Internet Engineering Task Force MMUSIC WG
Internet Draft J.Rosenberg,H.Schulzrinne
draft-rosenberg-mmusic-sdp-offer-answer-00.txt dynamicsoft,Columbia U.
October 26, 2001
Expires: April 2002

An Offer/Answer Model with SDP

STATUS OF THIS MEMO

Abstract

This document defines a mechanism by which two entities can make use of SDP arrive at a common view of a multimedia session between them. In the model, one participant offers the other a description of the desired session from their perspective, and the other participant answers with the desired session from their perspective. This offer/answer model is most useful in unicast sessions where information from both participants is needed for the complete view of the session. The offer/answer model is used by protocols like SIP.

1 Introduction

The Session Description Protocol (SDP) [1] was originally conceived as a way to describe multicast sessions carried on the Mbone. The Session Announcement Protocol (SAP) [2] was devised as a multicast mechanism to carry SDP messages. Although the SDP specification allows for unicast operation, it is not complete. Unlike multicast, where there is a global view of the session that is used by all participants, unicast sessions involve two participants, and a complete view of the session requires information from both participants, and agreement on parameters between them.

As an example, a multicast session requires conveying a single multicast address for a particular media stream. However, for a unicast session, two addresses are needed - one for each participant. As another example, a multicast session requires indication of which codecs will be used in the session. However, for unicast, the set of codecs needs to be determined by finding an overlap in the set supported by each participant.

As a result, even though SDP has the expressiveness to describe unicast sessions, it is missing the semantics and operational details of how it is actually done. In this draft, we remedy that by defining a simple offer/answer model based on SDP. In this model, one participant in the session generates an SDP that constitutes the offer - the set of media streams and codecs the offerer wishes to use, along with the IP addresses and ports the offer would like to use to receive the media. The offer is conveyed to the other participant, called the answerer. The answerer generates an answer, which is an SDP that responds to offer provided it. The answer has a matching media stream for each one in the offer, indicating whether the stream is accepted or not, along with the codecs that will be used and the IP addresses and ports that the answerer wants to use to receive media.

We also define guidelines for how the offer/answer model is used to update a session once it has begun.

The means by which the offers and answers are conveyed are outside the scope of this document. The offer/answer model defined here is the mandatory baseline mechanism used by the Session Initiation Protocol (SIP) [3].

2 Generating the initial offer

The offer (and answer) MUST be a valid SDP, as defined by RFC 2327 [1]. This means it MUST contain a v line, o line, s line and t line. Either an e line or p line MUST be present. However, it is RECOMMENDED that all implementations accept SDP without e, p, or s lines. The numeric value of the session id and version in the o line MUST be representable with a 64 bit signed integer. Although the SDP specification allows for multiple session descriptions to be concatenated together into a large SDP message, an SDP message used

in the offer/answer model MUST contain only a single session description.

The SDP "s=" line and the SIP Subject header field have different meanings when inviting to a multicast session. The session description line describes the subject of the multicast session, while the SIP Subject header field describes the reason for the invitation. For invitations to two-party sessions, the SDP "s=" line MAY consist of a single space character (0x20).

> Unfortunately, SDP does not allow to leave the "s=" line empty.

Once the offerer has sent the offer, it MUST be prepared to receive media described by that offer.

2.1 Unicast

The offer MUST contain zero or more media sections. Zero media sessions implies that the offerer wishes to communicate, but that the streams for the session will be added at a later time through re-INVITEs.

If a session description from an offerer contains a media stream which is listed as send (receive) only, it means that the offerer is only willing to send (receive) this stream, not receive (send). Media streams are marked as send-only or receive-only media streams using the SDP "a=sendonly" and "a=recvonly" attributes, respectively. If neither attribute is present, the default is both send and receive (which MAY be explicitly indicated with the "a=sendrecv" attribute).

