          **Session Description Protocol (SDP) Offer/Answer procedures for
                 Interactive Connectivity Establishment (ICE)
                      draft-ietf-mmusic-ice-sip-sdp-16**

Abstract

   This document describes Session Description Protocol (SDP) Offer/
   Answer procedures for carrying out Interactive Connectivity
   Establishment (ICE) between the agents.

Status of This Memo

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

   Internet-Drafts are working documents of the Internet Engineering
   Task Force (IETF). Note that other groups may also distribute
   working documents as Internet-Drafts. The list of current Internet-
   Drafts is at http://datatracker.ietf.org/drafts/current/.

   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."

   This Internet-Draft will expire on May 27, 2018.

Table of Contents

1. Introduction

   This document describes how Interactive Connectivity Establishment
   (ICE) is used with Session Description Protocol (SDP) offer/answer
   [RFC3264]. The ICE specification [ICE-BIS] describes procedures that
   are common to all usages of ICE and this document gives the
   additional details needed to use ICE with SDP offer/answer.

2. Terminology

   The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
   "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and
   "OPTIONAL" in this document are to be interpreted as described in RFC
   2119 [RFC2119].

Readers should be familiar with the terminology defined in [RFC3264], in [RFC7656], in [ICE-BIS] and the following:

Default Destination/Candidate: The default destination for a component of a media stream is the transport address that would be used by an agent that is not ICE aware. A default candidate for a component is one whose transport address matches the default destination for that component. For the RTP component, the default IP address is in the "c=" line of the SDP, and the port is in the "m=" line. For the RTCP component, the address and port are indicated using the "a=rtcp" attribute defined in [RFC3605], if present; otherwise, the RTCP component address is same as the address of the RTP component, and its port is one greater than the port of the RTP component.

3. ICE Candidate Exchange and Offer/Answer Mapping

[ICE-BIS] defines ICE candidate exchange as the process for ICE agents (Initiator and Responder) to exchange their candidate information required for ICE processing at the agents. For the purposes of this specification, the candidate exchange process corresponds to the [RFC3264] Offer/Answer protocol and the terminologies offerer and answerer correspond to the initiator and responder terminologies from [ICE-BIS] respectively.

4. SDP Offer/Answer Procedures

4.1. Initial Offer/Answer Exchange

4.1.1. Sending the Initial Offer

The offerer shall follow the procedures defined in section 5 of [ICE-BIS] to gather, prioritize and eliminate the redundant candidates. It then chooses the default candidates and encodes them in the SDP to be sent to its peer, the answerer.

4.1.1.1. Choosing Default Candidates

A candidate is said to be default if it would be the target of media from a non-ICE peer; that target is called the DEFAULT DESTINATION. An agent MUST choose a set of candidates, one for each component of each in-use media stream, to be default. A media stream is in-use if it does not have a port of zero (which is used in RFC 3264 to reject a media stream). Consequently, a media stream is in-use even if it is marked as a=inactive [RFC4566] or has a bandwidth value of zero.

An agent may choose any type of the candidate as the default, if the chosen candidates increases the likelihood of success with the peer

that is being contacted if ICE is not being used. It is recommended
that, when multiple candidates are used, UDP based candidates SHOULD
be included wherever possible and default candidate SHOULD be chosen
from one of those UDP candidates. The proto value MUST match the
transport protocol associated with the default candidate. If UDP
transport is used for the default candidate, the 'proto' value MUST
include UDP and the 'proto' value MUST be TCP when the transport is
TCP for the default candidate.

Since it is RECOMMENDED that default candidates be chosen based on
the likelihood of those candidates to work with the peer that is
being contacted if ICE is not being used. Many factors may influence
such a decision in a given agent. In scenarios where the agent is
fully aware of its peer's location and can reach the peer directly,
choosing the host candidates as the default may well be sufficient.
If the network configuration under which the agents operates is
static and known beforehand, either the host or the server reflexives
candidates can serve as the default candidates (depending on if a
given agent is behind NAT and their reachability). If the agent is
completely unaware of the peer's location or no assumptions can be
made of network characteristics and the connectivity, the relayed
candidates might be the only option as the default candidate. Having
the decision of choosing the default candidate as a configurable
option in the implementations might provide agents the flexibility to
take into account the aforementioned criteria. Barring such
configuration flexibility, it is RECOMMENDED that the default
candidates be the relayed candidates (if relayed candidates are
available), server reflexive candidates (if server reflexive
candidates are available), and finally host candidates.

4.1.1.2. Encoding the SDP

The process of encoding the SDP is identical between full and lite
implementations.

The agent will include an "m=" line for each Source Stream [RFC7656]
it wishes to use. The ordering of source streams in the SDP is
relevant for ICE. ICE will perform its connectivity checks for the
first "m=" line first, and consequently media will be able to flow
for that stream first. Agents SHOULD place their most important
source stream, if there is one, first in the SDP.

There will be a candidate attribute for each candidate for a
particular source stream. Section 5 provides detailed rules for
constructing this attribute.

STUN connectivity checks between agents are authenticated using the
short-term credential mechanism defined for STUN [RFC5389]. This

mechanism relies on a username and password that are exchanged
through protocol machinery between the client and server. The
username fragment and password are exchanged in the ice-ufrag and
ice-pwd attributes, respectively.

If an agent is a lite implementation, it MUST include an "a=ice-lite"
session-level attribute in its SDP to indicate this. If an agent is
a full implementation, it MUST NOT include this attribute.

Section 10 of [ICE-BIS] defines a new ICE option, 'ice2'. This
option is used by ICE Agents to indicate their compliancy with
[ICE-BIS] specification as compared to the [RFC5245]. If the
Offering agent is a [ICE-BIS] compliant implementation, a session
level ICE option to indicate the same (via the "a=ice-options:ice2"
SDP line) MUST be included.

The default candidates are added to the SDP as the default
destination for media. For source streams based on RTP, this is done
by placing the IP address and port of the RTP candidate into the "c="
and "m=" lines, respectively. If the agent is utilizing RTCP and if
RTCP candidate is present and is not equal to the same address and
the next higher port number of the RTP candidate, the agent MUST
encode the RTCP candidate using the a=rtcp attribute as defined in
[RFC3605]. If RTCP is not in use, the agent MUST signal that using
b=RS:0 and b=RR:0 as defined in [RFC3556]

The transport addresses that will be the default destination for
media when communicating with non-ICE peers MUST also be present as
candidates in one or more a=candidate lines.

ICE provides for extensibility by allowing an offer or answer to
contain a series of tokens that identify the ICE extensions used by
that agent. If an agent supports an ICE extension, it MUST include
the token defined for that extension in the ice-options attribute.

The following is an example SDP message that includes ICE attributes
(lines folded for readability):

```
   v=0
   o=jdoe 2890844526 2890842807 IN IP4 10.0.1.1
   s=
   c=IN IP4 192.0.2.3
   t=0 0
   a=ice-options:ice2
   a=ice-pwd:asd88fgpdd777uzjYhagZg
   a=ice-ufrag:8hhY
   m=audio 45664 RTP/AVP 0
   b=RS:0
   b=RR:0
   a=rtpmap:0 PCMU/8000
   a=candidate:1 1 UDP 2130706431 10.0.1.1 8998 typ host
   a=candidate:2 1 UDP 1694498815 192.0.2.3 45664 typ srflx raddr
    10.0.1.1 rport 8998
```

Once an agent has sent its offer or its answer, that agent MUST be
prepared to receive both STUN and media packets on each candidate.
As discussed in section 12.1 of [ICE-BIS], media packets can be sent
to a candidate prior to its appearance as the default destination for
media in an offer or answer.

4.1.2. Receiving the Initial Offer

On receiving the offer, the answerer verifies the support for ICE
(section 5.4 of [ICE-BIS]), determines its role (section 6.1.1 of
[ICE-BIS]), gathers candidates (section 5 of [ICE-BIS]), encodes the
candidates in an SDP answer and sends it to its peer, the offerer.
The answerer shall then follow the steps defined in sections 6.1.3
and 6.1.4 of [ICE-BIS] to schedule the ICE connectivity checks.

The below sub-sections provide additional requirements associated
with the processing of the offerer's SDP pertaining to this
specification.

4.1.2.1. ICE Option "ice2" considerations

If the SDP offer contains a session level ICE option, "ice2" , and if
the answering ICE Agent is also an [ICE-BIS] compliant
implementation, then the generated SDP answer MUST include the
session level "a=ice-options:ice2" SDP line.

4.1.2.2. Choosing Default Candidates

The process for selecting default candidates at the answerer is
identical to the process followed by the offerer, as described in
Section 4.1.1.1 for full implementations in this specification and
section 5.2 of [ICE-BIS] for lite implementations.

4.1.2.3. ICE Mismatch

The agent will proceed with the ICE procedures defined in [ICE-BIS] and this specification if, for each media stream in the SDP it received, the default destination for each component of that media stream appears in a candidate attribute. For example, in the case of RTP, the IP address and port in the "c=" and "m=" lines, respectively, appear in a candidate attribute and the value in the rtcp attribute appears in a candidate attribute.

If this condition is not met, the agent MUST process the SDP based on normal RFC 3264 procedures, without using any of the ICE mechanisms described in the remainder of this specification with the following exceptions:

1. The agent MUST follow the rules of section 11 of [ICE-BIS], which describe keepalive procedures for all agents.

2. If the agent is not proceeding with ICE because there were a=candidate attributes, but none that matched the default destination of the media stream, the agent MUST include an a=ice-mismatch attribute in its answer.

