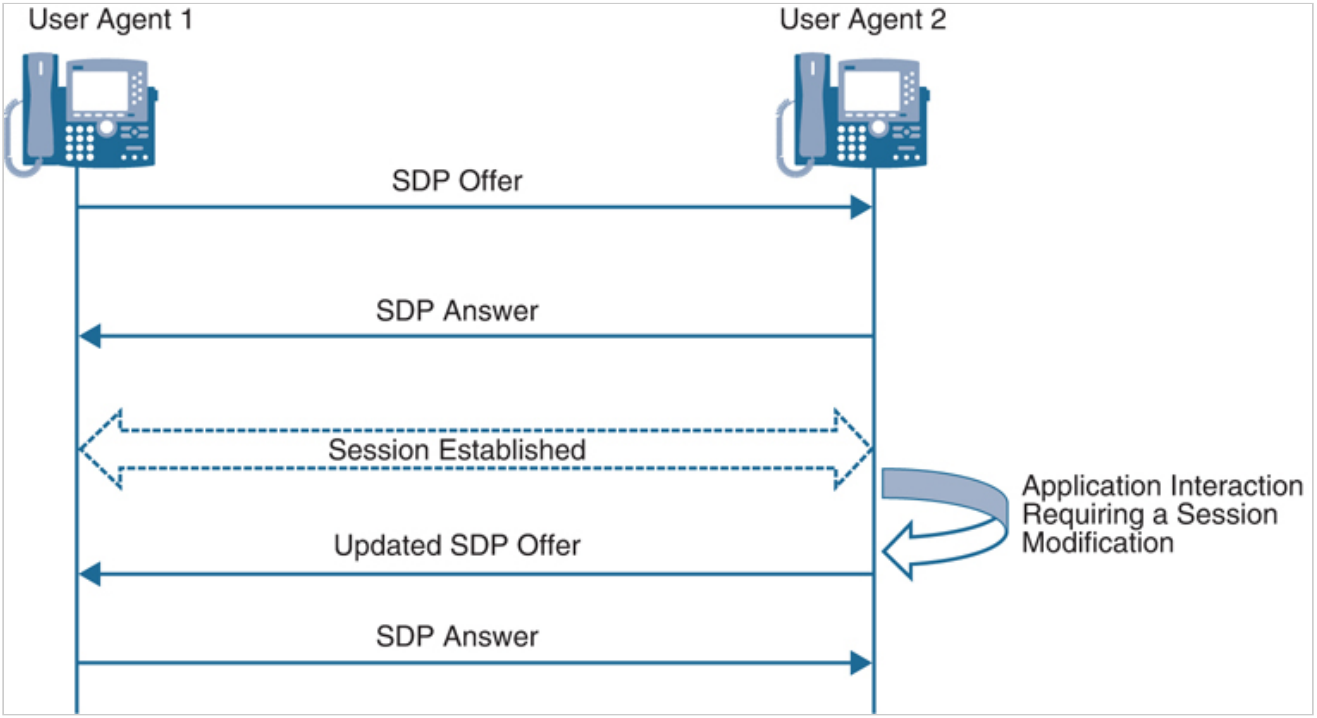

Figure 2-5 ***Modifying a Communication Session by Using the SDP Offer/Answer Exchange***

A user agent attempting to modify a communication session first constructs an updated SDP body whose content reflects the modifications required. These modifications can be simple or complex. Regardless of the degree of modification reflected in the SDP body, the user agent must increment the version number of the origin field (o=) by one. Example 2-6 highlights this concept.

The original SDP body in Example 2-8 contains the version number 5834. Sometime during the course of the communication session, the user agent requires a change to the list of media formats and accordingly proceeds to construct and transmit an updated SDP body. The updated SDP offer has its version number incremented by one and a modified media format list in the audio media line. This example also includes two very important SIP headers, which are included in the SIP message to describe and identify the contents of the SIP message body. For an SDP body, the Content-Type value application/sdp is used. Similarly, the Content-Length header field provides the total length of the SDP message body. If no message body is present in the SIP message, the Content-Length header field is set to 0. For the sake of brevity, the SIP header in Example 2-8 only displays the Content-Type and Content-Length SIP header fields.

