SIP (Session Initiation Protocol) Usage of Offer/Answer Model
draft-ietf-sipping-sip-offeranswer-03.txt

Status of this Memo

By submitting this Internet-Draft, each author represents that
any applicable patent or other IPR claims of which he or she is
aware have been or will be disclosed, and any of which he or she
becomes aware will be disclosed, in accordance with Section 6 of
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 November 28, 2007.

Abstract

SIP utilizes offer/answer model to establish and update multimedia
sessions. The descriptions on how to use offer/answer in SIP are
dispersed in the multiple RFCs. This document summarizes all the
current usage of offer/answer model in SIP communication.

Table of Contents

1. Summary of SIP usage of 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 offer/answer model. RFC 3264 [3] defines the
   offer/answer model, but does not specify which SIP message 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 the session description that conforms to the rules described in the standards-track RFCs can be interpreted as an offer or an answer. The rules for how to handle the offer/answer model are currently defined in several RFCs.

The offer/answer model defines the update of sessions. In SIP, dialog is used to match the offer/answer exchange to the session which is to be updated with it. In other words, only the offer/answer exchange in the SIP dialog can update the session which is managed with it.

1.1. Offer/Answer Exchange Pairs in SIP Messages

Currently, the rules on offer/answer model are defined in RFC 3261 [1], RFC 3262 [2] and RFC 3311 [4]. 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. Pattern 2 and pattern 4 can occur only when the INVITE request does not include an offer. '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 in the previous offer/answer exchange.

```
        Offer                Answer               RFC    Ini Est Early
      ----------------------------------------------------------------
1. INVITE Req.          2xx INVITE Resp.    RFC 3261  O   O    X
2. 2xx INVITE Resp.     ACK Req.            RFC 3261  O   O    X
3. INVITE Req.          1xx-rel INVITE Resp. RFC 3262  O   O    X
4. 1xx-rel INVITE Resp. PRACK Req.          RFC 3262  O   O    X
5. PRACK Req.           200 PRACK Resp.     RFC 3262  X   O    O
6. UPDATE Req.          2xx UPDATE Resp.    RFC 3311  X   O    O
```

Table 1. Summary of SIP Usage of Offer/Answer Model

In Table 1, '1xx-rel' corresponds to the reliable provisional
response which contains the 100rel option defined in RFC 3262 [2].

The 'Ini' column shows the ability to exchange the offer/answer to
initiate the session. 'O' indicates that the pattern can be used in
the initial offer/answer exchange, while 'X' indicates that it can
not. Only the initial INVITE transaction can be used to exchange the
offer/answer to establish 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.

1.2. Rejection against an Offer

How to reject an offer when it can not be accepted is not so clear
and some methods can not allow explicit rejection against an offer.
Corresponding to the patterns in Table 1, how to reject an offer is
shown in Table 2.

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)

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 followed by an UPDATE request possibly to
rearrange the session parameters if both ends support UPDATE method,

or alternatively terminate the dialog and send an error response to
the INVITE request. (Pattern 5)

When a UA receives a response with an offer which it can not accept,
a 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 a never-ending offer/answer loop.

```
   Offer                   Rejection
   -------------------------------------------------------
1. INVITE Req.            488 INVITE Response
2. 2xx INVITE Resp.       Answer in ACK Req. followed by new offer
                          OR termination of dialog
3. INVITE Req.            488 INVITE Response (same as Pattern 1.)
4. 1xx-rel INVITE Resp.   Answer in PRACK Req. followed by new offer
5. PRACK Req. (*)         200 PRACK Resp. followed by new offer
                          OR termination of dialog
6. UPDATE Req.            488 UPDATE Response
```

Table 2. Rejection against an Offer

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

1.3. 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.

2. Detailed Discussion on Offer/Answer Model for SIP

2.1. Offer/Answer for 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 RFC 3261 [1]. 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 makes the
rules complicated. 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. Session descriptions in an unreliable response that precedes a reliable response can be considered a "preview" of the session description that will be coming, and hence may be treated like an offer or 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 the different "preview" of answer in each dialog. UAs are expected to deal with this.