For recvonly and sendrecv streams, the port number and address in the session description indicate where the media stream should be sent to. For send-only RTP streams, the address and port number indicate where RTCP reports are to be sent to. Specifically, RTCP reports are sent to the port number one higher than the number indicated. The IP address and port present in the offer indicate nothing about the source IP address and source port of RTP and RTCP packets that will be sent by the offerer. A port number of zero in the offer indicates that the stream is offered but should never be used. This has no useful semantics in an initial INVITE, but is allowed for reasons of completeness, since the response can contain a zero port indicating a rejected stream (Section 3. Furthermore, existing streams can be terminated by setting the port to zero (Section 4. In general, a port number of zero indicates that the media stream is not wanted.

The list of payload types for each media stream conveys two pieces of

information, namely the set of codecs that the offerer is capable of sending and/or receiving (depending on the direction attributes), and the RTP payload type numbers used to identify those codecs. If multiple codecs are listed, it means that the offerer is capable of making use of any of those codecs during the call. In other words, the answerer MAY change codecs in the middle of the call, without sending a SIP message, to make use of any of those listed. For a send-only stream, the offer SHOULD indicate those codecs the offerer is willing to send for this stream. For a receive-only stream, the offer SHOULD indicate those codecs the offerer is willing to receive for this stream. For a send and receive stream, the offer SHOULD indicate those codecs that the offerer is willing to send and receive with.

For receive-only streams, the payload type numbers indicate the value of the payload type field in RTP packets the offerer is expecting to receive for that codec. For send-only streams, the payload type numbers indicate the value of the payload type field in RTP packets the offerer is planning to send for that codec type. For send-and-receive streams, the payload type numbers indicate the value of the payload type field the offerer expects to both send and receive. This means that the payload type for a codec is the same in both directions.

All media descriptions SHOULD contain "a=rtpmap" mappings from RTP payload types to encodings. If there is no "a=rtpmap", the static payload type table from RFC 1890 [4] is to be used.

> This allows easier migration away from static payload types.

In all cases, the codecs in the m line are listed in order of preference, with the first codec listed being preferred. In this case, preferred means that the recipient of the offer SHOULD use the codec with the highest preference that is acceptable to it.

If the ptime attribute is present for a stream, it indicates the desired packetization interval that the offerer would like to receive.

If multiple media streams of different types are present, it means that the offerer wishes to use those streams at the same time. A typical case is an audio and video stream as part of a videoconference.

If multiple media streams of the same type are present, it means that the offerer wishes to send (and/or receive), multiple streams at the same time. When sending multiple streams of the same type, the source

of the stream (such as the microphone or circuit on a gateway) is sent multiple times, once for each stream. Each stream MAY use different encodings. When receiving multiple streams of the same type, the streams MUST be mixed before playing them out. A typical usage example is a pre-paid calling card application, where the user can enter in a "long pound" at any time during a call to hangup and make a new call on the same card. This requires media from the user to the remote gateway, and to a system that looks for the long pound.

There are some codecs, such as the RTP payload format for DTMF tones and digits [5] and comfort noise codecs, which can only encode specific types of media content. When one of these codecs is present in an offered stream that is send-only or send-and-receive, it means that the offerer will send using that codec only when the content of the media stream is of a type that can be encoded with that codec. When the content of the media stream cannot be encoded with that codec, another codec indicated in the m line can be used. If there are no other codecs in the m line, nothing is sent. This is useful for handling the case where a UA would like to send DTMF only, using RFC 2833, to a remote server. This is indicated with a single media line containing only the DTMF codec.

2.2 Multicast

Construction of a session description for a multicast offer follows the procedures above, with a few exceptions.

The address listed in the c line MUST be a multicast address. It indicates the address that the offerer wishes to receive packets on.

The interpretation of send-only and receive-only for multicast media sessions differs from that for unicast sessions. For multicast, send-only means that the recipient of the session description (caller or callee) SHOULD only send media streams to the address and port indicated. Receive-only means that the recipient of the session description SHOULD only receive media on the address and port indicated.