Example 2-8 ***Incrementing the SDP Originator Version Number***

```
m=video 51372 RTP/AVP 126
a=rtpmap:126 H264/90000


[Modified SIP+SDP]

Content-Type: application/sdp
Content-Length: 227

v=0
o=CiscoSystemsSIP-GW-UserAgent 1597 5835 IN IP4 10.94.64.12
s=SIP Call
c=IN IP4 10.1.1.1
t=0 0
a=recvonly
m=audio 16590 RTP/AVP 0
a=rtpmap:0 PCMA/8000
a=ptime:20
m=video 51372 RTP/AVP 126
a=rtpmap:126 H264/90000
```

It is possible for a user agent to initiate a new SDP offer/answer exchange without changing the contents of the SDP body. While this exchange doesn't result in the modification of session characteristics, it could be used for reasons such as determining session freshness. If a new SDP body is identical to the previous SDP body, the version number must remain the same.

While generating an updated offer, the user agent must ensure that the number of media lines (m=) equals the number of media lines in the previous SDP body. In other words, if the previous SDP body had N media lines, the updated SDP body must contain at least N media lines. It is possible for the updated SDP body to contain more than N media lines since this is required when adding a new media line.

If SIP is the signaling protocol used to establish a media session (with SDP message body for media description), an existing session can be modified using either a re-INVITE or an UPDATE SIP message. Figure 2-6 illustrates both of these scenarios.

