

MMUSIC
Internet-Draft
Intended status: Standards Track
Expires: January 23, 2017

M. Petit-Huguenin
Impedance Mismatch
A. Keranen
Ericsson
S. Nandakumar
Cisco Systems
July 22, 2016

Using Interactive Connectivity Establishment (ICE) with
Session Description Protocol (SDP) offer/answer and Session Initiation
Protocol (SIP)
draft-ietf-mmusic-ice-sip-sdp-10

Abstract

This document describes how Interactive Connectivity Establishment (ICE) is used with Session Description Protocol (SDP) offer/answer and Session Initiation Protocol (SIP).

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 January 23, 2017.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must

include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1. Introduction	3
2. Terminology	4
3. ICE Candidate Exchange and Offer/Answer Mapping	4
4. SDP Offer/Answer Procedures	4
4.1. Initial Offer/Answer Exchange	4
4.1.1. Sending the Initial Offer	4
4.1.2. Receiving the Initial Offer	7
4.1.3. Receipt of the Initial Answer	8
4.1.4. Performing Connectivity Checks	9
4.1.5. Concluding ICE	9
4.2. Subsequent Offer/Answer Exchanges	10
4.2.1. Generating the Offer	10
4.2.2. Receiving the Offer and Generating an Answer	13
4.2.3. Receiving the Answer for a Subsequent Offer	16
4.2.4. Updating the Check and Valid Lists	17
5. Grammar	19
5.1. "candidate" Attribute	19
5.2. "remote-candidates" Attribute	22
5.3. "ice-lite" and "ice-mismatch" Attributes	22
5.4. "ice-ufrag" and "ice-pwd" Attributes	22
5.5. "ice-pacing" Attribute	23
5.6. "ice-options" Attribute	23
6. Keepalives	24
7. Media Handling	24
7.1. Sending Media	24
7.1.1. Procedures for All Implementations	24
7.2. Receiving Media	24
8. Usage with SIP	24
8.1. Latency Guidelines	24
8.1.1. Offer in INVITE	25

8.1.2. Offer in Response	26
8.2. SIP Option Tags and Media Feature Tags	26
8.3. Interactions with Forking	27
8.4. Interactions with Preconditions	27
8.5. Interactions with Third Party Call Control	27
9. Relationship with ANAT	28
10. Setting Ta and RTO for RTP Media Streams	28
11. Security Considerations	28
11.1. Attacks on the Offer/Answer Exchanges	28
11.2. Insider Attacks	29
11.2.1. The Voice Hammer Attack	29
11.2.2. Interactions with Application Layer Gateways and SIP	29
12. IANA Considerations	30
12.1. SDP Attributes	30
12.1.1. candidate Attribute	30
12.1.2. remote-candidates Attribute	31
12.1.3. ice-lite Attribute	31
12.1.4. ice-mismatch Attribute	32
12.1.5. ice-pwd Attribute	32
12.1.6. ice-ufraq Attribute	33
12.1.7. ice-pacing Attribute	33
12.1.8. ice-options Attribute	33
12.2. Interactive Connectivity Establishment (ICE) Options Registry	34
13. Acknowledgments	35
14. References	35
14.1. Normative References	35
14.2. Informative References	37
Appendix A. Examples	38
Appendix B. The remote-candidates Attribute	40
Appendix C. Why Is the Conflict Resolution Mechanism Needed?	40
Appendix D. Why Send an Updated Offer?	41
Authors' Addresses	42

1. Introduction

This document describes how Interactive Connectivity Establishment (ICE) is used with Session Description Protocol (SDP) offer/answer [RFC3264] and Session Initiation Protocol (SIP). 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 and SIP.

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

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 [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, it is in the rtcp attribute when present, and when not present, the IP address is in the "c=" line and 1 plus the port is in the "m=" line.