2.1.1. 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 INVITE 3-way handshake process.

```
 UAC                    UAS
  | F1  INVITE (SDP)    | <- The offer in offer/answer model
  |-------------------->|
  | F2     1xx (SDP)    | <- The SDP is not an official answer 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 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 offer
  |<-------------------| v in an UPDATE request.
  |                    |
  | F9 1xx-rel          | <- SDP should not be included in the
  |<-------------------|    subsequent 1xx-rel once offer/answer
  | F10   PRACK         |    has been completed.
  |-------------------->|
  | F11 2xx PRA         |
  |<-------------------|
  |                    |
  | F12 2xx INV         | <- SDP should not be included in the final
  |<-------------------|    response once offer/answer has been
  | F13    ACK          |    completed.
  |-------------------->|
```
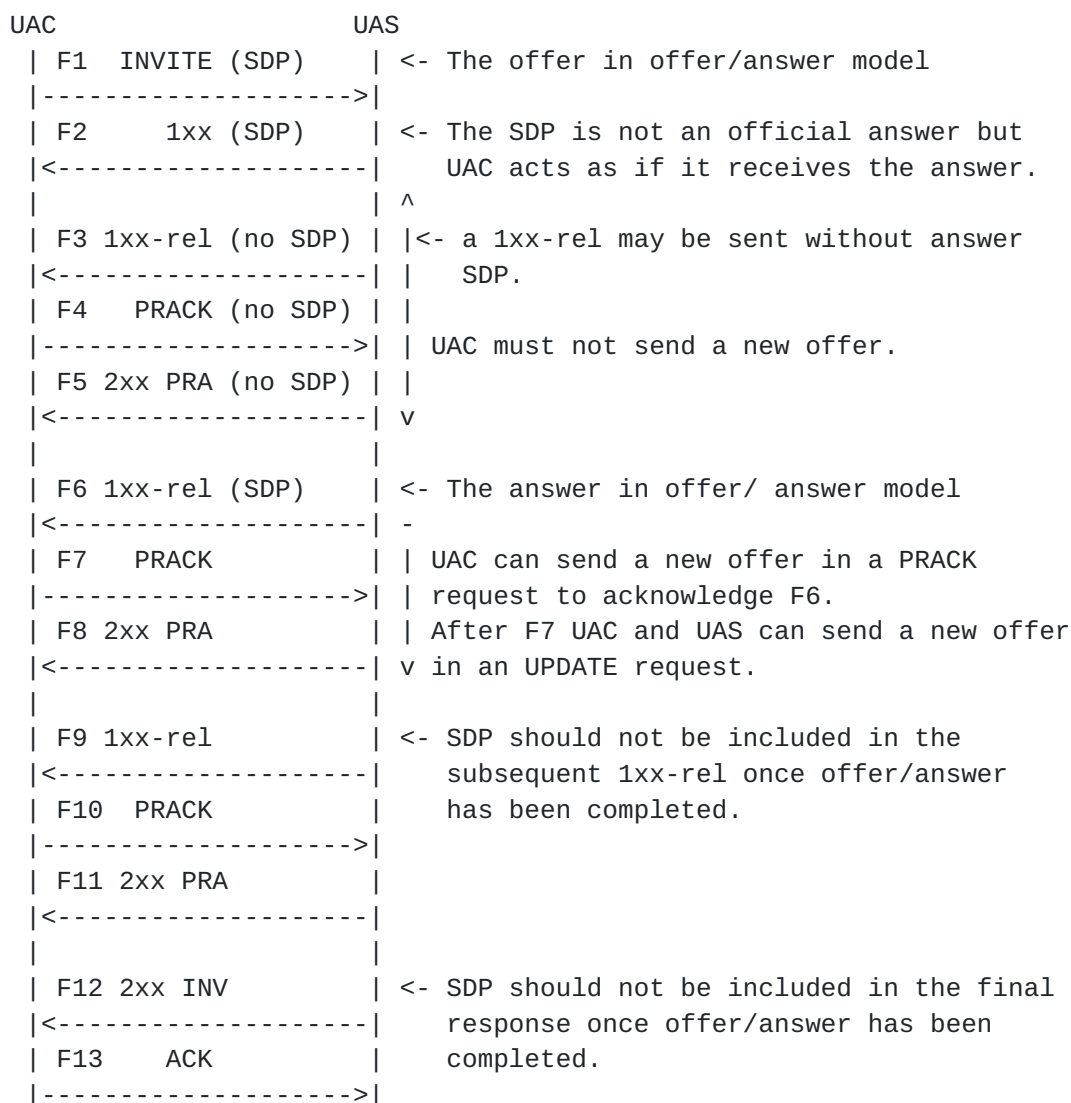
        Figure 1 Example of Offer/Answer with 100rel Extension (1)

   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, but is not officially the
   answer. Receiving F2, UAC should act as if it receives the answer.
   However, offer/answer exchange is not completed yet and 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, UAS
   must prepare to receive new offer from UAC with an UPDATE request or
   a PRACK request.

