SIP (Session Initiation Protocol) Usage of the Offer/Answer Model
draft-ietf-sipping-sip-offeranswer-12

Abstract

The Session Initiation Protocol (SIP) utilizes the offer/answer model to establish and update multimedia sessions using the Session Description Protocol (SDP). The description of the offer/answer model in SIP is dispersed across multiple RFCs. This document summarizes all the current usages of the offer/answer model in SIP communication.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on September 9, 2010.

Copyright Notice

Copyright (c) 2010 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal
Table of Contents

1. Introduction .................................................. 3
   1.1. Terminology ............................................. 3
2. Summary of SIP usage of the Offer/Answer Model ................ 3
   2.1. Offer/Answer Exchange Pairs in SIP Messages .......... 4
   2.2. Rejection of an Offer ................................. 5
   2.3. Session Description which is not Offer nor Answer . 6
3. Detailed Discussion of the Offer/Answer Model for SIP ....... 7
   3.1. Offer/Answer for the INVITE method with 100rel extension .......................... 7
   3.1.1. INVITE Request with SDP ......................... 8
   3.1.2. INVITE request without SDP ....................... 9
   3.2. Offer/Answer Exchange in Early Dialog .............. 10
   3.3. Offer/Answer Exchange in an Established Dialog .... 11
   3.4. Recovering From a Failed ReINVITE ................... 11
4. Exceptional Case Handling ...................................... 12
   4.1. Message Crossing Case Handling ....................... 12
   4.2. Glare Case Handling ................................. 14
5. Content of Offers and Answers .................................. 15
   5.1. General Principle for Constructing Offers and Answers .... 16
   5.2. Choice of Media Types and Formats to Include and Exclude ................. 16
   5.2.1. Sending an Initial INVITE with Offer ............... 16
   5.2.2. Responding with an Offer when the Initial INVITE has no Offer ............. 17
   5.2.3. Answering an Initial INVITE with Offer ............ 17
   5.2.4. Answering when the Initial INVITE had no Offer .... 18
   5.2.5. Subsequent Offers and Answers .................... 18
   5.3. Hold and Resume of media ................................ 19
   5.4. Behavior on receiving SDP with c=0.0.0.0 ............... 20
6. IANA Considerations .............................................. 20
7. Security Considerations ........................................ 21
8. Acknowledgement ................................................ 21
9. References ...................................................... 21
   9.1. Normative References ................................... 21
   9.2. Informative References ................................ 22
Authors’ Addresses ................................................. 22
1. Introduction

SIP utilizes the offer/answer model to establish and update sessions. The rules to govern the offer/answer behaviors in SIP are described in the several RFCs. ([RFC3261], [RFC3262], [RFC3264], [RFC3311], and [I-D.camarillo-sipcore-reinvite].)

The primary purpose of this document is to describe all forms of SIP usage of the offer/answer model in one document to help the readers to fully understand it. Also, this document tries to incorporate the results of the discussions on the controversial issues to avoid repeating the same discussions later.

This document does not make normative changes. Rather, it recommends how to use the existing standards to best effect.

1.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 ([RFC2119]). This document only uses these key words when referencing normative statements in existing RFCs.

2. Summary of SIP usage of the Offer/Answer Model

The offer/answer model itself is independent from the higher layer application protocols which utilize it. SIP is one of the applications using the offer/answer model. [RFC3264] defines the offer/answer model, but does not specify which SIP messages should convey an offer or an answer. This should be defined in the SIP core and extensions RFCs.

In theory, any SIP message can include a session description in its body. But a session description in a SIP message is not necessarily an offer or an answer. Only certain session description usages that conform to the rules described in standards-track RFCs can be interpreted as an offer or an answer. The rules for how to handle the offer/answer model are defined in several RFCs.

The offer/answer model defines a mechanism for update of sessions. In SIP, a dialog is used to associate an offer/answer exchange with the session which it is to update. In other words, only the offer/answer exchange in the SIP dialog can update the session which is managed by that dialog.
2.1. Offer/Answer Exchange Pairs in SIP Messages

Currently, the rules on the offer/answer model are defined in [RFC3261], [RFC3262], [RFC3264], [RFC3311] and [I-D.camarillo-sipcore-reinvite]. In these RFCs, only the six patterns shown in Table 1 are defined for exchanging an offer and an answer with SIP messages.