3 Generating the answer

The answer to an offered SDP is based on the offered SDP. If the answer is different in any way (different IP addresses, ports, etc.), the origin line MUST be different in the answer, since the answer is generated by a different entity. In that case, the version number in the o line of the answer is unrelated to the version number in the o line of the offer.

For each m line in the offer, there MUST be a corresponding m line in

the answer. The answer MUST contain exactly the same number of m lines as the offer. This allows for streams to be matched up based on their order. This implies that if the offer contained zero m lines, the answer MUST contain zero m lines.

An offered stream MAY be rejected in the answer, for any reason. The definition of rejected is both neither offerer and answerer MUST NOT generate media (or RTCP packets) for that stream. To reject an offered stream, the port number in the corresponding stream in the answer is set to zero. Any media formats listed are ignored. At least one MUST be present, as specified by SDP.

Once the answer has been sent, the answerer MUST be prepared to receive media as described in the answer.

3.1 Unicast

If a stream is offered with a unicast address, the answer MUST contain a unicast address.

If a stream is offered as sendonly, the corresponding stream MUST be marked as recvonly in the answer. Furthermore, the corresponding stream in the answer MUST contain at least one codec the answerer is willing to receive with from amongst those listed in the offer. The stream MAY indicate additional codecs, not listed in the corresponding stream in the offer, that the answerer is willing to receive with. The connection address and port indicate the address where the answerer wishes to receive RTP (RTCP will be received on the port which is one higher).

If a media stream is listed as recvonly in the offer, the answer MUST be marked as sendonly. Furthermore, the corresponding stream in the answer MUST contain at least one codec the answerer is willing to send with from amongst those listed in the offer. The IP address and port indicate the address where the answerer wishes to receive RTCP (RTCP will be received on the port number one higher than the one listed in the SDP).

If an offered media stream is listed as sendrecv (or contains no direction attribute, in which case it is sendrecv by default), the corresponding stream in the answer MAY be marked as sendonly, recvonly, or sendrecv. The default is sendrecv. If the stream in the answer is marked as sendonly, it MUST contain at least one codec the answerer is willing to send with from amongst those listed in the offer. The IP address and port indicate the address where the answerer wishes to receive RTCP. If the stream in the answer is marked as recvonly, it MUST contain at least one codec the answerer is willing to receive with from amongst those listed in the offer.

The stream MAY indicate additional codecs, not listed in the corresponding stream in the offer, that the answerer is willing to receive with. The connection address and port in the answer indicate the address where the answerer wishes to receive RTP and RTCP (RTCP will be received on the port number one higher than the one listed in the SDP). If the stream in the answer is marked as sendrecv, it MUST contain at least one codec the answerer is willing to both send and receive with, from amongst those listed in the offer. The stream MAY indicate additional codecs, not listed in the corresponding stream in the offer, that the answerer is willing to receive with. The connection address and port indicate the address where the answerer wishes to receive RTP (RTCP will be received on the port which is one higher).

The payload type numbers for a particular codec within a stream MUST be the same in offer and answer. In other words, a different dynamic payload type number for the same codec cannot be used in each direction.

In all cases, the codecs in the m line are listed in order of preference, with the first codec listed being preferred. In this case, preferred means that the recipient of the answer SHOULD use the codec with the highest preference that is acceptable to it.

The answerer MAY include a ptime attribute for any media stream; this indicates the packetization interval that the answerer would like to receive. There is no requirement that the packetization interval be the same in each direction for a particular stream.

Although the answerer MAY list the codecs in their desired order of preference, it is RECOMMENDED that unless there is a specific reason, the answer list codecs in the same relative order they were present in the offer. In other words, if a stream in the offer lists codecs 8, 22 and 48, in that order, and the answerer only supports codecs 8 and 48, it is RECOMMENDED that, if the answerer has no reason to change it, the ordering of codecs in the answer be 8, 48, and not 48, 8. This helps ensure that the same codec is used in both directions.