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 the [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 4 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. If the default candidates are not selected by the ICE algorithm when communicating with an ICE-aware peer, an updated offer/answer will be required after ICE processing completes in order to "fix up" the SDP

so that the default destination for media matches the candidates selected by ICE. If ICE happens to select the default candidates, no updated offer/answer is required.

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.

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. It is **RECOMMENDED** that the default candidates are 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 media stream it wishes to use. The ordering of media 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 media stream, if there is one, first in the SDP.

There will be a candidate attribute for each candidate for a particular media 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 7 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 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-ufraq: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 9.1 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 5.1.1 of [ICE-BIS]), determines its role (section 5.1.2 of [ICE-BIS]), gathers candidates (section 4 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 5.1.3 and 5.1.4 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 4.2 of [ICE-BIS] for lite implementations.

4.1.2.3. Verifying ICE Support

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 rtp 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 8 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 6.1.2.3 of [ICE-BIS].

4.1.3. Receipt of the Initial Answer

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 pairs, check lists, states, and so on. The only case in which processing of one pair impacts another is freeing of candidates, discussed below in Section 4.1.5.2.

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 6 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. Verifying ICE Support

The logic at the offerer is identical to that of the answerer as described in section 5.1.1 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 6.1.3.1.1 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 6.1.2.3 of [ICE-BIS].

4.1.5. Concluding ICE

Once all of the media streams are completed, the controlling endpoint sends an updated offer if the transport destination in the `"m="` and `"c="` lines for the media stream (called the `DEFAULT CANDIDATES`) don't match ICE's selected candidates.

4.1.5.1. Procedures for Full Implementations

4.1.5.1.1. Updating states

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.

4.1.5.2. Freeing Candidates

4.1.5.2.1. Full Implementation Procedures

When ICE is used with SIP, and an offer is forked to multiple recipients, ICE proceeds in parallel and independently with each answerer, all using the same local candidates. 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.

4.2. Subsequent Offer/Answer Exchanges

Either agent MAY generate a subsequent offer at any time allowed by [RFC3264]. The rules in Section 4.1.5 will cause the controlling agent to send an updated offer at the conclusion of ICE processing when ICE has selected different candidate pairs from the default pairs. This section defines rules for construction of subsequent offers and answers.

Should a subsequent offer be rejected, 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 6.3 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.

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.

Additional behavior depends on the state ICE processing for that media stream.

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, 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. This "fixes" the default destination for media to equal the destination ICE has selected for media.

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 4.2 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 6.2 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 sections advice 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.

4.2.3.1.1. ICE Restarts

If ICE support is indicated in the SDP answer, 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 RFC 3264 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.

4.2.3.1.2. Existing Media Streams with ICE Running

If ICE support is indicated in the SDP answer, 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 RFC 3264 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.

4.2.3.1.3. Existing Media Streams with ICE Completed

If ICE support is indicated in the SDP answer and if 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 RFC 3264 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 5.1.3 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 5.1.3 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 5.1.3 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 sets the first pair in the check list for the first media stream to Waiting, and then sets the state of all other pairs in that check list for the same component ID and with the same foundation to Waiting as well.

Next, the agent goes through each check list, starting with the highest-priority pair. If a pair has a state of Succeeded, and it has a component ID of 1, then all Frozen pairs in the same check list

with the same foundation whose component IDs are not 1 have their state set to Waiting. If, for a particular check list, there are pairs for each component of that media stream in the Succeeded state, the agent moves the state of all Frozen pairs for the first component of all other media streams (and thus in different check lists) with the same foundation to Waiting.

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.

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.