3. If the default candidates were relayed candidates learned through a TURN server, the agent MUST create permissions in the TURN server for the IP addresses learned from its peer in the SDP it just received. If this is not done, initial packets in the media stream from the peer may be lost.

4.1.2.4. Determining Role

In unusual cases, described in Appendix C, it is possible for both agents to mistakenly believe they are controlled or controlling. To resolve this, each agent MUST select a random number, called the tie-breaker, uniformly distributed between 0 and (2**64) - 1 (that is, a 64-bit positive integer). This number is used in connectivity checks to detect and repair this case, as described in section 7.1.3 of [ICE-BIS].

4.1.3. Receipt of the Initial Answer

On receiving the SDP answer, the offerer performs steps similar to answerer's processing of the offer. The offerer verifies the answerer's ICE support determines, its role, and processes the answerer's candidates to schedule the connectivity checks (section 7 of [ICE-BIS]).

If the offerer had included the "ice2" ICE Option in the offer and
the SDP answer also includes a similar session level ICE option, then
the peers are [ICE-BIS] compliant implementations. On the other
hand, if the SDP Answer lacks such a ICE option, the offerer defaults
to the procedures that are backward compatible with the [RFC5245]
specification.

4.1.3.1. ICE Mismatch

The logic at the offerer is identical to that of the answerer as
described in section 5.4 of [ICE-BIS], with the exception that an
offerer would not ever generate a=ice-mismatch attributes in an SDP.

In some cases, the answer may omit a=candidate attributes for the
media streams, and instead include an a=ice-mismatch attribute for
one or more of the media streams in the SDP. This signals to the
offerer that the answerer supports ICE, but that ICE processing was
not used for the session because a signaling intermediary modified
the default destination for media components without modifying the
corresponding candidate attributes. See Section 11.2.2 for a
discussion of cases where this can happen. This specification
provides no guidance on how an agent should proceed in such a failure
case.

4.1.4. Performing Connectivity Checks

The possibility for role conflicts described in section 7.3.1.1 of
[ICE-BIS] applies to this usage and hence all full agents MUST
implement the role conflict repairing mechanism. Also both full and
lite agents MUST utilize the ICE-CONTROLLED and ICE-CONTROLLING
attributes as described in section 7.1.3 of [ICE-BIS].

4.1.5. Concluding ICE

Once the state of each check list is Completed, If an agent is
controlling, it examines the highest-priority nominated candidate
pair for each component of each media stream. If any of those
candidate pairs differ from the default candidate pairs in the most
recent offer/answer exchange, the controlling agent MUST generate an
updated offer as described in Section 4.2.

However, If the support for 'ice2' ICE Option is in use, the highest-
priority nominated candidate is noted and sent in the subsequent
offer/answer exchange as the default candidate and no updated offer
is needed to fix the default candidate.

4.2. Subsequent Offer/Answer Exchanges

   Either agent MAY generate a subsequent offer at any time allowed by
   [RFC3264]. This section defines rules for construction of subsequent
   offers and answers.

   Should a subsequent offer fail, ICE processing continues as if the
   subsequent offer had never been made.

4.2.1. Generating the Offer

4.2.1.1. Procedures for All Implementations

4.2.1.1.1. ICE Restarts

   An agent MAY restart ICE processing for an existing media stream as
   defined in section 9 of [ICE-BIS].

   The rules governing the ICE restart imply that setting the IP address
   in the "c=" line to 0.0.0.0 will cause an ICE restart. Consequently,
   ICE implementations MUST NOT utilize this mechanism for call hold,
   and instead MUST use a=inactive and a=sendonly as described in
   [RFC3264].

   To restart ICE, an agent MUST change both the ice-pwd and the ice-
   ufrag for the media stream in an offer. Note that it is permissible
   to use a session-level attribute in one offer, but to provide the
   same ice-pwd or ice-ufrag as a media-level attribute in a subsequent
   offer. This is not a change in password, just a change in its
   representation, and does not cause an ICE restart.

   An agent sets the rest of the fields in the SDP for this media stream
   as it would in an initial offer of this media stream (see
   Section 4.1.1.2). Consequently, the set of candidates MAY include
   some, none, or all of the previous candidates for that stream and MAY
   include a totally new set of candidates.

4.2.1.1.2. Removing a Media Stream

   If an agent removes a media stream by setting its port to zero, it
   MUST NOT include any candidate attributes for that media stream and
   SHOULD NOT include any other ICE-related attributes defined in
   Section 5 for that media stream.

4.2.1.1.3. Adding a Media Stream

   If an agent wishes to add a new media stream, it sets the fields in
   the SDP for this media stream as if this was an initial offer for
   that media stream (see Section 4.1.1.2). This will cause ICE
   processing to begin for this media stream.

4.2.1.2. Procedures for Full Implementations

   This section describes additional procedures for full
   implementations, covering existing media streams.

4.2.1.2.1. Existing Media Streams with ICE Running

   If an agent generates an updated offer including a media stream that
   was previously established, and for which ICE checks are in the
   Running state, the agent follows the procedures defined here.

   An agent MUST include candidate attributes for all local candidates
   it had signaled previously for that media stream. The properties of
   that candidate as signaled in SDP -- the priority, foundation, type,
   and related transport address -- SHOULD remain the same. The IP
   address, port, and transport protocol, which fundamentally identify
   that candidate, MUST remain the same (if they change, it would be a
   new candidate). The component ID MUST remain the same. The agent
   MAY include additional candidates it did not offer previously (see
   section 4.2.4.1.1), but which it has gathered since the last offer/
   answer exchange, including peer reflexive candidates.

   The agent MAY change the default destination for media. As with
   initial offers, there MUST be a set of candidate attributes in the
   offer matching this default destination.

4.2.1.2.2. Existing Media Streams with ICE Completed

   If an agent generates an updated offer including a media stream that
   was previously established, and for which ICE checks are in the
   Completed state, the agent follows the procedures defined here.

   The default destination for media (i.e., the values of the IP
   addresses and ports in the "m=" and "c=" lines used for that media
   stream) MUST be the local candidate from the highest-priority
   nominated pair in the valid list for each component.

   The agent MUST include candidate attributes for candidates matching
   the default destination for each component of the media stream, and
   MUST NOT include any other candidates.

In addition, if the agent is controlling, it MUST include the
a=remote-candidates attribute for each media stream whose check list
is in the Completed state. The attribute contains the remote
candidates from the highest-priority nominated pair in the valid list
for each component of that media stream. It is needed to avoid a
race condition whereby the controlling agent chooses its pairs, but
the updated offer beats the connectivity checks to the controlled
agent, which doesn't even know these pairs are valid, let alone
selected. See Appendix B for elaboration on this race condition.

4.2.1.3. Procedures for Lite Implementations

4.2.1.3.1. Existing Media Streams with ICE Running

This section describes procedures for lite implementations for
existing streams for which ICE is running.

A lite implementation MUST include all of its candidates for each
component of each media stream in an a=candidate attribute in any
subsequent offer. These candidates are formed identically to the
procedures for initial offers, as described in section 5.2 of
[ICE-BIS].

A lite implementation MUST NOT add additional host candidates in a
subsequent offer. If an agent needs to offer additional candidates,
it MUST restart ICE.

The username fragments, password, and implementation level MUST
remain the same as used previously. If an agent needs to change one
of these, it MUST restart ICE for that media stream.

4.2.1.3.2. Existing Media Streams with ICE Completed

If ICE has completed for a media stream, the default destination for
that media stream MUST be set to the remote candidate of the
candidate pair for that component in the valid list. For a lite
implementation, there is always just a single candidate pair in the
valid list for each component of a media stream. Additionally, the
agent MUST include a candidate attribute for each default
destination.

Additionally, if the agent is controlling (which only happens when
both agents are lite), the agent MUST include the a=remote-candidates
attribute for each media stream. The attribute contains the remote
candidates from the candidate pairs in the valid list (one pair for
each component of each media stream).

4.2.2. Receiving the Offer and Generating an Answer

4.2.2.1. Procedures for All Implementations

When receiving a subsequent offer within an existing session, an
agent MUST reapply the verification procedures in Section 4.1.2.3
without regard to the results of verification from any previous
offer/answer exchanges. Indeed, it is possible that a previous
offer/answer exchange resulted in ICE not being used, but it is used
as a consequence of a subsequent exchange.

4.2.2.1.1. Detecting ICE Restart

If the offer contained a change in the a=ice-ufrag or a=ice-pwd
attributes compared to the previous SDP from the peer, it indicates
that ICE is restarting for this media stream. If all media streams
are restarting, then ICE is restarting overall.

If ICE is restarting for a media stream:

o The agent MUST change the a=ice-ufrag and a=ice-pwd attributes in
   the answer.

o The agent MAY change its implementation level in the answer.

An agent sets the rest of the fields in the SDP for this media stream
as it would in an initial answer to this media stream (see
Section 4.1.1.2). Consequently, the set of candidates MAY include
some, none, or all of the previous candidates for that stream and MAY
include a totally new set of candidates.

4.2.2.1.2. New Media Stream

If the offer contains a new media stream, the agent sets the fields
in the answer as if it had received an initial offer containing that
media stream (see Section 4.1.1.2). This will cause ICE processing
to begin for this media stream.

4.2.2.1.3. Removed Media Stream

If an offer contains a media stream whose port is zero, the agent
MUST NOT include any candidate attributes for that media stream in
its answer and SHOULD NOT include any other ICE-related attributes
defined in Section 5 for that media stream.