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

2.1.2. **INVITE request without SDP**

   When UAC does not include an SDP body in the INVITE request, it
   expects the offer to be received with the first reliable response.
   UAC will send the answer in the request to acknowledge the response,
   i.e. PRACK or ACK request for the reliable response. Other than that,
   no offer/answer exchanges can occur in the INVITE 3-way handshake
   process.

   For example, in Figure 2, only the SDP in F3 is the offer. The SDP in
   the non-reliable response (F2) is the preview of the offer and must
   be the same as the offer in F3, but is not officially the offer.
   Receiving F2, UAC can act as if it receives the offer. However, the
   official offer is not received until it receives the first reliable
   response. The first reliable response (F3) must include an SDP as an
   offer.

   UAS should not include SDP in the responses F6 and F9. However, UAC
   should prepare to receive SDP bodies in F6 and/or F9, and just ignore
   them for the case that the peer does not conform to the recommended
   implementation.

```
   UAC                     UAS
     | F1  INVITE (no SDP) |
     |-------------------->|
     | F2     1xx (SDP)    | <- SDP may be included but it is not the
     |<--------------------|    offer. UAC may act as if it receives
     |                     |    the offer.
     | F3 1xx-rel (SDP)    | <- The first 1xx-rel must contain an SDP
     |<--------------------|    as the offer.
     | F4   PRACK (SDP)    | <- A PRACK request to the first 1xx-rel
     |-------------------->|    must contain an SDP as the answer.
     | F5 2xx PRA (no SDP) | -
     |<--------------------| |
     |                     | |
     | F6 1xx-rel (no SDP) | <- The subsequent 1xx-rel should not
     |<--------------------| |  contain an SDP.
     | F7    PRACK         | |
     |-------------------->| | UAC can send a new offer in an UPDATE
     | F8 2xx PRA          | | request after F4.
     |<--------------------| v
     |                     |
     | F9 2xx INV (no SDP) | <- The final response should not
     |<--------------------|    contain an SDP.
     | F10    ACK          |
     |-------------------->|
```

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

2.2. **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 UA sending an INVITE request:

UA can send an UPDATE request with a new offer if both ends support
the UPDATE method. Support for the UPDATE method must be declared in
an Allow header in some prior messages in the dialog.

UA can send a PRACK request with a new offer only when acknowledging
the reliable provisional response with the answer to the offer in the
INVITE request. Compared to using the UPDATE method, using PRACK can
save messages to be exchanged between the UAs. However, as a PRACK
request should not be rejected, UA is recommended to send a PRACK
request only when it has strong reasons to assume the receiver will
accept it. For example, the procedure used in precondition extension
[5] is a case where a PRACK request should be used for updating the
session status in the early dialog.

From UA receiving an INVITE request:

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

2.3. **Offer/Answer Exchange in Established Dialog**

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

The UPDATE method is simpler and can save at least one message
compared with 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. **Exceptional Case Handling**

In RFC 3264 [3], the following restrictions are defined with regard
to sending a new offer.

>   "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 seems to be two
possible 'exceptional' cases to be considered in SIP offer/answer
usage, which are the 'message crossing' case and the 'glare' case.
One of the reasons why the usage of a SIP method to exchange
offer/answer needs to be carefully restricted in the RFCs is to make
sure that UA can detect and handle appropriately the 'exceptional'
cases to avoid the confusion.

3.1. **Message Crossing Case Handling**

When message packets are crossed in the transport network, an offer
may be received before the answer for the previous offer/answer
exchange as described in Figure 3. In such a case, UA A must detect
the session description of the offer2 is not the answer to the offer1.

```
 A                     B
 |offer1               |
 |----------------->|
 |           answer1|
 |<------\  /-------|
 |        \/        |
 |        /\  offer2|
 |<------/  \-------|
```
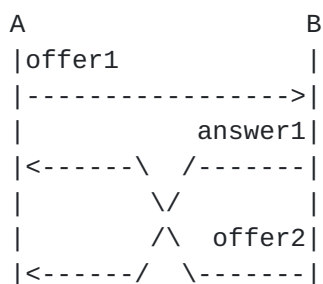
Figure 3 Message Crossing Case

When offer2 is in an UPDATE request or a re-INVITE request, a session
description can never be the answer. Then UA A must reject the
message including offer2 with a 491 response with Retry-After header
field.