If the answerer has no media formats in common for a particular offered stream, the answerer MUST reject that media stream.

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

3.2 Multicast

For multicast, receive and send multicast addresses are the same and all parties use the same port numbers to receive media data. If the

session description provided by the offerer is acceptable to the answerer, the answerer can choose not to include a session description or MAY echo the description in the response.

An answerer MAY return a session description with some of the payload types removed, or port numbers set to zero (but no other value). This indicates to the offerer that the answerer does not support the given stream or media types which were removed. An answerer MUST NOT change whether a given stream is send-only, receive-only, or send-and-receive.

If an answerer does not support multicast at all, it SHOULD reject the session description.

4 Modifying the session

At any point during the call, either participant MAY issue a re-INVITE to modify characteristics of the session. It is fundamental to the operation of SIP that the exact same offer-answer procedure defined above is used for re-INVITE. This means that a re-INVITE MAY contain no SDP, so that the 200 OK to the re-INVITE contains the offer. In this case, the offerer MUST offer the same SDP it provided previously. This is to ensure that the offered SDP in the 2xx will be acceptable to the UAC, as there is no way to reject it.

The offer in a re-INVITE MAY be identical to the last SDP provided to the other party (which may have been provided in an offer or an answer), or it MAY be different. We refer to the last SDP provided as the "previous SDP". If the offer is the same, the answer MAY be the same as the previous SDP from the answerer, or it MAY be different. If the offered SDP is different from the previous SDP, some constraints are placed on its construction, discussed below.

Nearly all aspects of the session can be modified. New streams can be added, existing streams can be deleted, and parameters of existing streams can change. When issuing an offer that modifies the session, the o line of the new SDP MUST be identical to that in the previous SDP, except that the version in the origin field MUST increment from the previous SDP. If the version in the origin line does not increment, the SDP MUST be identical to the SDP with that version number. The answerer MUST be prepared to receive an offer that contains SDP with a version that has not changed; this is effectively a no-op. However, the answerer MUST generate a valid answer (which MAY be the same as the previous SDP from the answerer, or MAY be different), according to the procedures defined in Section 3.

If an SDP is offered which is different from the previous SDP, the new SDP MUST have a matching media section for each media section in

the previous SDP. In other words, if the previous SDP had N media lines, the new SDP MUST have at least N media lines. The ith media stream in the previous SDP, counting from the top, matches the ith media stream in the new SDP, counting from the top. This matching is necessary in order for the answerer to determine which stream in the new SDP corresponds to a stream in the previous SDP. Because of these requirements, the number of m lines in a stream never decreases, but only increases. Deleted media streams from a previous SDP MUST NOT be removed from a new SDP.

4.1 Adding a media stream

New media streams are created by adding additional media descriptions below the existing ones. New media sections MUST appear below any existing media sections. The rules for formatting this media section are identical to those described in Section 2.

When the answerer receives an SDP with more media descriptions than the previous SDP from the offerer, the answerer knows that new media streams are being added. These can be rejected or accepted by placing a matching media description in the answer. The procedures for constructing the new media description in the answer are described in Section 3.

4.2 Removing a media stream

Existing media streams are removed by creating a new SDP with the port number for that stream set to zero. Otherwise, the media description SHOULD be formatted identically to the corresponding stream in the previous SDP.

A stream that is offered with a port of zero MUST be marked with port zero in the answer. Otherwise, the media description for the removed stream SHOULD be formatted identically to the corresponding stream in the previous SDP.

4.3 Modifying a media stream

Nearly all characteristics of a media stream can be modified.

The port number for a stream MAY be changed. To do this, the offerer creates a new media description, with the port number in the m line different from the corresponding stream in the previous SDP. If only the port number is to be changed, the rest of the media stream description SHOULD remain unchanged. The offerer MUST be prepared to receive media on both the old and new ports as soon as the offer is sent. The offerer MUST NOT cease listening for media on the old port until media arrives on the new port. At that time, it MAY cease

listening for media on the old port.