4.2.2.2. **Procedures for Full Implementations**

   Unless the agent has detected an ICE restart from the offer, the
   username fragments, password, and implementation level MUST remain
   the same as used previously. If an agent needs to change one of
   these it MUST restart ICE for that media stream by generating an
   offer; ICE cannot be restarted in an answer.

   Additional behaviors depend on the state of ICE processing for that
   media stream.

4.2.2.2.1. **Existing Media Streams with ICE Running and no remote-
             candidates**

   If ICE is running for a media stream, and the offer for that media
   stream lacked the remote-candidates attribute, the rules for
   construction of the answer are identical to those for the offerer as
   described in Section 4.2.1.2.1.

4.2.2.2.2. **Existing Media Streams with ICE Completed and no remote-
             candidates**

   If ICE is Completed for a media stream, and the offer for that media
   stream lacked the remote-candidates attribute, the rules for
   construction of the answer are identical to those for the offerer as
   described in Section 4.2.1.2.2, except that the answerer MUST NOT
   include the a=remote-candidates attribute in the answer.

4.2.2.2.3. **Existing Media Streams and remote-candidates**

   A controlled agent will receive an offer with the a=remote-candidates
   attribute for a media stream when its peer has concluded ICE
   processing for that media stream. This attribute is present in the
   offer to deal with a race condition between the receipt of the offer,
   and the receipt of the Binding Response that tells the answerer the
   candidate that will be selected by ICE. See Appendix B for an
   explanation of this race condition. Consequently, processing of an
   offer with this attribute depends on the winner of the race.

   The agent forms a candidate pair for each component of the media
   stream by:

   o Setting the remote candidate equal to the offerer's default
      destination for that component (e.g., the contents of the "m=" and
      "c=" lines for RTP, and the a=rtcp attribute for RTCP)

   o Setting the local candidate equal to the transport address for
      that same component in the a=remote-candidates attribute in the
      offer.

   The agent then sees if each of these candidate pairs is present in
   the valid list. If a particular pair is not in the valid list, the
   check has "lost" the race. Call such a pair a "losing pair".

   The agent finds all the pairs in the check list whose remote
   candidates equal the remote candidate in the losing pair:

   o If none of the pairs are In-Progress, and at least one is Failed,
      it is most likely that a network failure, such as a network
      partition or serious packet loss, has occurred. The agent SHOULD
      generate an answer for this media stream as if the remote-
      candidates attribute had not been present, and then restart ICE
      for this stream.

   o If at least one of the pairs is In-Progress, the agent SHOULD wait
      for those checks to complete, and as each completes, redo the
      processing in this section until there are no losing pairs.

   Once there are no losing pairs, the agent can generate the answer.
   It MUST set the default destination for media to the candidates in
   the remote-candidates attribute from the offer (each of which will
   now be the local candidate of a candidate pair in the valid list).
   It MUST include a candidate attribute in the answer for each
   candidate in the remote-candidates attribute in the offer.

4.2.2.3. Procedures for Lite Implementations

   If the received offer contains the remote-candidates attribute for a
   media stream, the agent forms a candidate pair for each component of
   the media stream by:

   o Setting the remote candidate equal to the offerer's default
      destination for that component (e.g., the contents of the "m=" and
      "c=" lines for RTP, and the a=rtcp attribute for RTCP).

   o Setting the local candidate equal to the transport address for
      that same component in the a=remote-candidates attribute in the
      offer.

   It then places those candidates into the Valid list for the media
   stream. The state of ICE processing for that media stream is set to
   Completed.

Furthermore, if the agent believed it was controlling, but the offer contained the remote-candidates attribute, both agents believe they are controlling. In this case, both would have sent updated offers around the same time. However, the signaling protocol carrying the offer/answer exchanges will have resolved this glare condition, so that one agent is always the 'winner' by having its offer received before its peer has sent an offer. The winner takes the role of controlling, so that the loser (the answerer under consideration in this section) MUST change its role to controlled. Consequently, if the agent was going to send an updated offer since, based on the rules in section 8.2 of [ICE-BIS], it was controlling, it no longer needs to.

Besides the potential role change, change in the Valid list, and state changes, the construction of the answer is performed identically to the construction of an offer as described in Section 4.2.1.3.

4.2.3. Receiving the Answer for a Subsequent Offer

Some deployments of ICE include e.g. SDP-Modifying Signaling-only Back-to-Back User Agents (B2BUAs) [RFC7092] that modify the SDP body during the subsequent offer/answer exchange. With the B2BUA being ICE-unaware, a subsequent answer might be manipulated and might not include ICE candidates although the initial answer did.

An example of a situation where such an "unexpected" answer might be experienced appears when such a B2BUA introduces a media server during call hold using 3rd party call-control procedures. Omitting further details how this is done this could result in an answer being received at the holding UA that was constructed by the B2BUA. With the B2BUA being ICE-unaware, that answer would not include ICE candidates.

Receiving an answer without ICE attributes in this situation might be unexpected, but would not necessarily impair the user experience.

In addition to procedures for the expected answer, the following section advices on how to recover from the unexpected situation.

4.2.3.1. Procedures for All Implementations

When receiving an answer within an existing session for a subsequent offer as specified in Section 4.2.1.2.2, an agent MUST verify ICE support as specified in Section 4.1.3.1.

If ICE support is indicated in the SDP answer and the offer was a restart, the agent MUST perform ICE restart procedures as specified

in Section 4.2.4. If ICE support is no longer indicated in the SDP
answer, the agent MUST fall-back to [RFC3264] procedures and SHOULD
NOT drop the dialog just because of missing ICE support. If the
agent sends a new offer later on, it SHOULD perform an ICE restart as
specified in Section 4.2.1.1.1.

If ICE support is indicated in the SDP answer and ICE is running, the
agent MUST continue ICE procedures as specified in Section 4.2.4.1.4.
If ICE support is no longer indicated in the SDP answer, the agent
MUST abort the ongoing ICE processing and fall-back to [RFC3264]
procedures. The agent SHOULD NOT drop the dialog just because of
missing ICE support. If the agent sends a new offer later on, it
SHOULD perform an ICE restart as specified in Section 4.2.1.1.1.

If ICE support is indicated in the SDP answer and if ICE is completed
and the answer conforms to Section 4.2.2.2.3, the agent MUST remain
in the ICE Completed state. If ICE support is no longer indicated in
the SDP answer, the agent MUST fall-back to [RFC3264] procedures and
SHOULD NOT drop the dialog just because of this unexpected answer.
Once the agent sends a new offer later on it MUST perform an ICE
restart.

4.2.4. Updating the Check and Valid Lists

4.2.4.1. Procedures for Full Implementations

4.2.4.1.1. ICE Restarts

The agent MUST remember the highest-priority nominated pairs in the
Valid list for each component of the media stream, called the
previous selected pairs, prior to the restart. The agent will
continue to send media using these pairs, as described in
Section 7.1. Once these destinations are noted, the agent MUST flush
the valid and check lists, and then recompute the check list and its
states as described in section 6.1.2 of [ICE-BIS].

4.2.4.1.2. New Media Stream

If the offer/answer exchange added a new media stream, the agent MUST
create a new check list for it (and an empty Valid list to start of
course), as described in section 6.1.2 of [ICE-BIS].

4.2.4.1.3. Removed Media Stream

If the offer/answer exchange removed a media stream, or an answer
rejected an offered media stream, an agent MUST flush the Valid list
for that media stream. It MUST terminate any STUN transactions in

progress for that media stream. An agent MUST remove the check list
for that media stream and cancel any pending ordinary checks for it.

4.2.4.1.4. ICE Continuing for Existing Media Stream

The valid list is not affected by an updated offer/answer exchange
unless ICE is restarting.

If an agent is in the Running state for that media stream, the check
list is updated (the check list is irrelevant if the state is
completed). To do that, the agent recomputes the check list using
the procedures described in section 6.1.2 of [ICE-BIS]. If a pair on
the new check list was also on the previous check list, and its state
was Waiting, In-Progress, Succeeded, or Failed, its state is copied
over. Otherwise, its state is set to Frozen.

If none of the check lists are active (meaning that the pairs in each
check list are Frozen), the full-mode agent follows steps in
Section 6.1.2.6 of [ICE-BIS] to place appropriate candidates in the
Waiting state to further continue ICE processing.

4.2.4.2. Procedures for Lite Implementations

If ICE is restarting for a media stream, the agent MUST start a new
Valid list for that media stream. It MUST remember the pairs in the
previous Valid list for each component of the media stream, called
the previous selected pairs, and continue to send media there as
described in Section 7.1. The state of ICE processing for each media
stream MUST change to Running, and the state of ICE processing MUST
change to Running.

5. Grammar

This specification defines eight new SDP attributes -- the
"candidate", "remote-candidates", "ice-lite", "ice-mismatch", "ice-
ufrag", "ice-pwd", "ice-pacing", and "ice-options" attributes. This
section also provides non-normative examples of the attributes
defined.

The syntax for the attributes follow Augmented BNF as defined in
[RFC5234].