The syntax of this attribute is defined using Augmented BNF as defined in [RFC5234]:

```

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, allowing for IPv4 addresses, IPv6 addresses, and fully qualified domain names (FQDNs). 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. If the DNS query returns more than one IP address, one is chosen, and then used for the remainder of ICE processing.

<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 TCP or 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.
- <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 10 in [ICE-BIS] for additional discussion on extending ICE to new media streams.
- <priority>**: is a positive integer between 1 and $(2^{*}31 - 1)$.
- <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>** is 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.

5.2. "remote-candidates" Attribute

The syntax of the "remote-candidates" attribute is defined using Augmented BNF as defined in RFC 5234 [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.

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 randomness 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.

5.5. "ice-pacing" Attribute

The "ice-pacing" attribute indicates the desired connectivity check pacing, in milliseconds, for this agent (see section 11 of [ICE-BIS]). The syntax is:

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

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 can indicate that a certain extension is supported by the agent and will be used or that the extension is used only if the other agent is willing to use it too. In order to avoid ambiguity, documents defining new options must indicate which case applies to the defined extensions.

6. Keepalives

The procedures defined in section 8 of [ICE-BIS] MUST be followed. 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

Note that the selected pair for a component of a media stream may 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 9.1.3 of [ICE-BIS] defines procedures for sending media common across Full and Lite implementations.

7.2. Receiving Media

See section 9.2 of [ICE-BIS] for procedures on receiving media.

8. Usage with SIP

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 ringing the phone of the called party.

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.

If an offer is received in an INVITE request, the answerer SHOULD begin to gather its candidates on receipt of the offer and then generate an answer in a provisional response once it has completed that process. 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 optimization that is specific to ICE. With this optimization, provisional responses containing an SDP answer that begins ICE processing for one or more media streams can be sent reliably without RFC 3262. To do this, the agent retransmits the provisional response with the exponential backoff timers described in RFC 3262. Retransmits MUST cease on receipt of a STUN Binding request for one of the media streams signaled in that SDP (because receipt of a Binding request indicates the offerer has received the answer) or on transmission of the answer in a 2xx response. If the peer agent is lite, there will never be a STUN Binding request. In such a case, 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). If no Binding request is received prior to the last retransmit, the agent does not consider the session terminated. 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 RFC 3262. 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. This optimization SHOULD NOT be used if both agents support PRACK. Note that the optimization is very specific to provisional response carrying answers that start ICE processing; it is not a general technique for 1xx reliability.

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], since it's a localized decision. 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 RFC 3262), 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 RFC 3312 [RFC3312] and RFC 4032 [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 RFC 3725, 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

RFC 4091 [RFC4091], the Alternative Network Address Types (ANAT) Semantics for the SDP grouping framework, and RFC 4092 [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.

This specification deprecates RFC 4091 and RFC 4092. Instead, agents wishing to support dual-stack will utilize ICE.

10. Setting Ta and RTO for RTP Media Streams