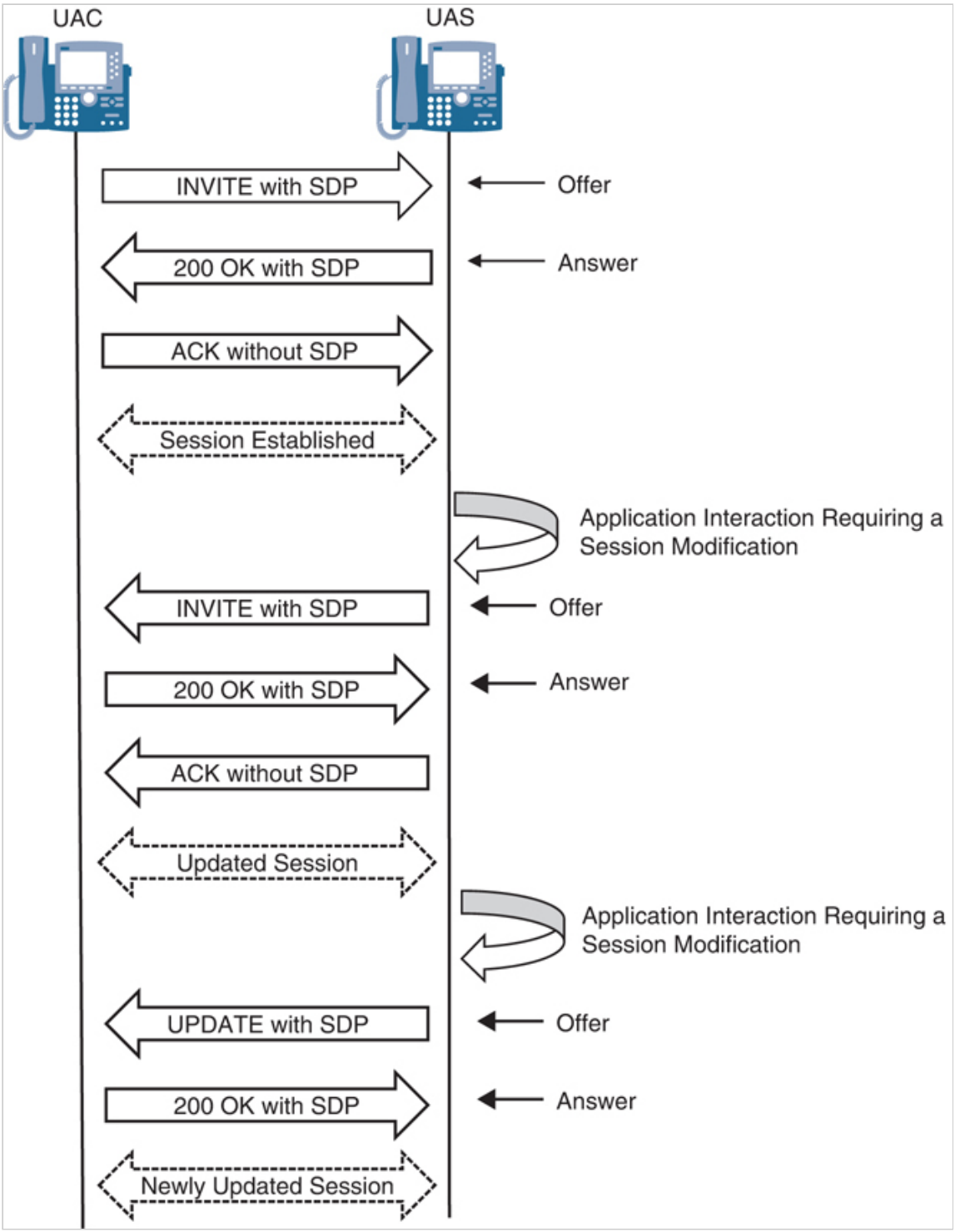

Figure 2-6 ***SIP re-INVITE and UPDATE Session Modifications***

The following steps occur in the example shown in Figure 2-6:

Step 1. The initial session is established using an early offer signaling exchange.

Step 2. An application interaction occurs and requires a change in media. This then triggers a mid-call re-INVITE, with a new offer modifying the original session.

Step 3. The UAS accepts the session modification and sends a 200 OK with an SDP body.

Step 4. An ACK to the 200 OK confirms the transaction and finishes the handshake.

Step 5. The session continues with the newly established session parameters until another application interaction occurs. At this point, the session is modified again—this time using an UPDATE SIP message with an SDP body containing an offer for the new session.

Step 6. The remote party accepts this UPDATE with an SDP body via a 200 OK, and an SDP body confirms the session modification.

Step 7. The session continues with the newly updated session parameters.

> **NOTE**
>
> This process of modifying the session through re-INVITE or UPDATE SIP message can continue as many times as needed by either user agent involved in the session. Also, note that the UPDATE transaction does not contain an ACK SIP message because the ACK is exclusive to the INVITE/re-INVITE transaction.

Adding a Media Stream

It is possible to add new media streams to a session by appending the appropriate media lines to an existing SDP body. For example, if an audio-only communication session is established between two participants and one of the participants wants to escalate the session to include video, that participant appends a video media line (m=video) to the existing SDP body and sends an updated offer. User agents that want to add a media stream must always append media lines to existing ones. This ordering ensures that the peer user agent is able to gauge any new media line additions. For example, Example 2-9 shows a session established between a video-capable phone and a phone that does not support video. The original video-capable IP phone is then transferred to another video-capable phone, and a new video session is added to the existing media sessions.

Example 2-9 ***Adding a Media Stream***

```
m=audio 21040 RTP/AVP 114 9 124 113 115 0 8 116 18 101
b=TIAS:64000
a=rtpmap:114 opus/48000/2
a=fmtp:114  maxplaybackrate=16000;sprop-maxcapturerate=16000;
  stereo=0;sprop-stereo=0;usedtx=0
a=rtpmap:9 G722/8000
a=rtpmap:124 iSAC/16000
a=rtpmap:113 AMR-WB/16000
a=fmtp:113 mode-change-capability=2
a=rtpmap:115 AMR-WB/16000
a=fmtp:115 octet-align=1;mode-change-capability=2
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:116 iLBC/8000
a=maxptime:20
a=fmtp:116 mode=20
a=rtpmap:18 G729/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15
a=trafficclass:conversational.audio.avconf.aq:admitted
m=video 29584 RTP/AVP 100 126 97
b=TIAS:384000
a=rtpmap:100 H264/90000
a=fmtp:100 profile-level-id=640C16;packetization-mode=1;max-m
  max-fs=3600;max-rcmd-nalu-size=256000;level-asymmetry-allow
a=rtpmap:126 H264/90000
a=fmtp:126 profile-level-id=428016;packetization-mode=1;max-m
  max-fs=3600;max-rcmd-nalu-size=256000;level-asymmetry-allow
a=rtpmap:97 H264/90000
a=fmtp:97 profile-level-id=428016;packetization-mode=0;max-mb
  max-fs=3600;max-rcmd-nalu-size=256000;level-asymmetry-allow
a=imageattr:* recv [x=800,y=480,q=0.60] [x=1280,y=720,q=0.50]
a=content:main
a=rtcp-fb:* nack pli
a=rtcp-fb:* ccm fir
a=rtcp-fb:* ccm tmmbr
a=trafficclass:conversational.video.avconf.aq:admitted
```