5.1. "candidate" Attribute

The candidate attribute is a media-level attribute only. It contains
a transport address for a candidate that can be used for connectivity
checks.

```
candidate-attribute   = "candidate" ":" foundation SP component-id SP
                          transport SP
                          priority SP
                          connection-address SP      ;from RFC 4566
                          port           ;port from RFC 4566
                          SP cand-type
                          [SP rel-addr]
                          [SP rel-port]
                          *(SP extension-att-name SP
                              extension-att-value)

foundation            = 1*32ice-char
component-id          = 1*5DIGIT
transport             = "UDP" / transport-extension
transport-extension   = token                ; from RFC 3261
priority              = 1*10DIGIT
cand-type             = "typ" SP candidate-types
candidate-types       = "host" / "srflx" / "prflx" / "relay" / token
rel-addr              = "raddr" SP connection-address
rel-port              = "rport" SP port
extension-att-name    = token
extension-att-value   = *VCHAR
ice-char              = ALPHA / DIGIT / "+" / "/"
```

This grammar encodes the primary information about a candidate: its
IP address, port and transport protocol, and its properties: the
foundation, component ID, priority, type, and related transport
address:

<connection-address>: is taken from RFC 4566 [RFC4566]. It is the
   IP address of the candidate. When parsing this field, an agent
   can differentiate an IPv4 address and an IPv6 address by presence
   of a colon in its value -- the presence of a colon indicates IPv6.
   An agent MUST ignore candidate lines that include candidates with
   IP address versions that are not supported or recognized. An IP
   address SHOULD be used, but an FQDN MAY be used in place of an IP
   address. In that case, when receiving an offer or answer
   containing an FQDN in an a=candidate attribute, the FQDN is looked
   up in the DNS first using an AAAA record (assuming the agent
   supports IPv6), and if no result is found or the agent only
   supports IPv4, using an A record. The rules from section 6 of
   [RFC6724] is followed by fixing the source address to be one from
   the candidate pair to be matched against destination addresses
   reported by FQDN, in cases where the DNS query returns more than
   one IP address.

<port>: is also taken from RFC 4566 [RFC4566]. It is the port of
   the candidate.

<transport>: indicates the transport protocol for the candidate.
   This specification only defines UDP. However, extensibility is
   provided to allow for future transport protocols to be used with
   ICE, such as the Datagram Congestion Control Protocol (DCCP)
   [RFC4340].

<foundation>: is composed of 1 to 32 <ice-char>s. It is an
   identifier that is equivalent for two candidates that are of the
   same type, share the same base, and come from the same STUN
   server. The foundation is used to optimize ICE performance in the
   Frozen algorithm as described in section 6.1.2 of [ICE-BIS]

<component-id>: is a positive integer between 1 and 256 that
   identifies the specific component of the media stream for which
   this is a candidate. It MUST start at 1 and MUST increment by 1
   for each component of a particular candidate. For media streams
   based on RTP, candidates for the actual RTP media MUST have a
   component ID of 1, and candidates for RTCP MUST have a component
   ID of 2. See section 14 in [ICE-BIS] for additional discussion on
   extending ICE to new media streams.

<priority>: is a positive integer between 1 and (2**31 - 1). The
   procedures for computing candidate's priority is described in
   section 5.1.2 of [ICE-BIS].

<cand-type>: encodes the type of candidate. This specification
   defines the values "host", "srflx", "prflx", and "relay" for host,
   server reflexive, peer reflexive, and relayed candidates,
   respectively. The set of candidate types is extensible for the
   future.

<rel-addr> and <rel-port>: convey transport addresses related to the
   candidate, useful for diagnostics and other purposes. <rel-addr>
   and <rel-port> MUST be present for server reflexive, peer
   reflexive, and relayed candidates. If a candidate is server or
   peer reflexive, <rel-addr> and <rel-port> are equal to the base
   for that server or peer reflexive candidate. If the candidate is
   relayed, <rel-addr> and <rel-port> are equal to the mapped address
   in the Allocate response that provided the client with that
   relayed candidate (see section Appendix B.3 of [ICE-BIS] for a
   discussion of its purpose). If the candidate is a host candidate,
   <rel-addr> and <rel-port> MUST be omitted.

   In some cases, e.g., for privacy reasons, an agent may not want to
   reveal the related address and port. In this case the address
   MUST be set to "0.0.0.0" (for IPv4 candidates) or "::" (for IPv6
   candidates) and the port to zero.

The candidate attribute can itself be extended. The grammar allows
for new name/value pairs to be added at the end of the attribute. An
implementation MUST ignore any name/value pairs it doesn't
understand.

Example: SDP line for UDP server reflexive candidate attribute for the RTP
component

a=candidate:2 1 UDP 1694498815 192.0.2.3 45664 typ
srflx raddr 10.0.1.1 rport 8998

5.2. "remote-candidates" Attribute

The syntax of the "remote-candidates" attribute is defined using
Augmented BNF as defined in [RFC5234]. The remote-candidates
attribute is a media-level attribute only.

```
remote-candidate-att = "remote-candidates:" remote-candidate
                       0*(SP remote-candidate)
remote-candidate = component-ID SP connection-address SP port
```

The attribute contains a connection-address and port for each
component. The ordering of components is irrelevant. However, a
value MUST be present for each component of a media stream. This
attribute MUST be included in an offer by a controlling agent for a
media stream that is Completed, and MUST NOT be included in any other
case.

Example: Remote candidates SDP lines for the RTP and RTCP components:

a=remote-candidates:1 192.0.2.3 45664
a=remote-candidates:2 192.0.2.3 45665

5.3. "ice-lite" and "ice-mismatch" Attributes

The syntax of the "ice-lite" and "ice-mismatch" attributes, both of
which are flags, is:

```
ice-lite            = "ice-lite"
ice-mismatch        = "ice-mismatch"
```

"ice-lite" is a session-level attribute only, and indicates that an
agent is a lite implementation. "ice-mismatch" is a media-level
attribute only, and when present in an answer, indicates that the
offer arrived with a default destination for a media component that
didn't have a corresponding candidate attribute.

5.4. "ice-ufrag" and "ice-pwd" Attributes

   The "ice-ufrag" and "ice-pwd" attributes convey the username fragment
   and password used by ICE for message integrity. Their syntax is:

   ice-pwd-att   = "ice-pwd:" password
   ice-ufrag-att   = "ice-ufrag:" ufrag
   password   = 22*256ice-char
   ufrag   = 4*256ice-char

   The "ice-pwd" and "ice-ufrag" attributes can appear at either the
   session-level or media-level. When present in both, the value in the
   media-level takes precedence. Thus, the value at the session-level
   is effectively a default that applies to all media streams, unless
   overridden by a media-level value. Whether present at the session or
   media-level, there MUST be an ice-pwd and ice-ufrag attribute for
   each media stream. If two media streams have identical ice-ufrag's,
   they MUST have identical ice-pwd's.

   The ice-ufrag and ice-pwd attributes MUST be chosen randomly at the
   beginning of a session. The ice-ufrag attribute MUST contain at
   least 24 bits of randomness, and the ice-pwd attribute MUST contain
   at least 128 bits of randomness. This means that the ice-ufrag
   attribute will be at least 4 characters long, and the ice-pwd at
   least 22 characters long, since the grammar for these attributes
   allows for 6 bits of information per character. The attributes MAY
   be longer than 4 and 22 characters, respectively, of course, up to
   256 characters. The upper limit allows for buffer sizing in
   implementations. Its large upper limit allows for increased amounts
   of randomness to be added over time. For compatibility with the 512
   character limitation for the STUN username attribute value and for
   bandwidth conservation considerations, the ice-ufrag attribute MUST
   NOT be longer than 32 characters when sending, but an implementation
   MUST accept up to 256 characters when receiving.

   Example shows sample ice-ufrag and ice-pwd SDP lines:

   a=ice-pwd:asd88fgpdd777uzjYhagZg
   a=ice-ufrag:8hhY

5.5. "ice-pacing" Attribute

   The "ice-pacing" attribute indicates the desired connectivity check
   pacing, in milliseconds, for this agent (see section 15 of
   [ICE-BIS]). The syntax is:

   ice-pacing-att   = "ice-pacing:" pacing-value
   pacing-value   = 1*10DIGIT

   Example shows ice-pacing value of 5 ms:

   a=ice-pacing:5

5.6. "ice-options" Attribute

   The "ice-options" attribute is a session- and media-level attribute.
   It contains a series of tokens that identify the options supported by
   the agent. Its grammar is:

   ice-options   = "ice-options:" ice-option-tag
                               0*(SP ice-option-tag)
   ice-option-tag   = 1*ice-char

   The existence of an ice-option in an offer indicates that a certain
   extension is supported by the agent and is willing to use it, if the
   peer agent also includes the same extension in the answer. There
   might be further extension specific negotiations needed between the
   agents that determine how the extensions gets used in a given
   session. The details of the negotiation procedures, if present, MUST
   be defined by the specification defining the extension.

   Example shows 'rtp+ecn' ice-option SDP line from <<RFC6679>>:

   a=ice-options:rtp+ecn

6. Keepalives

   All the ICE agents MUST follow the procedures defined in section 11
   of [ICE-BIS] for sending keepalives. The keepalives MUST be sent
   regardless of whether the media stream is currently inactive,
   sendonly, recvonly, or sendrecv, and regardless of the presence or
   value of the bandwidth attribute. An agent can determine that its
   peer supports ICE by the presence of a=candidate attributes for each
   media session.

7. Media Handling

7.1. Sending Media

   The selected pair for a component of a media stream might not equal
   the default pair for that same component from the most recent offer/
   answer exchange. When this happens, the selected pair is used for
   media, not the default pair. When ICE first completes, if the
   selected pairs aren't a match for the default pairs, the controlling
   agent sends an updated offer/answer exchange to remedy this
   disparity. However, until that updated offer arrives, there will not

be a match. Furthermore, in very unusual cases, the default
candidates in the updated offer/answer will not be a match.