During the gathering phase of ICE (section 4.1.1 [ICE-BIS]) and while ICE is performing connectivity checks (section 6 [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 11 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 RFC 3264 [RFC3264] apply. These require techniques for message integrity and encryption for offers and answers, which are satisfied by the SIPS mechanism [RFC3261] when SIP is used. As such, the usage of SIPS 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 its 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.

11.2.2. Interactions with Application Layer Gateways and SIP

Application Layer Gateways (ALGs) are functions present in a 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 rtp 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

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.

12.1.1.6. 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.

12.1.1.7. 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.

12.1.1.8. 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 RFC 5245 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 "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 7 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 RFC 5245, authored by Jonathan Rosenberg.

Some of the text in this document was taken from RFC 6336, authored by Magnus Westerlund and Colin Perkins.

Thanks to Thomas Stach for the text in Section 4.2.3 and Roman Shpount for suggesting RTCP candidate handling in Section 4.1.1.2

Thanks to following experts for their review and constructive feedback: Christer Holmberg.

14. References

14.1. Normative References

- [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>>.

- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550, July 2003, <<http://www.rfc-editor.org/info/rfc3550>>.
- [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, DOI 10.17487/RFC4091, 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, DOI 10.17487/RFC4092, 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>>.
- [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>>.
- [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>>.
- [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.

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 12 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.

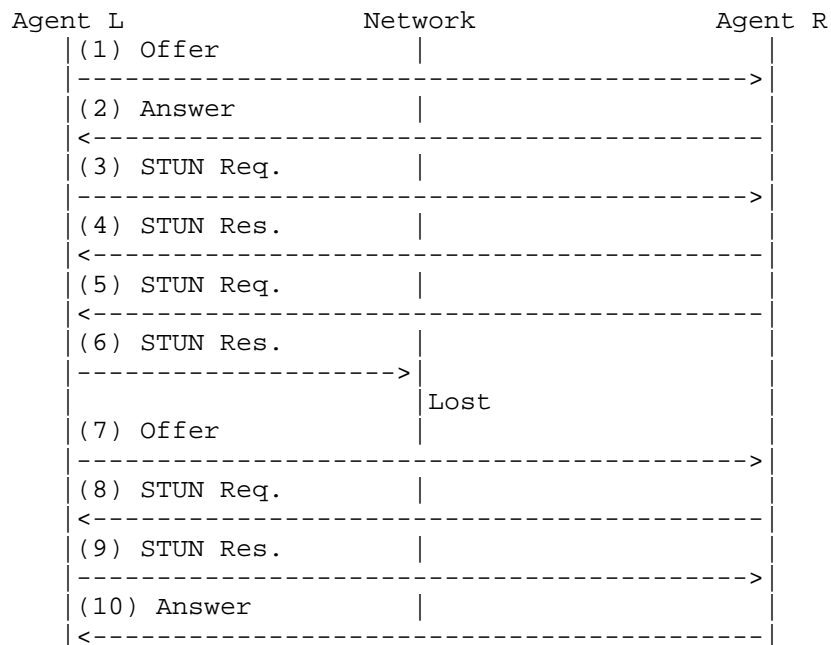


Figure 1: Race Condition Flow

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:

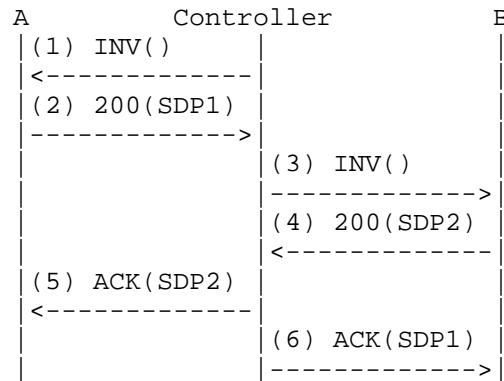


Figure 2: Role Conflict Flow

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 95035
USA

Email: snandaku@cisco.com

MMUSIC
Internet-Draft
Obsoletes: 5245 (if approved)
Intended status: Standards Track
Expires: February 14, 2020

M. Petit-Huguenin
Impedance Mismatch
S. Nandakumar
Cisco Systems
C. Holmberg
A. Keranen
Ericsson
R. Shpount
TurboBridge
August 13, 2019

Session Description Protocol (SDP) Offer/Answer procedures for
Interactive Connectivity Establishment (ICE)
draft-ietf-mmusic-ice-sip-sdp-39

Abstract

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

This document obsoletes RFC 5245.

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 <https://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 February 14, 2020.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents

(<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1.	Introduction	3
2.	Conventions	4
3.	Terminology	4
4.	SDP Offer/Answer Procedures	4
4.1.	Introduction	4
4.2.	Generic Procedures	5
4.2.1.	Encoding	5
4.2.2.	RTP/RTCP Considerations	6
4.2.3.	Determining Role	6
4.2.4.	STUN Considerations	6
4.2.5.	Verifying ICE Support Procedures	7
4.2.6.	SDP Example	8
4.3.	Initial Offer/Answer Exchange	8
4.3.1.	Sending the Initial Offer	8
4.3.2.	Sending the Initial Answer	9
4.3.3.	Receiving the Initial Answer	10
4.3.4.	Concluding ICE	10
4.4.	Subsequent Offer/Answer Exchanges	11
4.4.1.	Sending Subsequent Offer	11
4.4.2.	Sending Subsequent Answer	14
4.4.3.	Receiving Answer for a Subsequent Offer	16
5.	Grammar	17
5.1.	"candidate" Attribute	18
5.2.	"remote-candidates" Attribute	20
5.3.	"ice-lite" and "ice-mismatch" Attributes	21
5.4.	"ice-ufrag" and "ice-pwd" Attributes	21

5.5.	"ice-pacing" Attribute	22
5.6.	"ice-options" Attribute	22
6.	Keepalives	23
7.	SIP Considerations	23
7.1.	Latency Guidelines	23
7.1.1.	Offer in INVITE	24
7.1.2.	Offer in Response	25
7.2.	SIP Option Tags and Media Feature Tags	25
7.3.	Interactions with Forking	25
7.4.	Interactions with Preconditions	25