Note that an offer/answer exchange initiated by an INVITE request must follow exactly one of the patterns 1, 2, 3, 4. When an initial INVITE causes multiple dialogs due to forking, an offer/answer exchange is carried out independently in each distinct dialog. When an INVITE request contains no offer, only pattern 2 or pattern 4 apply. ‘The first reliable non-failure message’ must have an offer if there is no offer in the INVITE request. This means that UA which receives the INVITE request without an offer must include an offer in the first reliable response with 100rel extension. If no reliable provisional response has been sent, the UAS must include an offer when sending 2xx response.

In pattern 3, the first reliable provisional response may or may not have an answer. When a reliable provisional response contains a session description, and is the first to do so, then that session description is the answer to the offer in the INVITE request. The answer can not be updated, and a new offer can not be sent in a subsequent reliable response for the same INVITE transaction.

In pattern 5, a PRACK request can contain an offer only if the reliable response which it acknowledges contains an answer to the previous offer/answer exchange.

NOTE: It is legal to have UPDATE/2xx exchanges without offer/answer exchanges (pattern 6). However when re-INVITEs are sent for non-offer/answer purposes, an offer/answer exchange is required. In that case the prior SDP will typically be repeated.

There may be ONLY ONE offer/answer negotiation in progress for a single dialog at any point in time. Section 4 explains how to ensure this. When an INVITE results in multiple dialogs each has a separate offer/answer negotiation.

NOTE: This is when using a Content-Disposition of "session". There may be a second offer/answer negotiation in progress using a Content-Disposition of "early-session" [RFC3959]. That is not addressed by this draft.
In Table 1, ‘1xx-rel’ corresponds to the reliable provisional response which contains the 100rel option defined in [RFC3262].

The ‘Ini’ column shows the ability to exchange the offer/answer to initiate the session. ‘Y’ indicates that the pattern can be used in the initial offer/answer exchange, while ‘N’ indicates that it can not. Only the initial INVITE transaction can be used to exchange the offer/answer to establish a multimedia session.

The ‘Est’ column shows the ability to update the established session.

The ‘Early’ column indicates which patterns may be used to modify the established session in an early dialog. There are two ways to exchange a subsequent offer/answer in an early dialog.

2.2. Rejection of an Offer

It is not always clear how to reject an offer when it is unacceptable, and some methods do not allow explicit rejection of an offer. For each of the patterns in Table 1, Table 2 shows how to reject an offer.

When a UA receives an INVITE request with an unacceptable offer, it should respond with a 488 response, preferably with Warning header field indicating the reason of the rejection, unless another response code is more appropriate to reject it. (Pattern 1 and Pattern 3.)

If this is a reINVITE extra care must be taken, as detailed in [I-D.camarillo-sipcore-reinvite]. Specifically, if the offer contains any changes or additions to media stream properties, and those have already been used to transmit/receive media before the final response is sent, then a 2xx response should be sent, with a syntactically correct response. This may optionally be followed by an UPDATE request to rearrange the session parameters if both ends support the UPDATE method. Alternatively the UA may terminate the dialog and send an error response to the INVITE request.
When a UA receives an UPDATE request with an offer which it can not accept, it should respond with a 488 response preferably with Warning header field indicating the reason of the rejection, unless another response code is more appropriate to reject it. (Pattern 6)

When a UA receives a PRACK request with an offer which it can not accept, it may respond with a 200 response with a syntactically correct session description. This may optionally be followed by an UPDATE request to rearrange the session parameters if both ends support the UPDATE method. Alternatively the UA may terminate the dialog and send an error response to the INVITE request. (Pattern 5) (While it may be tempting to respond with a 488 response in this case, that is not recommended, because it does not acknowledge the response.)

When a UA receives a response with an offer which it can not accept, the UA does not have a way to reject it explicitly. Therefore, a UA should respond to the offer with the correct session description and rearrange the session parameters by initiating a new offer/answer exchange, or alternatively terminate the session. (Pattern 2 and Pattern 4) When initiating a new offer/answer, a UA should take care not to cause an infinite offer/answer loop.