7.1.1. Procedures for All Implementations

Section 12.1.3 of [ICE-BIS] defines procedures for sending media
common across Full and Lite implementations.

7.2. Receiving Media

See section 12.2 of [ICE-BIS] for procedures on receiving media.

8. SIP Considerations

Note that ICE is not intended for NAT traversal for SIP, which is
assumed to be provided via another mechanism [RFC5626].

When ICE is used with SIP, forking may result in a single offer
generating a multiplicity of answers. In that case, ICE proceeds
completely in parallel and independently for each answer, treating
the combination of its offer and each answer as an independent offer/
answer exchange, with its own set of local candidates, pairs, check
lists, states, and so on.

Once ICE processing has reached the Completed state for all peers for
media streams using those candidates, the agent SHOULD wait an
additional three seconds, and then it MAY cease responding to checks
or generating triggered checks on that candidate. It MAY free the
candidate at that time. Freeing of server reflexive candidates is
never explicit; it happens by lack of a keepalive. The three-second
delay handles cases when aggressive nomination is used, and the
selected pairs can quickly change after ICE has completed.

8.1. Latency Guidelines

ICE requires a series of STUN-based connectivity checks to take place
between endpoints. These checks start from the answerer on
generation of its answer, and start from the offerer when it receives
the answer. These checks can take time to complete, and as such, the
selection of messages to use with offers and answers can affect
perceived user latency. Two latency figures are of particular
interest. These are the post-pickup delay and the post-dial delay.
The post-pickup delay refers to the time between when a user "answers
the phone" and when any speech they utter can be delivered to the
caller. The post-dial delay refers to the time between when a user
enters the destination address for the user and ringback begins as a
consequence of having successfully started alerting the called user
agent.

Two cases can be considered -- one where the offer is present in the
initial INVITE and one where it is in a response.

8.1.1. **Offer in INVITE**

To reduce post-dial delays, it is RECOMMENDED that the caller begin
gathering candidates prior to actually sending its initial INVITE.
This can be started upon user interface cues that a call is pending,
such as activity on a keypad or the phone going off-hook.

On the receipt of the offer, the answerer SHOULD generate an answer
in a provisional response once it has compelted candidate gathering.
ICE requires that a provisional response with an SDP be transmitted
reliably. This can be done through the existing Provisional Response
Acknowledgment (PRACK) mechanism [RFC3262] or through an ICE specific
optimization, wherein, the agent retransmits the provisional response
with the exponential backoff timers described in [RFC3262]. Such
retransmissions MUST cease on receipt of a STUN Binding request for
one of the media streams signaled in that SDP or on transmission of
the answer in a 2xx response. If no Binding request is received
prior to the last retransmit, the agent does not consider the session
terminated. For the ICE lite peers, the agent MUST cease
retransmitting the 18x after sending it four times (ICE will actually
work even if the peer never receives the 18x; however, experience has
shown that sending it is important for middleboxes and firewall
traversal).

It should be noted that the ICE specific optimization is very
specific to provisional response carrying answers that start ICE
processing and it is not a general technique for 1xx reliability.
Also such an optimization SHOULD NOT be used if both agents support
PRACK.

Despite the fact that the provisional response will be delivered
reliably, the rules for when an agent can send an updated offer or
answer do not change from those specified in [RFC3262].
Specifically, if the INVITE contained an offer, the same answer
appears in all of the 1xx and in the 2xx response to the INVITE.
Only after that 2xx has been sent can an updated offer/answer
exchange occur.

Alternatively, an agent MAY delay sending an answer until the 200 OK;
however, this results in a poor user experience and is NOT
RECOMMENDED.

Once the answer has been sent, the agent SHOULD begin its
connectivity checks. Once candidate pairs for each component of a

media stream enter the valid list, the answerer can begin sending
media on that media stream.

However, prior to this point, any media that needs to be sent towards
the caller (such as SIP early media [RFC3960]) MUST NOT be
transmitted. For this reason, implementations SHOULD delay alerting
the called party until candidates for each component of each media
stream have entered the valid list. In the case of a PSTN gateway,
this would mean that the setup message into the PSTN is delayed until
this point. Doing this increases the post-dial delay, but has the
effect of eliminating 'ghost rings'. Ghost rings are cases where the
called party hears the phone ring, picks up, but hears nothing and
cannot be heard. This technique works without requiring support for,
or usage of, preconditions [RFC3312]. It also has the benefit of
guaranteeing that not a single packet of media will get clipped, so
that post-pickup delay is zero. If an agent chooses to delay local
alerting in this way, it SHOULD generate a 180 response once alerting
begins.

8.1.2. Offer in Response

In addition to uses where the offer is in an INVITE, and the answer
is in the provisional and/or 200 OK response, ICE works with cases
where the offer appears in the response. In such cases, which are
common in third party call control [RFC3725], ICE agents SHOULD
generate their offers in a reliable provisional response (which MUST
utilize [RFC3262]), and not alert the user on receipt of the INVITE.
The answer will arrive in a PRACK. This allows for ICE processing to
take place prior to alerting, so that there is no post-pickup delay,
at the expense of increased call setup delays. Once ICE completes,
the callee can alert the user and then generate a 200 OK when they
answer. The 200 OK would contain no SDP, since the offer/answer
exchange has completed.

Alternatively, agents MAY place the offer in a 2xx instead (in which
case the answer comes in the ACK). When this happens, the callee
will alert the user on receipt of the INVITE, and the ICE exchanges
will take place only after the user answers. This has the effect of
reducing call setup delay, but can cause substantial post-pickup
delays and media clipping.

8.2. SIP Option Tags and Media Feature Tags

[RFC5768] specifies a SIP option tag and media feature tag for usage
with ICE. ICE implementations using SIP SHOULD support this
specification, which uses a feature tag in registrations to
facilitate interoperability through signaling intermediaries.

8.3. Interactions with Forking

   ICE interacts very well with forking. Indeed, ICE fixes some of the
   problems associated with forking. Without ICE, when a call forks and
   the caller receives multiple incoming media streams, it cannot
   determine which media stream corresponds to which callee.

   With ICE, this problem is resolved. The connectivity checks which
   occur prior to transmission of media carry username fragments, which
   in turn are correlated to a specific callee. Subsequent media
   packets that arrive on the same candidate pair as the connectivity
   check will be associated with that same callee. Thus, the caller can
   perform this correlation as long as it has received an answer.

8.4. Interactions with Preconditions

   Quality of Service (QoS) preconditions, which are defined in
   [RFC3312] and [RFC4032], apply only to the transport addresses listed
   as the default targets for media in an offer/answer. If ICE changes
   the transport address where media is received, this change is
   reflected in an updated offer that changes the default destination
   for media to match ICE's selection. As such, it appears like any
   other re-INVITE would, and is fully treated in RFCs 3312 and 4032,
   which apply without regard to the fact that the destination for media
   is changing due to ICE negotiations occurring "in the background".

   Indeed, an agent SHOULD NOT indicate that QoS preconditions have been
   met until the checks have completed and selected the candidate pairs
   to be used for media.

   ICE also has (purposeful) interactions with connectivity
   preconditions [RFC5898]. Those interactions are described there.
   Note that the procedures described in Section 8.1 describe their own
   type of "preconditions", albeit with less functionality than those
   provided by the explicit preconditions in [RFC5898].

8.5. Interactions with Third Party Call Control

   ICE works with Flows I, III, and IV as described in [RFC3725]. Flow
   I works without the controller supporting or being aware of ICE.
   Flow IV will work as long as the controller passes along the ICE
   attributes without alteration. Flow II is fundamentally incompatible
   with ICE; each agent will believe itself to be the answerer and thus
   never generate a re-INVITE.

   The flows for continued operation, as described in Section 7 of
   [RFC3725], require additional behavior of ICE implementations to
   support. In particular, if an agent receives a mid-dialog re-INVITE

that contains no offer, it MUST restart ICE for each media stream and
go through the process of gathering new candidates. Furthermore,
that list of candidates SHOULD include the ones currently being used
for media.

9. Relationship with ANAT

[RFC4091], the Alternative Network Address Types (ANAT) Semantics for
the SDP grouping framework, and [RFC4092], its usage with SIP, define
a mechanism for indicating that an agent can support both IPv4 and
IPv6 for a media stream, and it does so by including two "m=" lines,
one for v4 and one for v6. This is similar to ICE, which allows for
an agent to indicate multiple transport addresses using the candidate
attribute. However, ANAT relies on static selection to pick between
choices, rather than a dynamic connectivity check used by ICE.

It is RECOMMENDED that ICE be used in realizing the dual-stack use-
cases in agents that support ICE.

10. Setting Ta and RTO for RTP Media Streams

During the gathering phase of ICE (section 5.1.1 [ICE-BIS]) and while
ICE is performing connectivity checks (section 7 [ICE-BIS]), an agent
sends STUN and TURN transactions. These transactions are paced at a
rate of one every Ta milliseconds, and utilize a specific RTO. See
Section 15 of [ICE-BIS] for details on how the values of Ta and RTO
are computed with a real-time media stream of known maximum bandwidth
to rate-control the ICE exchanges.

11. Security Considerations

11.1. Attacks on the Offer/Answer Exchanges

An attacker that can modify or disrupt the offer/answer exchanges
themselves can readily launch a variety of attacks with ICE. They
could direct media to a target of a DoS attack, they could insert
themselves into the media stream, and so on. These are similar to
the general security considerations for offer/answer exchanges, and
the security considerations in [RFC3264] apply. These require
techniques for message integrity and encryption for offers and
answers, which are satisfied by the TLS mechanism [RFC3261] when SIP
is used. As such, the usage of TLS with ICE is RECOMMENDED.