[Video Escalation (SIP+SDP) - New Answer]

```
Content-Type: application/sdp
Content-Length: 830

v=0
o=CiscoSystemsCCM-SIP 280365 5 IN IP4 172.18.110.91
s=SIP Call
c=IN IP4 14.50.214.109
b=TIAS:384000
b=AS:384
t=0 0
m=audio 25106 RTP/AVP 114 101
b=TIAS:64000
a=rtpmap:114 opus/48000/2
a=fmtp:114  maxplaybackrate=16000;sprop-maxcapturerate=16000;
  stereo=0;sprop-stereo=0;usedtx=0
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15
a=trafficclass:conversational.audio.avconf.aq:admitted
m=video 28130 RTP/AVP 100
b=TIAS:320000
a=rtpmap:100 H264/90000
a=fmtp:100 profile-level-id=640C16;packetization-mode=1;max-m
  max-fs=3600;max-rcmd-nalu-size=256000;level-asymmetry-allow
a=imageattr:* recv [x=800,y=480,q=0.60] [x=1280,y=720,q=0.50]
a=content:main
a=rtcp-fb:* nack pli
a=rtcp-fb:* ccm fir
a=rtcp-fb:* ccm tmmbr
a=trafficclass:conversational.video.avconf.aq:admitted
```

It is also possible to add a media stream to a communication session by activating a previously disabled media stream. Consider a scenario in which a user agent attempts to establish a communication session that includes audio and video media types. If the peer user agent does not support video, it rejects the video media line by setting the port to zero (refer to Example 2-7). During the course of the communication session, either user agent may decide to reuse the video media line slot to introduce a new media stream. The new media stream can have a video media type or any other valid media type.

Removing a Media Stream

A user agent can remove an existing media stream by constructing a new SDP body and setting the media port of the corresponding media stream to zero. When such an SDP body is received by the peer user agent, it is treated as a non-negotiable and explicit indication to disable a given media stream. Therefore, the peer user agent must construct an answer with the port for the media stream in question also set to zero.

Media streams that are deleted by an updated SDP offer/answer exchange cease to exchange RTP or RTCP traffic. Any resources allocated for such media streams can be de-allocated.

NOTE

The concept of zeroing out a port may also be referred to as *disabling a media stream*.

Modifying the Address, Port, Transport, or Media Format

As mentioned earlier in this chapter, nearly every aspect of a communication session can be modified. The way in which media streams can be added and removed using the SDP offer/answer exchange is discussed in the previous section. During the course of a communication session, it may happen that a user agent discovers a new interface that is known to be more reliable than the current interface engaged in media transmission and reception. To ensure that the newly discovered, higher-priority interface takes over for media transmission and reception, the user agent has to construct an updated SDP offer in which the connection information field is modified to reflect the new interface identity. In most instances, a modification of the connection information field proceeds with a change in the port number(s) for a media line(s), and this is reflected in the update SDP offer.

NOTE

There are several other scenarios that require modification of the media IP address and port. These include, but are not limited to, media redirection from one endpoint to another, call hold, and call resume.

Even after updating the connection information and port, a user agent must be prepared to receive media on the old IP address/port pair for a reasonable amount of time. This is because the peer user agent has to accept the updated SDP offer, proceed to process the changes, and then program its internal software subsystems accordingly. You should also be aware that it is possible for an answerer to update its own IP address/port pair in the answer to an updated offer.

When setting up a communication session, the participants converge on a media format by using the SDP information that is exchanged. In addition, it is perfectly acceptable for a user agent to attempt to change the media format midsession. The way in which a user agent changes the media format midsession is achieved by first constructing an updated SDP offer such that the media line(s) contains a completely new set of media formats (not present in the previous SDP) or a set of media formats that partially overlap with the previous SDP body. The offer can be rejected or accepted by the answerer. When accepted, the media format used is determined by SDP in the answer.

Previous Section 4. Overview of H.323 | Next Section