<table>
<thead>
<tr>
<th>Offer</th>
<th>Rejection</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. INVITE Req. (*)</td>
<td>488 INVITE Response</td>
</tr>
<tr>
<td>2. 2xx INVITE Resp.</td>
<td>Answer in ACK Req. followed by new offer&lt;br&gt;OR termination of dialog</td>
</tr>
<tr>
<td>3. INVITE Req.</td>
<td>488 INVITE Response (same as Pattern 1.)</td>
</tr>
<tr>
<td>4. 1xx-rel INVITE Resp.</td>
<td>Answer in PRACK Req. followed by new offer&lt;br&gt;OR termination of dialog</td>
</tr>
<tr>
<td>5. PRACK Req. (**)</td>
<td>200 PRACK Resp. followed by new offer&lt;br&gt;OR termination of dialog</td>
</tr>
<tr>
<td>6. UPDATE Req.</td>
<td>488 UPDATE Response</td>
</tr>
</tbody>
</table>

(*) If this was a reINVITE, a failure response should not be sent if media has already been exchanged using the new offer.

(**) A UA should only use PRACK to send an offer when it has strong reasons to expect the receiver will accept the offer.

Table 2. Rejection of an Offer

2.3. Session Description which is not Offer nor Answer

As previously stated, a session description in a SIP message is not necessarily an offer or an answer. For example, SIP can use a session description to describe capabilities apart from offer/answer exchange. Examples of this are 200 OK responses for OPTIONS and 488
responses for INVITE.

3. Detailed Discussion of the Offer/Answer Model for SIP

3.1. Offer/Answer method for the INVITE method with 100rel extension

The INVITE method provides the basic procedure for offer/answer exchange in SIP. Without the 100rel option, the rules are simple as described in [RFC3261]. If an INVITE request includes a session description, pattern 1 is applied and if an INVITE request does not include a session description, pattern 2 is applied.

With 100rel, pattern 3 and pattern 4 are added and this complicates the rules. An INVITE request may cause multiple responses. Note that even if both UAs support the 100rel extension, not all the provisional responses may be sent reliably. Note also that a reliable provisional response is allowed without a session description if the UAS does not wish to send the answer yet. An unreliable provisional response may include a session description in the body if the UAS has not sent a reliable response, but its session description is neither an offer nor an answer. All the session descriptions in the unreliable responses to the INVITE request must be identical to the answer which is included in the reliable response. A session description in an unreliable response that precedes a reliable response can be considered a "preview" of the answer that will be coming, and hence may be treated like an answer until the actual one arrives.

NOTE: This "preview" session description rule applies to a single offer/answer exchange. In parallel offer/answer exchanges (caused by forking) a UA may obviously receive a different "preview" of an answer in each dialog. UAs are expected to deal with this.

Although RFC 3261 says a UA should accept media once an INVITE with an offer has been sent, in many cases, an answer (or, at least a preview of it) is required in order for media to be accepted. Two examples of why this might be required are:

- To avoid receiving media from undesired sources, some User Agents assume symmetric RTP will be used, ignore all incoming media packets until an address/port has been received from the other end, and then use that address/port to filter incoming media packets.

- In some networks, an intermediate node must authorize a media stream before it can flow and requires a confirming answer to the offer before doing so.
Therefore, a UAS should send an SDP answer reliably (if possible) before it starts sending media. And, if neither the UAC nor the UAS support 100rel, the UAS should send a preview of the answer before it starts sending media.

### 3.1.1. INVITE Request with SDP

When a UAC includes an SDP body in the INVITE request as an offer, it expects the answer to be received with one of the reliable responses. Other than that, no offer/answer exchanges can occur in the messages within the INVITE transaction.