11.2. Insider Attacks

In addition to attacks where the attacker is a third party trying to
insert fake offers, answers, or STUN messages, there are several

attacks possible with ICE when the attacker is an authenticated and
valid participant in the ICE exchange.

11.2.1. The Voice Hammer Attack

The voice hammer attack is an amplification attack. In this attack,
the attacker initiates sessions to other agents, and maliciously
includes the IP address and port of a DoS target as the destination
for media traffic signaled in the SDP. This causes substantial
amplification; a single offer/answer exchange can create a continuing
flood of media packets, possibly at high rates (consider video
sources). This attack is not specific to ICE, but ICE can help
provide remediation.

Specifically, if ICE is used, the agent receiving the malicious SDP
will first perform connectivity checks to the target of media before
sending media there. If this target is a third-party host, the
checks will not succeed, and media is never sent.

Unfortunately, ICE doesn't help if it's not used, in which case an
attacker could simply send the offer without the ICE parameters.
However, in environments where the set of clients is known, and is
limited to ones that support ICE, the server can reject any offers or
answers that don't indicate ICE support.

User Agents that are not willing to receive non-ICE answers MUST
include an "ice" Option Tag in the Require Header Field in their
offer. Clients that rejects non-ICE offers SHOULD use a 421 response
code, together with an Option Tag "ice" in the Require Header Field
in the response.

11.2.2. Interactions with Application Layer Gateways and SIP

Application Layer Gateways (ALGs) are functions present in a Network
Address Translation (NAT) device that inspect the contents of packets
and modify them, in order to facilitate NAT traversal for application
protocols. Session Border Controllers (SBCs) are close cousins of
ALGs, but are less transparent since they actually exist as
application-layer SIP intermediaries. ICE has interactions with SBCs
and ALGs.

If an ALG is SIP aware but not ICE aware, ICE will work through it as
long as the ALG correctly modifies the SDP. A correct ALG
implementation behaves as follows:

o The ALG does not modify the "m=" and "c=" lines or the rtcp
   attribute if they contain external addresses.

   o If the "m=" and "c=" lines contain internal addresses, the
      modification depends on the state of the ALG:

      * If the ALG already has a binding established that maps an
         external port to an internal IP address and port matching the
         values in the "m=" and "c=" lines or rtcp attribute, the ALG
         uses that binding instead of creating a new one.

      * If the ALG does not already have a binding, it creates a new
         one and modifies the SDP, rewriting the "m=" and "c=" lines and
         rtcp attribute.

   Unfortunately, many ALGs are known to work poorly in these corner
   cases. ICE does not try to work around broken ALGs, as this is
   outside the scope of its functionality. ICE can help diagnose these
   conditions, which often show up as a mismatch between the set of
   candidates and the "m=" and "c=" lines and rtcp attributes. The ice-
   mismatch attribute is used for this purpose.

   ICE works best through ALGs when the signaling is run over TLS. This
   prevents the ALG from manipulating the SDP messages and interfering
   with ICE operation. Implementations that are expected to be deployed
   behind ALGs SHOULD provide for TLS transport of the SDP.

   If an SBC is SIP aware but not ICE aware, the result depends on the
   behavior of the SBC. If it is acting as a proper Back-to-Back User
   Agent (B2BUA), the SBC will remove any SDP attributes it doesn't
   understand, including the ICE attributes. Consequently, the call
   will appear to both endpoints as if the other side doesn't support
   ICE. This will result in ICE being disabled, and media flowing
   through the SBC, if the SBC has requested it. If, however, the SBC
   passes the ICE attributes without modification, yet modifies the
   default destination for media (contained in the "m=" and "c=" lines
   and rtcp attribute), this will be detected as an ICE mismatch, and
   ICE processing is aborted for the call. It is outside of the scope
   of ICE for it to act as a tool for "working around" SBCs. If one is
   present, ICE will not be used and the SBC techniques take precedence.

12. IANA Considerations

12.1. SDP Attributes

   The original ICE specification defined seven new SDP attributes per
   the procedures of Section 8.2.4 of [RFC4566]. The registration
   information is reproduced here.

[12.1.1](#). candidate Attribute

Contact Name: Jonathan Rosenberg, jdrosen@jdrosen.net.

Attribute Name: candidate

Long Form: candidate

Type of Attribute: media-level

Charset Considerations: The attribute is not subject to the charset
   attribute.

Purpose: This attribute is used with Interactive Connectivity
   Establishment (ICE), and provides one of many possible candidate
   addresses for communication. These addresses are validated with
   an end-to-end connectivity check using Session Traversal Utilities
   for NAT (STUN).

Appropriate Values: See [Section 5](#) of RFC XXXX.

[12.1.2](#). remote-candidates Attribute

Contact Name: Jonathan Rosenberg, jdrosen@jdrosen.net.

Attribute Name: remote-candidates

Long Form: remote-candidates

Type of Attribute: media-level

Charset Considerations: The attribute is not subject to the charset
   attribute.

Purpose: This attribute is used with Interactive Connectivity
   Establishment (ICE), and provides the identity of the remote
   candidates that the offerer wishes the answerer to use in its
   answer.

Appropriate Values: See [Section 5](#) of RFC XXXX.

[12.1.3](#). ice-lite Attribute

Contact Name: Jonathan Rosenberg, jdrosen@jdrosen.net.

Attribute Name: ice-lite

Long Form: ice-lite

Type of Attribute: session-level

Charset Considerations: The attribute is not subject to the charset
    attribute.

Purpose: This attribute is used with Interactive Connectivity
    Establishment (ICE), and indicates that an agent has the minimum
    functionality required to support ICE inter-operation with a peer
    that has a full implementation.

Appropriate Values: See Section 5 of RFC XXXX.

12.1.4. ice-mismatch Attribute

Contact Name: Jonathan Rosenberg, jdrosen@jdrosen.net.

Attribute Name: ice-mismatch

Long Form: ice-mismatch

Type of Attribute: session-level

Charset Considerations: The attribute is not subject to the charset
    attribute.

Purpose: This attribute is used with Interactive Connectivity
    Establishment (ICE), and indicates that an agent is ICE capable,
    but did not proceed with ICE due to a mismatch of candidates with
    the default destination for media signaled in the SDP.

Appropriate Values: See Section 5 of RFC XXXX.

12.1.5. ice-pwd Attribute

Contact Name: Jonathan Rosenberg, jdrosen@jdrosen.net.

Attribute Name: ice-pwd

Long Form: ice-pwd

Type of Attribute: session- or media-level

Charset Considerations: The attribute is not subject to the charset
    attribute.

Purpose: This attribute is used with Interactive Connectivity
    Establishment (ICE), and provides the password used to protect
    STUN connectivity checks.

Appropriate Values: See Section 5 of RFC XXXX.

12.1.6. ice-ufrag Attribute

Contact Name: Jonathan Rosenberg, jdrosen@jdrosen.net.

Attribute Name: ice-ufrag

Long Form: ice-ufrag

Type of Attribute: session- or media-level

Charset Considerations: The attribute is not subject to the charset
   attribute.

Purpose: This attribute is used with Interactive Connectivity
   Establishment (ICE), and provides the fragments used to construct
   the username in STUN connectivity checks.

Appropriate Values: See Section 5 of RFC XXXX.

12.1.7. ice-pacing Attribute

Contact Name: Jonathan Rosenberg, jdrosen@jdrosen.net.

Attribute Name: ice-pacing

Long Form: ice-pacing

Type of Attribute: session-level

Charset Considerations: The attribute is not subject to the charset
   attribute.

Purpose: This attribute is used with Interactive Connectivity
   Establishment (ICE) to indicate desired connectivity check pacing
   values.

Appropriate Values: See Section 5 of RFC XXXX.

12.1.8. ice-options Attribute

Contact Name: Jonathan Rosenberg, jdrosen@jdrosen.net.

Attribute Name: ice-options

Long Form: ice-options

Type of Attribute: session- or media-level

Charset Considerations: The attribute is not subject to the charset
    attribute.

Purpose: This attribute is used with Interactive Connectivity
    Establishment (ICE), and indicates the ICE options or extensions
    used by the agent.

Appropriate Values: See Section 5 of RFC XXXX.

12.2. Interactive Connectivity Establishment (ICE) Options Registry

IANA maintains a registry for ice-options identifiers under the
Specification Required policy as defined in "Guidelines for Writing
an IANA Considerations Section in RFCs" [RFC5226].

ICE options are of unlimited length according to the syntax in
Section 5.6; however, they are RECOMMENDED to be no longer than 20
characters. This is to reduce message sizes and allow for efficient
parsing.

In [RFC5245] ICE options could only be defined at the session level.
ICE options can now also be defined at the media level. This can be
used when aggregating between different ICE agents in the same
endpoint, but future options may require to be defined at the media-
level. To ensure compatibility with legacy implementation, the
media-level ICE options MUST be aggregated into a session-level ICE
option. Because aggregation rules depend on the specifics of each
option, all new ICE options MUST also define in their specification
how the media-level ICE option values are aggregated to generate the
value of the session-level ICE option.

[RFC6679] defines the "rtp+ecn" ICE option. The aggregation rule for
this ICE option is that if all aggregated media using ICE contain a
media-level "rtp+ecn" ICE option then an "rtp+ecn" ICE option MUST be
inserted at the session-level. If one of the media does not contain
the option, then it MUST NOT be inserted at the session-level.