The corresponding media stream in the answer MAY be the same as the stream in the previous SDP from the answerer, or MAY be different. If the updated stream is accepted by the answerer, the answerer SHOULD begin sending traffic for that stream to the new port immediately. If the answerer changes the port from the previous SDP, it MUST be prepared to receive media on both the old and new ports as soon as the answer is sent. The answerer MUST NOT cease listening for media on the old port until media arrives on the new port. At that time, it MAY cease listening for media on the old port.

To change the IP address where media is sent to, the same procedure is followed for changing the port number. The only difference is that the connection line is updated, not the port number.

The list of codecs used in the session MAY be changed. To do this, the offerer creates a new media description, with the list of media formats in the m line different from the corresponding stream in the previous SDP. This list MAY include new codecs, and MAY remove codecs present from the previous SDP. When a new codec is used with a dynamic payload type number, it MUST NOT reuse a dynamic payload type number used previously in the session. The stream of RTP packets is only loosely synchronized with the SIP/SDP that controls them. So, if a user agent A sends media with dynamic payload type 10, and then performs a re-INVITE that removes codec 10, and then later another re-INVITE that adds a new codec, reusing dynamic payload type 10, an old packet might arrive at the recipient. It would have no way to know whether payload type 10 indicates the old codec or the new. Rather than attempting to synchronize by passing RTP sequence numbers in SDP (which requires additional state), we simply disallow reuse.

The corresponding media stream in the answer is formulated as described in Section 3. If the new list of codecs for a stream changes the choice of which codec is used, the new codec SHOULD be used immediately. That means the offerer MUST be prepared to receive media with a new codec as soon as it sends the offer, and the answerer MUST be prepared to receive media with a new codec as soon as it sends the answer.

The media type (audio, video, etc.) for a stream MAY be changed. This is particularly useful for changing between voice and fax in a single stream, which are both separate media types. To do this, the offerer creates a new media description, with a new media type, in place of the description in the previous SDP which is to be changed. The IP address and port for the stream MAY change, or MAY remain the same. However, the list of payload type numbers for the new codecs MUST be different than any used previously for this stream.

The corresponding media stream in the answer is formulated as described in Section 3. Assuming the stream is acceptable, the answerer SHOULD begin sending with the new media type and codecs as soon as it receives the offer.

The transport for a stream MAY be changed. The process for doing this is identical to changing the port, excepting the transport is updated, not the port.

Any other attributes in a media description MAY be updated in an offer.

4.4 Putting a media stream on hold

If a party in a call wants to put the other party "on hold", i.e., request that it temporarily stops sending one or more media streams, a party offers the other an updated SDP.

If the stream to be placed on hold was previously a sendrecv media stream, it is placed on hold by marking it as sendonly. If the stream to be placed on hold was previously a recvonly media stream, it is placed on hold by marking it inactive. The inactive direction attribute is specified in RFC 3108 [6].

This means that a stream is placed "on hold" separately in each direction. Each stream is placed "on hold" independently. The recipient of an offer for a stream on-hold SHOULD NOT automatically return an answer with the corresponding stream on hold. An SDP with all streams "on hold" is referred to as held SDP

> Certain third party call control scenarios do not work when a UA responds to held SDP with held SDP.

Typically, when a user "presses" hold, the UA will generate a re-INVITE with all streams in the SDP indicating a direction of sendonly, and it will also locally mute, so that no media is sent to the far end, and no media is played out.

RFC2543 specified that placing a user on hold was accomplished by setting the connection address to 0.0.0.0. This has been deprecated, since it doesn't allow for RTCP to be used with held streams, and breaks with connection oriented media. However, a UA MUST be capable of receiving SDP with a connection address of 0.0.0.0, in which case it means that neither RTP nor RTCP should be sent to the peer.