7.5.	Interactions with Third Party Call Control	26
8.	Interactions with Application Layer Gateways and SIP	26
9.	Security Considerations	27
9.1.	IP Address Privacy	28
9.2.	Attacks on the Offer/Answer Exchanges	28
9.3.	The Voice Hammer Attack	28
10.	IANA Considerations	29
10.1.	SDP Attributes	29
10.1.1.	candidate Attribute	29
10.1.2.	remote-candidates Attribute	29
10.1.3.	ice-lite Attribute	30
10.1.4.	ice-mismatch Attribute	30
10.1.5.	ice-pwd Attribute	31
10.1.6.	ice-ufrag Attribute	31
10.1.7.	ice-options Attribute	32
10.1.8.	ice-pacing Attribute	32
10.2.	Interactive Connectivity Establishment (ICE) Options Registry	33
10.3.	Candidate Attribute Extension Subregistry Establishment	33
11.	Acknowledgments	34
12.	Changes from RFC 5245	34
13.	References	34
13.1.	Normative References	34
13.2.	Informative References	36
Appendix A.	Examples	37
Appendix B.	The remote-candidates Attribute	39
Appendix C.	Why Is the Conflict Resolution Mechanism Needed?	40
Appendix D.	Why Send an Updated Offer?	41
Appendix E.	Contributors	42
Authors' Addresses		42

1. Introduction

This document describes how Interactive Connectivity Establishment (ICE) is used with Session Description Protocol (SDP) offer/answer [RFC3264]. The ICE specification [RFC8445] 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.

This document obsoletes RFC 5245.

NOTE: Previously both the common ICE procedures, and the SDP offer/answer specific details, were described in [RFC5245]. [RFC8445] obsoleted [RFC5245], and the SDP offer/answer specific details were removed from the document. Section 12 describes the changes to the SDP offer/answer specific details specified in this document.

2. Conventions

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 BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Terminology

Readers should be familiar with the terminology defined in [RFC3264], in [RFC8445] and the following:

Default Destination/Candidate: The default destination for a component of a data 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 connection address is in the "c=" line of the SDP, and the port and transport protocol are 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 the same as the address of the RTP component, and its port is one greater than the port of the RTP component.

4. SDP Offer/Answer Procedures

4.1. Introduction

[RFC8445] 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 terms "offerer" and "answerer" correspond to the initiator and responder roles from [RFC8445] respectively.

Once the initiating agent has gathered, pruned, and prioritized its set of candidates [RFC8445], the candidate exchange with the peer agent begins.

4.2. Generic Procedures

4.2.1. Encoding

Section 5 provides detailed rules for constructing various SDP attributes defined in this specification.

4.2.1.1. Data Streams

Each data stream [RFC8445] is represented by an SDP media description ("m=" section).

4.2.1.2. Candidates

Within an "m=" section, each candidate (including the default candidate) associated with the data stream is represented by an SDP candidate attribute.

Prior to nomination, the "c=" line associated with an "m=" section contains the connection address of the default candidate, while the "m=" line contains the port and transport protocol of the default candidate for that "m=" section.

After nomination, the "c=" line for a given "m=" section contains the connection address of the nominated candidate (the local candidate of the nominated candidate pair) and the "m=" line contains the port and transport protocol corresponding to the nominated candidate for that "m=" section.

4.2.1.3. Username and Password

The ICE username is represented by an SDP ice-ufrag attribute and the ICE password is represented by an SDP ice-pwd attribute.

4.2.1.4. Lite Implementations

An ICE lite implementation [RFC8445] MUST include an SDP ice-lite attribute. A full implementation MUST NOT include that attribute.

4.2.1.5. ICE Extensions

An agent uses the SDP ice-options attribute to indicate support of ICE extensions.

An agent compliant to this specification MUST include an SDP ice-options attribute with an "ice2" attribute value [RFC8445]. If an agent receives an SDP offer or answer that indicates ICE support, but that does not contain an SDP ice-options attribute with an "ice2"

attribute value, the agent can assume that the peer is compliant to [RFC5245].

4.2.1.6. Inactive and Disabled Data Streams