Section 10 of [ICE-BIS] defines "ice2" ICE option. Since "ice2" is a
session level ICE option, no aggregation rules apply.

A registration request MUST include the following information:

o The ICE option identifier to be registered

o Name, Email, and Address of a contact person for the registration

   o Organization or individuals having the change control

   o Short description of the ICE extension to which the option relates

   o Reference(s) to the specification defining the ICE option and the
      related extensions

13. Acknowledgments

   A large part of the text in this document was taken from [RFC5245],
   authored by Jonathan Rosenberg.

   Some of the text in this document was taken from [RFC6336], authored
   by Magnus Westerlund and Colin Perkins.

   Thanks to Thomas Stach for the text in Section 4.2.3, Roman Shpount
   for suggesting RTCP candidate handling in Section 4.1.1.2 and Simon
   Perreault for advising on IPV6 address selection when candidate-
   address includes FQDN.

   Thanks to following experts for their reviews and constructive
   feedback: Christer Holmberg, Adam Roach and the MMUSIC WG.

14. References

14.1. Normative References

   [ICE-BIS] Keranen, A. and J. Rosenberg, "Interactive Connectivity
              Establishment (ICE): A Protocol for Network Address
              Translator (NAT) Traversal for Offer/Answer Protocols",
              draft-ietf-ice-rfc5245bis-00 (work in progress), March
              2015.

   [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
              Requirement Levels", BCP 14, RFC 2119,
              DOI 10.17487/RFC2119, March 1997,
              <http://www.rfc-editor.org/info/rfc2119>.

   [RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston,
              A., Peterson, J., Sparks, R., Handley, M., and E.
              Schooler, "SIP: Session Initiation Protocol", RFC 3261,
              DOI 10.17487/RFC3261, June 2002,
              <http://www.rfc-editor.org/info/rfc3261>.

   [RFC3262] Rosenberg, J. and H. Schulzrinne, "Reliability of
              Provisional Responses in Session Initiation Protocol
              (SIP)", RFC 3262, DOI 10.17487/RFC3262, June 2002,
              <http://www.rfc-editor.org/info/rfc3262>.

   [RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model
              with Session Description Protocol (SDP)", RFC 3264,
              DOI 10.17487/RFC3264, June 2002,
              <http://www.rfc-editor.org/info/rfc3264>.

   [RFC3312] Camarillo, G., Ed., Marshall, W., Ed., and J. Rosenberg,
              "Integration of Resource Management and Session Initiation
              Protocol (SIP)", RFC 3312, DOI 10.17487/RFC3312, October
              2002, <http://www.rfc-editor.org/info/rfc3312>.

   [RFC3556] Casner, S., "Session Description Protocol (SDP) Bandwidth
              Modifiers for RTP Control Protocol (RTCP) Bandwidth",
              RFC 3556, DOI 10.17487/RFC3556, July 2003,
              <http://www.rfc-editor.org/info/rfc3556>.

   [RFC3605] Huitema, C., "Real Time Control Protocol (RTCP) attribute
              in Session Description Protocol (SDP)", RFC 3605,
              DOI 10.17487/RFC3605, October 2003,
              <http://www.rfc-editor.org/info/rfc3605>.

   [RFC4032] Camarillo, G. and P. Kyzivat, "Update to the Session
              Initiation Protocol (SIP) Preconditions Framework",
              RFC 4032, DOI 10.17487/RFC4032, March 2005,
              <http://www.rfc-editor.org/info/rfc4032>.

   [RFC4091] Camarillo, G. and J. Rosenberg, "The Alternative Network
              Address Types (ANAT) Semantics for the Session Description
              Protocol (SDP) Grouping Framework", RFC 4091, June 2005,
              <http://www.rfc-editor.org/info/rfc4091>.

   [RFC4092] Camarillo, G. and J. Rosenberg, "Usage of the Session
              Description Protocol (SDP) Alternative Network Address
              Types (ANAT) Semantics in the Session Initiation Protocol
              (SIP)", RFC 4092, June 2005,
              <http://www.rfc-editor.org/info/rfc4092>.

   [RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session
              Description Protocol", RFC 4566, DOI 10.17487/RFC4566,
              July 2006, <http://www.rfc-editor.org/info/rfc4566>.

   [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an
              IANA Considerations Section in RFCs", BCP 26, RFC 5226,
              DOI 10.17487/RFC5226, May 2008,
              <http://www.rfc-editor.org/info/rfc5226>.

   [RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax
              Specifications: ABNF", STD 68, RFC 5234,
              DOI 10.17487/RFC5234, January 2008,
              <http://www.rfc-editor.org/info/rfc5234>.

   [RFC5245] Rosenberg, J., "Interactive Connectivity Establishment
              (ICE): A Protocol for Network Address Translator (NAT)
              Traversal for Offer/Answer Protocols", RFC 5245,
              DOI 10.17487/RFC5245, April 2010,
              <http://www.rfc-editor.org/info/rfc5245>.

   [RFC5389] Rosenberg, J., Mahy, R., Matthews, P., and D. Wing,
              "Session Traversal Utilities for NAT (STUN)", RFC 5389,
              DOI 10.17487/RFC5389, October 2008,
              <http://www.rfc-editor.org/info/rfc5389>.

   [RFC5768] Rosenberg, J., "Indicating Support for Interactive
              Connectivity Establishment (ICE) in the Session Initiation
              Protocol (SIP)", RFC 5768, DOI 10.17487/RFC5768, April
              2010, <http://www.rfc-editor.org/info/rfc5768>.

   [RFC6336] Westerlund, M. and C. Perkins, "IANA Registry for
              Interactive Connectivity Establishment (ICE) Options",
              RFC 6336, April 2010,
              <http://www.rfc-editor.org/info/rfc6336>.

   [RFC6679] Westerlund, M., Johansson, I., Perkins, C., O'Hanlon, P.,
              and K. Carlberg, "Explicit Congestion Notification (ECN)
              for RTP over UDP", RFC 6679, DOI 10.17487/RFC6679, August
              2012, <http://www.rfc-editor.org/info/rfc6679>.

   [RFC6724] Thaler, D., Draves, R., Matsumoto, A., and T. Chown,
              "Default Address Selection for Internet Protocol Version 6
              (IPv6)", RFC 6724, September 2012,
              <http://www.rfc-editor.org/info/rfc6724>.

   [RFC7092] Kaplan, H. and V. Pascual, "A Taxonomy of Session
              Initiation Protocol (SIP) Back-to-Back User Agents",
              RFC 7092, DOI 10.17487/RFC7092, December 2013,
              <http://www.rfc-editor.org/info/rfc7092>.

   [RFC7656] Lennox, J., Gross, K., Nandakumar, S., Salgueiro, G., and
              B. Burman, Ed., "A Taxonomy of Semantics and Mechanisms
              for Real-Time Transport Protocol (RTP) Sources", RFC 7656,
              DOI 10.17487/RFC7656, November 2015,
              <http://www.rfc-editor.org/info/rfc7656>.

14.2. Informative References

   [RFC3725] Rosenberg, J., Peterson, J., Schulzrinne, H., and G.
              Camarillo, "Best Current Practices for Third Party Call
              Control (3pcc) in the Session Initiation Protocol (SIP)",
              BCP 85, RFC 3725, DOI 10.17487/RFC3725, April 2004,
              <http://www.rfc-editor.org/info/rfc3725>.

   [RFC3960] Camarillo, G. and H. Schulzrinne, "Early Media and Ringing
              Tone Generation in the Session Initiation Protocol (SIP)",
              RFC 3960, DOI 10.17487/RFC3960, December 2004,
              <http://www.rfc-editor.org/info/rfc3960>.

   [RFC4340] Kohler, E., Handley, M., and S. Floyd, "Datagram
              Congestion Control Protocol (DCCP)", RFC 4340,
              DOI 10.17487/RFC4340, March 2006,
              <http://www.rfc-editor.org/info/rfc4340>.

   [RFC5626] Jennings, C., Ed., Mahy, R., Ed., and F. Audet, Ed.,
              "Managing Client-Initiated Connections in the Session
              Initiation Protocol (SIP)", RFC 5626,
              DOI 10.17487/RFC5626, October 2009,
              <http://www.rfc-editor.org/info/rfc5626>.

   [RFC5898] Andreasen, F., Camarillo, G., Oran, D., and D. Wing,
              "Connectivity Preconditions for Session Description
              Protocol (SDP) Media Streams", RFC 5898,
              DOI 10.17487/RFC5898, July 2010,
              <http://www.rfc-editor.org/info/rfc5898>.