#### UAC | UAS
---|---
| F1 INVITE (SDP) | <- The offer in the offer/answer model
| F2 1xx (SDP) | <- The offer/answer exchange is not closed yet, but UAC acts as if it receives the answer.
| F3 1xx-rel (no SDP) | <- a 1xx-rel may be sent without answer SDP.
| F4 PRACK (no SDP) | UAC must not send a new offer.
| F5 2xx PRA (no SDP) | v
| F6 1xx-rel (SDP) | <- The answer in the offer/answer model
| F7 PRACK | UAC can send a new offer in a PRACK request to acknowledge F6.
| F8 2xx PRA | After F7 UAC and UAS can send a new v offer in an UPDATE request.
| F9 1xx-rel | <- SDP should not be included in the subsequent 1xx-rel once offer/answer has been completed.
| F10 PRACK | |
| F11 2xx PRA | |
| F12 2xx INV | <- SDP should not be included in the final response once offer/answer has been completed.
| F13 ACK | |

Figure 1 Example of Offer/Answer with 100rel Extension (1)

For example, in Figure 1, only the SDP in F6 is the answer. The SDP in the non-reliable response (F2) is the preview of the answer and
must be the same as the answer in F6. Receiving F2, the UAC should act as if it receives the answer. However, offer/answer exchange is not completed yet and the UAC must not send a new offer until it receives the same SDP in the first reliable response, which is the real answer. After sending the SDP in F6, the UAS must prepare to receive a new offer from the UAC with an UPDATE request or a PRACK request.

The UAS does not include SDP in responses F9 and F12. However, the UAC should prepare to receive SDP bodies in F9 and/or F12, and just ignore them, to handle a peer that does not conform to the recommended implementation.

3.1.2. INVITE request without SDP

When a UAC does not include an SDP body in the INVITE request, it expects the offer to be received with the first reliable response. The UAC will send the answer in the request to acknowledge the response, i.e. PRACK or ACK request of the reliable response. Other than that, no offer/answer exchanges can occur in the messages within the INVITE transaction.

NOTE: The UAS should not include SDP in the responses F6 and F9. However, the UAC should prepare to receive SDP bodies in F6 and/or F9, and just ignore them to handle a peer that does not conform to the recommended implementation.
UAC | UAS
---|---
F1  INVITE (no SDP) |  
| F2  1xx
| F3  1xx-rel (SDP) <- The first 1xx-rel must contain SDP as the offer.
| F4  PRACK (SDP) <- A PRACK request to the first 1xx-rel must contain SDP as the answer.
| F5  2xx PRA (no SDP) -
| F6  1xx-rel (no SDP) <- The subsequent 1xx-rel should not contain SDP.
| F7  PRACK | UAC can send a new offer in an UPDATE request after F4.
| F8  2xx PRA
| F9  2xx INV (no SDP) <- The final response should not contain SDP.
| F10  ACK

Figure 2 Example of Offer/Answer with 100rel Extension (2)

Note that in the case that the UAC needs to prompt the user to accept or reject the offer, the reliable provisional response with SDP as an offer (pattern 4) can result in retransmission until the PRACK request can be sent. The UAC should take care to avoid this situation when it sends the INVITE request without SDP.

### 3.2. Offer/Answer Exchange in Early Dialog

When both UAs support the 100rel extension, they can update the session in the early dialog once the first offer/answer exchange has been completed.

From a UA sending an INVITE request:

A UA can send an UPDATE request with a new offer if both ends support the UPDATE method. Note that if the UAS needs to prompt the user to accept or reject the offer, the delay can result in retransmission of the UPDATE request.

A UA can send a PRACK request with a new offer only when acknowledging the reliable provisional response carrying the answer.
to an offer in the INVITE request. Compared to using the UPDATE method, using PRACK can reduce the number of messages exchanged between the UAs. However, to avoid problems or delays caused by PRACK offer rejection, the UA is recommended to send a PRACK request only when it has strong reasons to expect the receiver will accept it. For example, the procedure used in precondition extension [RFC3312] is a case where a PRACK request should be used for updating the session status in an early dialog. Note also that if a UAS needs to prompt the user to accept or reject the offer, the delay can result in retransmission of the PRACK request.

From a UA receiving an INVITE request:

A UA can send an UPDATE request with a new offer if both ends support the UPDATE method. A UAS can not send a new offer in the reliable provisional response, so the UPDATE method is the only method for a UAS to update an early session.

### 3.3. Offer/Answer Exchange in an Established Dialog

Both the re-INVITE and UPDATE methods can be used in an established dialog to update the session.

The UPDATE method is simpler and can save at least one message compared with the INVITE method. But both ends must support the UPDATE method for it to be used.