5 Example

For example, assume that the caller Alice has included the following description in her INVITE request. It includes a bidirectional audio stream and two bidirectional video streams, using H.261 (payload type 31) and MPEG (payload type 32). The offered SDP is:

```
v=0
o=alice 2890844526 2890844526 IN IP4 host.anywhere.com
s=New board design
e=alice@foo.org
t=0 0
c=IN IP4 host.anywhere.com
m=audio 49170 RTP/AVP 0
a=rtpmap:0 PCMU/8000
m=video 51372 RTP/AVP 31
a=rtpmap:31 H261/90000
m=video 53000 RTP/AVP 32
a=rtpmap:32 MPV/90000
```

The callee, Bob, does not want to receive or send the first video stream, so it returns the media description below as the answer:

```
v=0
o=bob 2890844730 2890844730 IN IP4 host.example.com
s=New board design
e=bob@bar.com
t=0 0
c=IN IP4 host.example.com
m=audio 47920 RTP/AVP 0 1
a=rtpmap:0 PCMU/8000
m=video 0 RTP/AVP 31
m=video 53000 RTP/AVP 32
a=rtpmap:32 MPV/90000
```

At some point later, Bob decides to change the port where he will receive the audio stream (from 47920 to 6400), and at the same time, add an additional audio stream as receive only, using the RTP payload format for events. Bob offers the following SDP in the INVITE:

```
v=0
```

```
o=bob 2890844730 2890844731 IN IP4 host.example.com
s=New board design
e=bob@bar.com
t=0 0
c=IN IP4 host.example.com
m=audio 6400 RTP/AVP 0 1
a=rtpmap:0 PCMU/8000
m=video 0 RTP/AVP 31
m=video 53000 RTP/AVP 32
a=rtpmap:32 MPV/90000
m=audio 8864 RTP/AVP 110
a=rtpmap:110 telephone-events
a=recvonly
```

Alice accepts the additional media stream, and so generates the following answer:

```
v=0
o=alice 2890844526 2890844527 IN IP4 host.anywhere.com
s=New board design
e=alice@foo.org
t=0 0
c=IN IP4 host.anywhere.com
m=audio 49170 RTP/AVP 0
a=rtpmap:0 PCMU/8000
m=video 51372 RTP/AVP 31
a=rtpmap:31 H261/90000
m=video 53000 RTP/AVP 32
a=rtpmap:32 MPV/90000
m=audio 4520 RTP/AVP 110
a=rtpmap:110 telephone-events
a=sendonly
```

6 Author's Addresses

Jonathan Rosenberg
dynamicsoft
72 Eagle Rock Avenue
First Floor
East Hanover, NJ 07936

email: jdrosen@dynamicsoft.com

Henning Schulzrinne
Dept. of Computer Science
Columbia University
1214 Amsterdam Avenue
New York, NY 10027
USA
email: schulzrinne@cs.columbia.edu

7 Bibliography

[1] M. Handley and V. Jacobson, "SDP: session description protocol," Request for Comments 2327, Internet Engineering Task Force, Apr. 1998.

[2] M. Handley, C. Perkins, and E. Whelan, "Session announcement protocol," Request for Comments 2974, Internet Engineering Task Force, Oct. 2000.

[3] M. Handley, H. Schulzrinne, E. Schooler, and J. Rosenberg, "SIP: Session initiation protocol," Internet Draft, Internet Engineering Task Force, Nov. 2000. Work in progress.

[4] H. Schulzrinne, "RTP profile for audio and video conferences with minimal control," Request for Comments 1890, Internet Engineering Task Force, Jan. 1996.

[5] H. Schulzrinne and S. Petrack, "RTP payload for DTMF digits, telephony tones and telephony signals," Request for Comments 2833, Internet Engineering Task Force, May 2000.

[6] R. Kumar and M. Mostafa, "Conventions for the use of the session description protocol (SDP) for ATM bearer connections," Request for Comments 3108, Internet Engineering Task Force, May 2001.

Table of Contents