If an "m=" section is marked as inactive [RFC4566], or has a bandwidth value of zero [RFC4566], the agent MUST still include ICE-related SDP attributes.

If the port value associated with an "m=" section is set to zero (implying a disabled stream) as defined in section 8.2 of [RFC3264], the agent SHOULD NOT include ICE-related SDP candidate attributes in that "m=" section, unless an SDP extension specifying otherwise is used.

4.2.2. RTP/RTCP Considerations

If an agent utilizes both RTP and RTCP, and separate ports are used for RTP and RTCP, the agent MUST include SDP candidate attributes for both the RTP and RTCP components.

The agent includes an SDP rtcp attribute following the procedures in [RFC3605]. Hence, in the cases where the RTCP port value is one higher than the RTP port value and the RTCP component address the same as the address of the RTP component, the SDP rtcp attribute might be omitted.

NOTE: [RFC5245] required that an agent always includes the SDP rtcp attribute, even if the RTCP port value was one higher than the RTP port value. This specification aligns the rtcp attribute procedures with [RFC3605].

If the agent does not utilize RTCP, it indicates that by including b=RS:0 and b=RR:0 SDP attributes, as described in [RFC3556].

4.2.3. Determining Role

The offerer acts as the Initiating agent. The answerer acts as the Responding agent. The ICE roles (controlling and controlled) are determined using the procedures in [RFC8445].

4.2.4. STUN Considerations

Once an agent has provided its local candidates to its peer in an SDP offer or answer, the agent MUST be prepared to receive STUN connectivity check Binding requests on those candidates.

4.2.5. Verifying ICE Support Procedures

An ICE agent is considered to indicate support of ICE by including at least the SDP `ice-pwd` and `ice-ufrag` attributes in an offer or answer. An ICE agent compliant with this specification **MUST** also include an SDP `ice-options` attribute with an `"ice2"` attribute value.

The agents will proceed with the ICE procedures defined in [RFC8445] and this specification if, for each data stream in the SDP it received, the default destination for each component of that data stream appears in a candidate attribute. For example, in the case of RTP, the connection address, port, and transport protocol in the `"c="` and `"m="` lines, respectively, appear in a candidate attribute and the value in the `rtcp` attribute appears in a candidate attribute.

This specification provides no guidance on how an agent should proceed in the cases where the above condition is not met with the few exceptions noted below:

1. The presence of certain application layer gateways might modify the transport address information as described in Section 8. The behavior of the responding agent in such a situation is implementation dependent. Informally, the responding agent might consider the mismatched transport address information as a plausible new candidate learnt from the peer and continue its ICE processing with that transport address included. Alternatively, the responding agent **MAY** include an `"a=ice-mismatch"` attribute in its answer for such data streams. If an agent chooses to include an `"a=ice-mismatch"` attribute in its answer for a data stream, then it **MUST** also omit `"a=candidate"` attributes, **MUST** terminate the usage of ICE procedures and [RFC3264] procedures **MUST** be used instead for this data stream.
2. The transport address from the peer for the default destination is set to IPv4/IPv6 address values `"0.0.0.0"/":::"` and port value of `"9"`. This **MUST NOT** be considered as a ICE failure by the peer agent and the ICE processing **MUST** continue as usual.
3. In some cases, the controlling/initiator agent may receive the SDP answer that may omit `"a=candidate"` attributes for the data stream, and instead include a media level `"a=ice-mismatch"` attribute. This signals to the offerer that the answerer supports ICE, but that ICE processing was not used for this data stream. In this case, ICE processing **MUST** be terminated for this data stream and [RFC3264] procedures **MUST** be followed instead.
4. The transport address from the peer for the default destination is an FQDN. Regardless of the procedures used to resolve FQDN or

the resolution result, this MUST NOT be considered as a ICE failure by the peer agent and the ICE processing MUST continue as usual.

4.2.6. SDP Example

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