The INVITE method needs at least three messages to complete but no extensions are needed. Additionally, the INVITE method allows the peer to take time to decide whether it will accept a session update or not by sending provisional responses. That is, re-INVITE allows the UAS to interact with the user at the peer, while UPDATE needs to be answered automatically by the UAS. It is noted that re-INVITE should be answered immediately unless such a user interaction is needed. Otherwise, some 3pcc flows will break.

### 3.4. Recovering From a Failed ReINVITE

If a reINVITE fails, the session parameters in effect prior to the reINVITE MUST remain unchanged, as if no re-INVITE had been issued. ([RFC3261] section 14.1.) This remains the case even if multiple offer/answer exchanges have occurred between the sending of the reINVITE and its failure, and even if media has been exchanged using the proposed changes in the session. Because this can be difficult to achieve in practice, newer specifications call for the UAS to send a 2xx response to a reINVITE in cases where rolling back changes would be problematic.
Nevertheless, a UAC may receive a failure response to a reINVITE after proposed changes that must be rolled back have already been used. In such a case, the UAC should send an UPDATE offering the SDP that has been reinstated. (See [I-D.camarillo-sipcore-reinvite] for details.)

4. Exceptional Case Handling

In [RFC3264], the following restrictions are defined with regard to sending a new offer.

"At any time, either agent MAY generate a new offer that updates the session. However, it MUST NOT generate a new offer if it has received an offer which it has not yet answered or rejected. It MUST NOT generate a new offer if it has generated a prior offer for which it has not yet received an answer or a rejection."

Assuming that the above rules are guaranteed, there seem to be two possible ‘exceptional’ cases to be considered in SIP offer/answer usage: the ‘message crossing’ case, and the ‘glare’ case. One of the reasons why the usage of SIP methods to exchange offer/answer needs to be carefully restricted in the RFCs is to ensure that the UA can detect and handle appropriately the ‘exceptional’ cases to avoid incompatible behavior.

4.1. Message Crossing Case Handling

When message packets cross in the transport network, an offer may be received before the answer for the previous offer/answer exchange, as shown in Figure 3. In such a case, UA A must detect that the session description SDP-2 is not the answer to offer1.

```
   A                  B
SDP-1 (offer1)  SDP-2 (answer1)
M1 ----------------->
SDP-2 (answer1)  SDP-3 (offer2)
M2 <------\  /------>
     \/
SDP-3 <-------/  \------>
M3

Figure 3 Message Crossing Case
```

Because of the restrictions on placement of offers and answers (summarized in Table 1) there are a limited number of valid exchanges of messages that may lead to this message crossing case. These are
enumerated in Table 3. (This table only shows messages containing offers or answers. There could be other messages, without session descriptions, which are not shown.)

There are variants, shown in Figures 4a and 4b, which are dependent on an INVITE (Mx) that contains no offer. These are also included in Table 3.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDP-1</td>
<td>offer1 (UPD)</td>
</tr>
<tr>
<td>M1</td>
<td>===============&gt;</td>
</tr>
<tr>
<td>re-INV</td>
<td>(no offer)</td>
</tr>
</tbody>
</table>
| Mx                     | --------------------->|--+
| SDP-2                  | answer1 (2xx-UPD)     |
| M2                     | ===============>      |
| SDP-3                  | offer2               |
| M3                     | (1xx-rel/2xx)         |
| SDP-4                  | answer2 (PRACK/ACK)   |
| My                     | ===============>      |

Figure 4a Avoidable message crossing cases

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>re-INV</td>
<td>(no offer)</td>
</tr>
<tr>
<td>Mx</td>
<td>===============&gt;</td>
</tr>
<tr>
<td>SDP-1</td>
<td>offer1 (UPD)</td>
</tr>
<tr>
<td>M1</td>
<td>===============&gt;</td>
</tr>
<tr>
<td>SDP-2</td>
<td>answer1 (2xx-UPD)</td>
</tr>
<tr>
<td>M2</td>
<td>===============&gt;</td>
</tr>
<tr>
<td>SDP-3</td>
<td>offer2</td>
</tr>
<tr>
<td>M3</td>
<td>(1xx-rel/2xx)</td>
</tr>
<tr>
<td>SDP-4</td>
<td>answer2 (PRACK/ACK)</td>
</tr>
<tr>
<td>My</td>
<td>===============&gt;</td>
</tr>
</tbody>
</table>

Figure 4b Avoidable message crossing cases
Internet-Draft     SIP Usage of the Offer/Answer Model        March 2010

Table 3. Offer / Answer Crossing Message Sequences

(1)  This is indistinguishable from true glare. UA A should respond
to M2 with a 491 response.