When offer2 is in a PRACK request, that is, when a PRACK request to
acknowledge the reliable provisional response with an answer to the
offer in the INVITE request contains a session description, UA A
knows it is an offer. As a PRACK request should not be rejected, UA A
is recommended to wait for the answer1 before sending a PRACK
response with the answer to the offer2. Note that if UA A does not
send a new offer until the reliable provisional response with an

answer to the offer in the INVITE request is acknowledged with a
PRACK request, this case never happens. Therefore, to make
implementations simple, a UA acting as a UAS for an INVITE
transaction is recommended not to send an UPDATE request with an
offer until the reliable response with an answer to the offer in the
INVITE request is acknowledged with a PRACK request.

When offer2 is in a reliable provisional response or a successful
final response, UA A knows it is not the answer to the offer1. For a
reliable response to an initial INVITE request, this case never
happens. For a reliable response to a re-INVITE request, UA A can
detect the offer2 is not the answer1. In this case, UA A can not
reject offer2 in a reliable response, it is recommended to wait for
answer1 before sending a PRACK request with the answer to offer2.
Note that this case only occurs when UA A, while waiting for an
answer, sends an INVITE request without session description.

3.2. Glare Case Handling

When both ends in a dialog send a new offer at nearly the same time,
UA may receive a new offer before it receives the answer to the offer
it sent as described in Figure 4. This case is called a 'glare' case
in general.

```
 A                      B
 |offer1      offer2|
 |-------\  /-------|
 |        \/        |
 |        /\        |
 |<------/  \------>|
```

Figure 4 Glare Case

When offer2 is in an 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), the PRACK may be
accepted with 200 or may be rejected with a 491 response. A 491
response is valid to satisfy the offer/answer model but it may delay
the completion of the reliable response transfer mechanism or, in
worst case, may result in the failure to complete the SIP transaction
because there is no clear retry rule when a PRACK request is rejected
with a 491 response. 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.

To avoid a glare condition involving an offer in a response, 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.

[4](#). Content of Offers and Answers

While RFCs 3264[3] and 3312[5] 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:

- NAT traversal and ICE

- specific codecs and their parameters

- the negotiation of secure media streams

- grouping of media streams

- preconditions

[4.1](#). 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 RFC 3264[3] 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.

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

4.2.1. Sending 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 [6] 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
[6] 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.

4.2.2. Responding with Offer when 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.

4.2.3. Answering Initial INVITE with Offer

When a UAS receives an initial INVITE with an offer, what media lines
the answer may contain is constrained by RFC 3264.[3] 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, but with those not offered removed.

The media formats the answer may contain are constrained by RFC 3264
[3]. 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.

4.2.4. Answering when 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.

4.2.5. Subsequent Offers and Answers

The guidelines above (sections 4.1. and 4.2.1. through 4.2.4.) apply,
but constraints in RFC 3264 [3] must also be followed. The following
are of particular note because they have proven troublesome:

o 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.

o 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. 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 is implementation dependent. If a
  UA needs to negotiate a 'new' SDP session, it should use the
  INVITE/Replaces method.

o 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.

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

4.3. Hold and Resume of media

RFC 3264 [3] specifies (non-normatively) 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 RFC 3264 [3] 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 4.1.)
is important here. The initiation of "hold" is a local action. It
should affect 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.

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 "a=sendrecv".

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". 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 4.1.) 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. Remaining Issues or Best Practices on Offer/Answer

This document clarifies the offer/answer usage in SIP and summarizes the correct or recommended behaviors along with the existing RFCs. To create any new normative behaviors beyond these RFCs is not the intent of this document.

However, through the scrutiny of the offer/answer model in SIP, some issues are found to be unresolved within the current set of RFCs. Those remaining issues are described in this section mainly for further study.

5.1. Rejecting PRACK Offer

As stated in section 1.2. and 2.2. , it is recommended not to send an offer in a PRACK request unless UAC has strong reasons to assume the receiver will accept it. Even so, there may be the cases when the UAS has to reject the offer for some reason. The current RFCs do not provide the way to reject the offer and at the same time to acknowledge the reliable response.

Several ideas were presented to resolve this issue, such as sending 2xx PRACK response without SDP to reject the offer, or sending an SDP with decreased o-line version value. Some of the candidates may also be adapted as a way to reject an unacceptable offer in a response. Anyway, those candidates violate the current rules and lose backward compatibility to some extent (e.g. section 5 of RFC 3262). It is beyond the scope of this document and remains for further study.