Appendix A. Examples

   For the example shown in section 16 of [ICE-BIS] the resulting offer
   (message 5) encoded in SDP looks like:

```
   v=0
   o=jdoe 2890844526 2890842807 IN IP6 $L-PRIV-1.IP
   s=
   c=IN IP6 $NAT-PUB-1.IP
   t=0 0
   a=ice-pwd:asd88fgpdd777uzjYhagZg
   a=ice-ufrag:8hhY
   m=audio $NAT-PUB-1.PORT RTP/AVP 0
   b=RS:0
   b=RR:0
   a=rtpmap:0 PCMU/8000
   a=candidate:1 1 UDP 2130706431 $L-PRIV-1.IP $L-PRIV-1.PORT typ host
   a=candidate:2 1 UDP 1694498815 $NAT-PUB-1.IP $NAT-PUB-1.PORT typ
    srflx raddr $L-PRIV-1.IP rport $L-PRIV-1.PORT
```

   The offer, with the variables replaced with their values, will look
   like (lines folded for clarity):

```
v=0
o=jdoe 2890844526 2890842807 IN IP6 fe80::6676:baff:fe9c:ee4a
s=
c=IN IP6 2001:420:c0e0:1005::61
t=0 0
a=ice-pwd:asd88fgpdd777uzjYhagZg
a=ice-ufrag:8hhY
m=audio 45664 RTP/AVP 0
b=RS:0
b=RR:0
a=rtpmap:0 PCMU/8000
a=candidate:1 1 UDP 2130706431 fe80::6676:baff:fe9c:ee4a 8998 typ host
a=candidate:2 1 UDP 1694498815 2001:420:c0e0:1005::61 45664 typ srflx raddr
 fe80::6676:baff:fe9c:ee4a rport 8998
```

   The resulting answer looks like:

```
   v=0
   o=bob 2808844564 2808844564 IN IP4 $R-PUB-1.IP
   s=
   c=IN IP4 $R-PUB-1.IP
   t=0 0
   a=ice-pwd:YH75Fviy6338Vbrhrlp8Yh
   a=ice-ufrag:9uB6
   m=audio $R-PUB-1.PORT RTP/AVP 0
   b=RS:0
   b=RR:0
   a=rtpmap:0 PCMU/8000
   a=candidate:1 1 UDP 2130706431 $R-PUB-1.IP $R-PUB-1.PORT typ host
```

   With the variables filled in:

```
v=0
o=bob 2808844564 2808844564 IN IP4 192.0.2.1
s=
c=IN IP4 192.0.2.1
t=0 0
a=ice-pwd:YH75Fviy6338Vbrhrlp8Yh
a=ice-ufrag:9uB6
m=audio 3478 RTP/AVP 0
b=RS:0
b=RR:0
a=rtpmap:0 PCMU/8000
a=candidate:1 1 UDP 2130706431 192.0.2.1 3478 typ host
```

Appendix B. The remote-candidates Attribute

   The a=remote-candidates attribute exists to eliminate a race
   condition between the updated offer and the response to the STUN
   Binding request that moved a candidate into the Valid list. This
   race condition is shown in Figure 1. On receipt of message 4, agent
   L adds a candidate pair to the valid list. If there was only a
   single media stream with a single component, agent L could now send
   an updated offer. However, the check from agent R has not yet
   generated a response, and agent R receives the updated offer (message
   7) before getting the response (message 9). Thus, it does not yet
   know that this particular pair is valid. To eliminate this
   condition, the actual candidates at R that were selected by the
   offerer (the remote candidates) are included in the offer itself, and
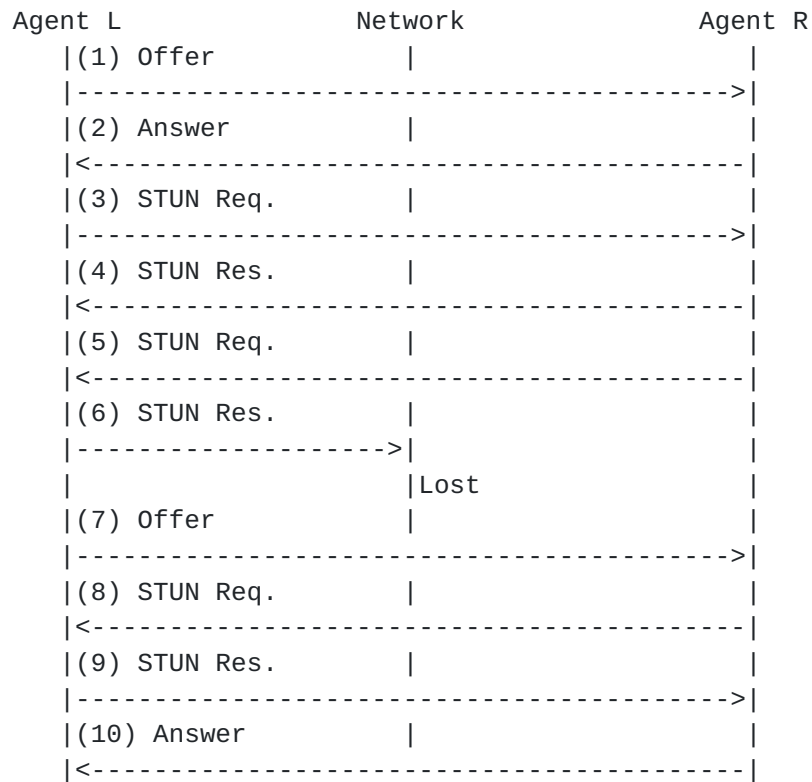   the answerer delays its answer until those pairs validate.

```
   Agent L                 Network              Agent R
      |(1) Offer            |                     |
      |------------------------------------------>|
      |(2) Answer           |                     |
      |<------------------------------------------|
      |(3) STUN Req.        |                     |
      |------------------------------------------>|
      |(4) STUN Res.        |                     |
      |<------------------------------------------|
      |(5) STUN Req.        |                     |
      |<------------------------------------------|
      |(6) STUN Res.        |                     |
      |-------------------->|                     |
      |                     |Lost                 |
      |(7) Offer            |                     |
      |------------------------------------------>|
      |(8) STUN Req.        |                     |
      |<------------------------------------------|
      |(9) STUN Res.        |                     |
      |------------------------------------------>|
      |(10) Answer          |                     |
      |<------------------------------------------|
```

                      Figure 1: Race Condition Flow

Appendix C. Why Is the Conflict Resolution Mechanism Needed?

   When ICE runs between two peers, one agent acts as controlled, and
   the other as controlling. Rules are defined as a function of
   implementation type and offerer/answerer to determine who is
   controlling and who is controlled. However, the specification
   mentions that, in some cases, both sides might believe they are
   controlling, or both sides might believe they are controlled. How
   can this happen?

   The condition when both agents believe they are controlled shows up
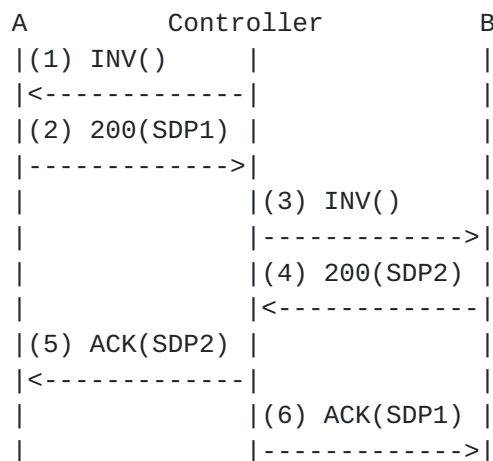   in third party call control cases. Consider the following flow:

```
        A           Controller          B
        |(1) INV()     |                |
        |<-------------|                |
        |(2) 200(SDP1) |                |
        |------------->|                |
        |              |(3) INV()       |
        |              |--------------->|
        |              |(4) 200(SDP2)   |
        |              |<---------------|
        |(5) ACK(SDP2) |                |
        |<-------------|                |
        |              |(6) ACK(SDP1)   |
        |              |--------------->|
```

              Figure 2: Role Conflict Flow

   This flow is a variation on flow III of RFC 3725 [RFC3725]. In fact,
   it works better than flow III since it produces fewer messages. In
   this flow, the controller sends an offerless INVITE to agent A, which
   responds with its offer, SDP1. The agent then sends an offerless
   INVITE to agent B, which it responds to with its offer, SDP2. The
   controller then uses the offer from each agent to generate the
   answers. When this flow is used, ICE will run between agents A and
   B, but both will believe they are in the controlling role. With the
   role conflict resolution procedures, this flow will function properly
   when ICE is used.

   At this time, there are no documented flows that can result in the
   case where both agents believe they are controlled. However, the
   conflict resolution procedures allow for this case, should a flow
   arise that would fit into this category.

Appendix D. Why Send an Updated Offer?

   Section 11.1 describes rules for sending media. Both agents can send
   media once ICE checks complete, without waiting for an updated offer.
   Indeed, the only purpose of the updated offer is to "correct" the SDP
   so that the default destination for media matches where media is
   being sent based on ICE procedures (which will be the highest-
   priority nominated candidate pair).

   This begs the question -- why is the updated offer/answer exchange
   needed at all? Indeed, in a pure offer/answer environment, it would
   not be. The offerer and answerer will agree on the candidates to use
   through ICE, and then can begin using them. As far as the agents
   themselves are concerned, the updated offer/answer provides no new
   information. However, in practice, numerous components along the
   signaling path look at the SDP information. These include entities

performing off-path QoS reservations, NAT traversal components such
as ALGs and Session Border Controllers (SBCs), and diagnostic tools
that passively monitor the network. For these tools to continue to
function without change, the core property of SDP -- that the
existing, pre-ICE definitions of the addresses used for media -- the
"m=" and "c=" lines and the rtcp attribute -- must be retained. For
this reason, an updated offer must be sent.

Authors' Addresses

   Marc Petit-Huguenin
   Impedance Mismatch

   Email: marc@petit-huguenin.org

   Ari Keranen
   Ericsson
   Jorvas 02420
   Finland

   Email: ari.keranen@ericsson.com

   Suhas Nandakumar
   Cisco Systems
   707 Tasman Dr
   Milpitas, CA 95035
   USA

   Email: snandaku@cisco.com