(2)  This can only occur in situations depicted in figures 4a and 4b. It is easier for UA A to avoid these situations than to recover from them. The situation in Figure 4a can be avoided by refraining from sending a re-INVITE without offer when an unanswered offer is outstanding. The situation in Figure 4b can be avoided by refraining from sending any message containing an offer while an INVITE without offer is outstanding.

(3)  There are no valid sequences that result in these cases.

Summarizing, a UA that has an outstanding unanswered offer should:
o  refrain from sending a re-INVITE without an offer;
o  reject (491) an INVITE or UPDATE containing an offer.

4.2. Glare Case Handling

When both ends in a dialog send a new offer at nearly the same time, as described in Figure 5, a UA may receive a new offer before it receives the answer to the offer it sent. This case is usually called a ‘glare’ case.

Kyzivat & Sawada        Expires September 9, 2010              [Page 14]
When offer2 is in a UPDATE request or (re-)INVITE request, it must be rejected with a 491 response.

When offer2 is in a PRACK request (within the current rules, only possible if offer1 is in an UPDATE request), UA A has a dilemma: all PRACKs are supposed to be accepted with 200 response, yet there is no way to indicate the problem with a 200 response. At best it could proceed on the assumption that its INVITE will be rejected with a 491. To avoid this glare condition, UA A should not send an offer if it has already sent a reliable provisional response containing an answer to a previous offer and has not received the corresponding PRACK request.

Glare can also occur when offer2 is in a 1xx or 2xx response. To avoid this situation, when UA A has sent a (re)INVITE request without session description, it should not send an offer until it has received an offer in a reliable response to the (re)INVITE, and sent an answer to that offer.

5. Content of Offers and Answers

While [RFC3264] and [RFC3312] give some guidance, questions remain about exactly what should be included in an offer or answer. This is especially a problem when the common "hold" feature has been activated, and when there is the potential for a multimedia call.

Details of behavior depend on the capabilities and state of the User Agent. The kinds of recommendations that can be made are limited by the model of device capabilities and state that is presumed to exist.

This section focuses on a few key aspects of offers and answers that have been identified as troublesome, and will consider other aspects to be out of scope. This section considers:
- choice of supported media types and formats to include and exclude
- hold and resume of media

The following are out of scope for this document:
5.1. General Principle for Constructing Offers and Answers

A UA should send an offer that indicates what it, and its user, are interested in using/doing at that time, without regard for what the other party in the call may have indicated previously. This is the case even when the offer is sent in response to an INVITE or re-INVITE that contains no offer. (However in the case of re-INVITE the constraints of RFCs 3261 and 3264 must be observed.)

A UA should send an answer that includes as close an approximation to what the UA and its user are interested in doing at that time, while remaining consistent with the offer/answer rules of [RFC3264] and other RFCs.

NOTE: "at that time" is important. The device may permit the user to configure which supported media are to be used by default.

In some cases a UA may not have direct knowledge of what it is interested in doing at a particular time. If it is an intermediary it may be able to delegate the decision. In the worst case it may apply a default, such as assuming it wants to use all of its capabilities.

5.2. Choice of Media Types and Formats to Include and Exclude

5.2.1. Sending an Initial INVITE with Offer

When a UAC sends an initial INVITE with an offer, it has complete freedom to choose which media type(s) and media format(s) (payload types in the case of RTP) it should include in the offer.

The media types may be all or a subset of the media the UAC is capable of supporting, with the particular subset being determined by the design and configuration (e.g., via [I-D.ietf-sipping-config-framework]) of the UAC combined with input from the user interface of the UAC.

The media formats may be all or a subset of the media formats the UAC is capable of supporting for the corresponding media type, with the particular subset being determined by the design and configuration of the UAC combined with input from the user interface of the UAC.
Including all supported media formats will maximize the possibility that the other party will have a supported format in common. But including many can result in an unacceptably large SDP body.