> NOTE: Deprecation of the usage of offer in PRACK may be another solution. As the precondition mechanism specification [2] explicitly shows a usage of sending offer in PRACK, its deprecation could cause backward compatibility issues.

5.2. Commit/Rollback of Offer/Answer on Unsuccessful re-INVITE Transaction

When a re-INVITE transaction fails, often the dialog remains with the session bound to it. The issue here is what the session status is if offer/answer exchange has been completed (if a session description has been sent in a reliable provisional response to the re-INVITE request), or if subsequent offer/answer exchanges have taken place (using UPDATE or PRACK transactions), before the re-INVITE transaction is terminated with a final error response (Figure 5). One option is to take those offer/answer exchanges not committed yet and to make the session status rollback to the one before re-INVITE transaction was initiated. Another option is to take those exchanges committed and to keep the session status as it is even after re-INVITE fails. There is no clear consensus on which one is the correct behavior.

There are some cases where it is useful to exchange offer(s)/answer(s) even before re-INVITE completes. The case of adding a new media (like adding video to audio only session) which requires permission from the peer through some user interaction is one example. Precondition procedures can be another case which may require several offer/answer exchanges in one re-INVITE transaction.
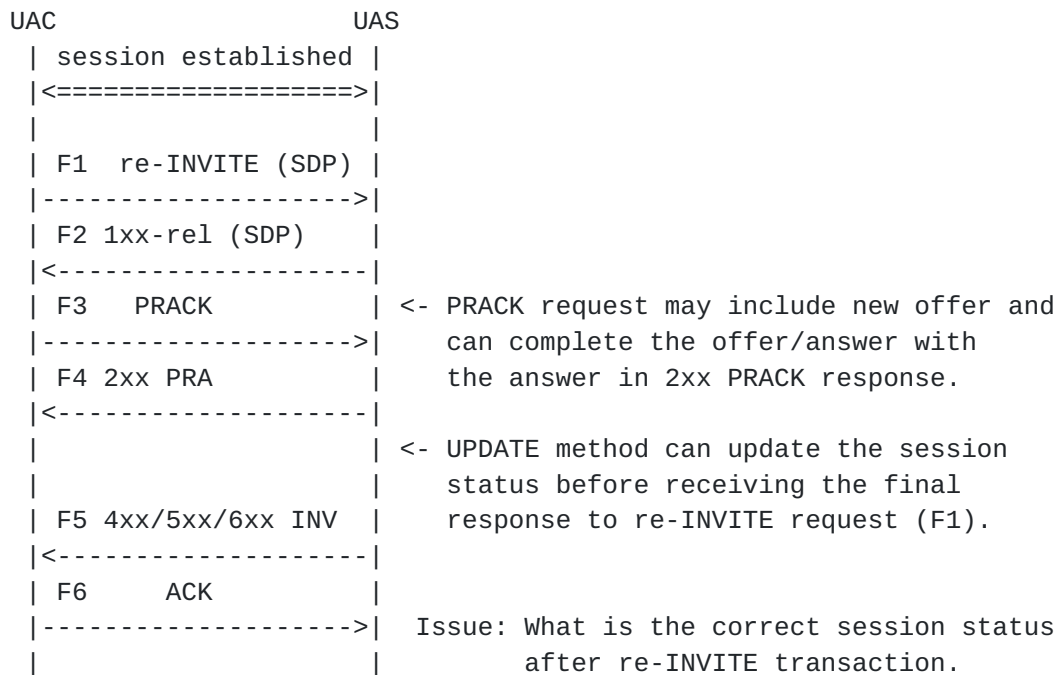
```
   UAC                     UAS
    | session established |
    |<==================>|
    |                    |
    | F1  re-INVITE (SDP) |
    |------------------->|
    | F2 1xx-rel (SDP)   |
    |<-------------------|
    | F3    PRACK        | <- PRACK request may include new offer and
    |------------------->|     can complete the offer/answer with
    | F4 2xx PRA         |     the answer in 2xx PRACK response.
    |<-------------------|
    |                    | <- UPDATE method can update the session
    |                    |     status before receiving the final
    | F5 4xx/5xx/6xx INV |     response to re-INVITE request (F1).
    |<-------------------|
    | F6     ACK         |
    |------------------->|  Issue: What is the correct session status
    |                    |          after re-INVITE transaction.
```

Figure 5 Commit/Rollback Issue with re-INVITE transaction

To make bad things worse, if a new offer from UAC and the final
response to re-INVITE are sent at nearly the same time, the UAS can
not know whether this new offer was sent before or after UAC received
the final failure response (Figure 6). Note that the ACK request to
the failure response is sent hop-by-hop basis, therefore even after
receiving the ACK request, UAS can not make sure that UPDATE request
was sent after the final response had been reached to the other end.