5.2.2. Responding with an Offer when the Initial INVITE has no Offer

When a UAS has received an initial INVITE without an offer, it must include an offer in the first reliable response to the INVITE. It has largely the same options as when sending an initial INVITE with an offer, but there are some differences. The choice may be governed by both static (default) selections of media types as well as dynamic selections made by a user via interaction with the device while it is alerting.

NOTE: The offer may be sent in a provisional response, before the user of the device has been alerted and had an opportunity to select media options for the call. In this case the UAS cannot include any call-specific options from the user of the device. If there is a possibility that the user of the device will wish to change what is offered before answering the call, then special care should be taken. If PRACK and UPDATE are supported by caller and callee then an initial offer can be sent reliably, and changed with an UPDATE if the user desires a change. If PRACK and UPDATE are not supported then the initial offer cannot be changed until the call is fully established. In that case either the offer should be delayed until the 200 is sent, or else the offer should include the minimum set of media the user is able to select.

5.2.3. Answering an Initial INVITE with Offer

When a UAS receives an initial INVITE with an offer, what media lines the answer may contain is constrained by [RFC3264]. The answer must contain the same number of m-lines as the offer, and they must contain the same media types. Each media line may be accepted, by including a non-zero port number, or rejected by including a zero port number in the answer. The media lines that are accepted should typically be those that would have been offered had the INVITE not contained an offer, excluding those not offered.

The media formats the answer may contain are constrained by [RFC3264]. For each accepted m-line in the answer, there must be at least one media format in common with the corresponding m-line of the offer. The UAS may also include other media formats it is able to support at this time. However there is little benefit to including added types.

If the UAS does not wish to indicate support for any of the media types in a particular media line of the offer it must reject the
corresponding media line, by setting the port number to zero.

5.2.4. Answering when the Initial INVITE had no Offer

When a UAC has sent an initial INVITE without an offer, and then receives a response with the first offer, it should answer in the same way as a UAS receiving an initial INVITE with an offer.

5.2.5. Subsequent Offers and Answers

The guidelines above (Section 5.1 and Section 5.2.1 through Section 5.2.4) apply, but constraints in [RFC3264] must also be followed. The following are of particular note because they have proven troublesome:
- The number of m-lines may not be reduced in a subsequent offer. Previously rejected media streams must remain, or be reused to offer the same or a different stream. (Section 6 of [RFC3264].)
- In the o-line, only the version number may change, and if it changes it must increment by one from the one previously sent as an offer or answer. (Section 8 of [RFC3264].) If it doesn't change then the entire SDP body must be identical to what was previously sent as an offer or answer. Changing the o-line, except version number value, during the session is an error case. The behavior when receiving such a non-compliant offer/answer SDP body is implementation dependent. If a UA needs to negotiate a ‘new’ SDP session, it should use the INVITE/Replaces method.
- In the case of RTP, the mapping from a particular dynamic payload type number to a particular codec within that media stream (m-line) must not change for the duration of the session. (Section 8.3.2 of [RFC3264].)

NOTE: This may be impossible for a B2BUA to follow in some cases (e.g. 3pcc transfer) if it does not terminate media.

When the new offer is sent in response to an offerless (re)INVITE, it should be constructed according to the General Principle for Constructing Offers and Answers (Section 5.1): all codecs the UA is currently willing and able to use should be included, not just the ones that were negotiated by previous offer/answer exchanges. The same is true for media types – so if UA A initially offered audio and video to UA B, and they end up with only audio, and UA B sends an offerless (re)INVITE to UA A, A’s resulting offer should most likely re-attempt video, by reusing the zeroed m-line used previously.

NOTE: The behavior above is recommended, but it is not always achievable – for example in some interworking scenarios. Or, the offerer may simply not have enough resources to offer "everything" at that point. Even if the UAS is not able to offer any other SDP
that the one currently being used, it should not reject the re-
INVITE. Instead, it should generate an offer with the currently
used SDP with o-line unchanged.

5.3. Hold and Resume of media

[RFC3264] specifies (using non-normative language) that "hold" should be indicated in an established session by sending a new offer containing "a=sendonly" for each media stream to be held. An answerer is then to respond with "a=recvonly" to acknowledge that the hold request has been understood.