Sending a new UPDATE request from UAC to synchronize the status
anytime after the re-INVITE fails may be a good option. This solution,
however, requires that the UPDATE method be supported by both ends
and needs care to avoid flapping when each end tries to advertise
their different views of the session status.

The proper handling of this issue is undefined by existing standards.
Resolution is beyond the scope of this document, and will require a
new normative document. Such a document is the responsibility of the
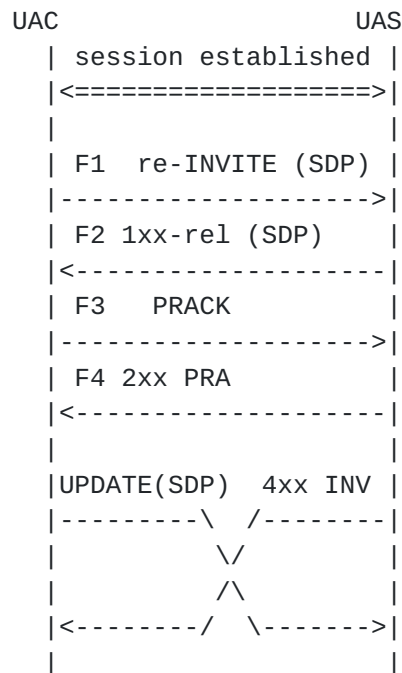SIP working group, and is for further study.

```
UAC                    UAS
  | session established |
  |<==================>|
  |                    |
  | F1  re-INVITE (SDP) |
  |------------------->|
  | F2 1xx-rel (SDP)    |
  |<-------------------|
  | F3    PRACK        |
  |------------------->|
  | F4 2xx PRA         |
  |<-------------------|
  |                    |
  |UPDATE(SDP)  4xx INV |
  |---------\  /--------|
  |          \/        |
  |          /\        |
  |<--------/  \------>|
  |                    |
```

                 Figure 6 Commit/Rollback Issue with Race Condition

5.3. Offer in a Reliable Response

In RFC 3261, it is stated that when an INVITE is sent without an offer, the first reliable response MUST contain an offer. There was discussion on whether this rule can be loosened up. There is no clear explanation why this restriction is defined. However, this rule will be left as it is, unless the strong necessity to loose it up will come up in the future.

6. Add New Offer/Answer Usage in SIP

It is not recommended to add new SIP methods for the offer/answer exchange beyond the ways described in this document. However, it may be requested to have new offer/answer exchange methods as SIP extensions evolve. In this clause, what should be taken into considerations is noted.

6.1. Explicit Usage

New method definitions should define offer/answer usage explicitly without any ambiguity.

6.2. Rejection of an Offer

New method definitions should define how to reject an offer where possible.

6.3. Backward Compatibility

New methods must keep backward compatibility.

6.4. Exceptional Case Handling

New methods should take care of how to handle exceptional cases,
message crossing case and glare case.

7. Security Considerations

There are not any security issues beyond the referenced RFCs.

8. References

8.1. Normative References

[1] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A.,
      Peterson, J., Sparks, R., Handley, M. and E. Schooler, "SIP:
      Session Initiation Protocol", RFC 3261, June 2002.

[2] Rosenberg, J. and H. Schulzrinne, "Reliability of Provisional
      Responses in the Session Initiation Protocol (SIP)", RFC 3262,
      June 2002.

[3] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with
      SDP", RFC 3264, June 2002.

[4] Rosenberg, J., "The Session Initiation Protocol (SIP) UPDATE
      Method", RFC 3311, September 2002.

[5] Camarillo, G., Marshall, W., and J. Rosenberg, "Integration of
      Resource Management and Session Initiation Protocol (SIP)", RFC
      3312, October 2002.

8.2. Informative References

[6] Hilt, V., Camarillo, G., and J. Rosenberg, "A User Agent
      Profile Data Set for Media Policy", draft-ietf-sipping-media-
      policy-dataset-04 (work in progress), May 2007.

Author's Addresses

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

   Email: tu-sawada@kddi.com

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

   Email: pkyzivat@cisco.com

Acknowledgment