Note that the use of sendonly/recvonly is not limited to hold. These may be used for other reasons, such as devices that are only capable of sending or receiving. So receiving an offer with "a=sendonly" must not be treated as a certain indication that the offerer has placed the media stream on hold.

This model is based on an assumption that the UA initiating the hold will want to play Music on Hold, which is not always the case. A UA may, if desired, initiate hold by offering "a=inactive" if it does not intend to transmit any media while in hold status.

The rules of [RFC3264] constrain what may be in an answer when the offer contains "sendonly", "recvonly", or "inactive" in an a-line. But they do not constrain what must be in a subsequent offer. The General Principle for Constructing Offers and Answers (Section 5.1) is important here. The initiation of "hold" is a local action. It should reflect the desired state of the UA. It then affects what the UA includes in offers and answers until the local state is reset.

The receipt of an offer containing "a=sendonly" or "a=inactive" and the sending of a compatible answer should not change the desired state of the recipient. However, a UA that has been "placed on hold" may itself desire to initiate its own hold status, based on local input.

If UA2 has previously been "placed on hold" by UA1, via receipt of "a=sendonly", then it may initiate its own hold by sending a new offer containing "a=sendonly" to UA1. Upon receipt of that, UA1 will answer with "a=inactive" because that is the only valid answer that reflects its desire not to receive media.

NOTE: Section 8.4 of RFC3264 contains a conflicting recommendation that the offer contain "a=inactive" in this case. We interpret that recommendation to be non-normative. The use of "a=sendonly" in this case will never produce a worse outcome, and can produce a better outcome in useful cases.
Once in this state, to resume a two way exchange of media each side must reset its local hold status. If UA1 is first to go off hold it will then send an offer with "a=sendrecv". The UA2 will respond with its desired state of "a=sendonly" because that is a permitted response. When UA2 desires to also resume, it will send an offer with "a=sendrecv". In this case, because UA1 has the same desire it will respond with "a=sendrecv". In the same case, when UA2 receives the offer with "a=sendrecv", if it has decided it wants to reset its local hold but has not yet signaled the intent, it may send "a=sendrecv" in the answer.

If UA2 has been "placed on hold" by UA1 via receipt of "a=inactive", and subsequently wants to initiate its own hold, also using "a=inactive", it need not send a new offer, since the only valid response is "a=inactive" and that is already in effect. However, its local desired state will now be either "inactive" or "a=sendonly". This affects what it will send in future offers and answers.

If a UA has occasion to send another offer in the session, without any desire to change the hold status (e.g. in response to a re-INVITE without an offer, or when sending a re-INVITE to refresh the session timer) it should follow the General Principle for Constructing Offers and Answers (Section 5.1). If it previously initiated a "hold" by sending "a=sendonly" or "a=inactive" then it should offer that again. If it had not previously initiated "hold" then it should offer "a=sendrecv", even if it had previously been forced to answer something else. Without this behavior it is possible to get "stuck on hold" in some cases, especially when a third-party call controller is involved.

5.4. Behavior on receiving SDP with c=0.0.0.0

[RFC3264] specifies that an agent 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.

If a UA generates an answer to the offer received with c=0.0.0.0, the direction attribute of the accepted media stream in the answer must be based on direction attribute of the offered stream and rules specified in RFC 3264 to form the a-line in the answer. c=0.0.0.0 has no special meaning for the direction attribute of the accepted stream in the answer.

6. IANA Considerations

This document has no actions for IANA.
7. Security Considerations

There are not any security issues beyond the referenced RFCs.

8. Acknowledgement

The authors would like to thank Christer Holmberg, Rajeev Seth, Nataraju A B, Byron Campen, Jonathan Rosenberg, Gonzalo Camarillo and Shinji Okumura for their thorough reviews and comments. Many of their suggestions and ideas have been incorporated in this document.

9. References

9.1. Normative References


9.2. Informative References


Authors’ Addresses

Paul H. Kyzivat
Cisco Systems, Inc.
1414 Massachusetts Avenue
Boxborough, MA 01719
USA

Email: pkyzivat@cisco.com

Takuya Sawada
KDDI Corporation
3-10-10, Iidabashi, Chiyoda-ku
Tokyo
Japan

Email: tu-sawada@kddi.com