```
v=0
o=jdoe 2890844526 2890842807 IN IP4 203.0.113.141
s=
c=IN IP4 192.0.2.3
t=0 0
a=ice-options:ice2
a=ice-pacing:50
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 203.0.113.141 8998 typ host
a=candidate:2 1 UDP 1694498815 192.0.2.3 45664 typ srflx raddr
  203.0.113.141 rport 8998
```

4.3. Initial Offer/Answer Exchange

4.3.1. Sending the Initial Offer

When an offerer generates the initial offer, in each "m=" section it MUST include SDP candidate attributes for each available candidate associated with the "m=" section. In addition, the offerer MUST include an SDP ice-ufrag attribute, an SDP ice-pwd attribute and an SDP ice-options attribute with an "ice2" attribute value in the offer. If the offerer is a full ICE implementation, it SHOULD include an ice-pacing attribute in the offer (if not included, the default value will apply). A lite ICE implementation MUST NOT include the ice-pacing attribute in the offer (as it will not perform connectivity checks).

It is valid for an offer "m=" line to include no SDP candidate attributes and with default destination set to the IP address values "0.0.0.0"/":::" and port value of "9". This implies that the offering agent is only going to use peer reflexive candidates or that additional candidates would be provided in subsequent signaling messages.

Note: Within the scope of this document, "Initial Offer" refers to the first SDP offer that is sent in order to negotiate usage of ICE. It might, or might not, be the initial SDP offer of the SDP session.

Note: The procedures in this document only consider "m=" sections associated with data streams where ICE is used.

4.3.2. Sending the Initial Answer

When an answerer receives an initial offer that indicates that the offerer supports ICE, and if the answerer accepts the offer and the usage of ICE, in each "m=" section within the answer, it MUST include SDP candidate attributes for each available candidate associated with the "m=" section. In addition, the answerer MUST include an SDP ice-frag attribute, an SDP ice-pwd attribute and an SDP ice-options attribute with an "ice2" attribute value in the answer. If the answerer is a full ICE implementation, it SHOULD include an ice-pacing attribute in the answer (if not included, the default value will apply). A lite ICE implementation MUST NOT include the ice-pacing attribute in the answer (as it will not perform connectivity checks).

In each "m=" line, the answerer MUST use the same transport protocol as was used in the offer "m=" line. If none of the candidates in the "m=" line in the answer use the same transport protocol as indicated in the offer "m=" line, then, in order to avoid ICE mismatch, the default destination MUST be set to IP address values "0.0.0.0"/":::" and port value of "9".

It is also valid for an answer "m=" line to include no SDP candidate attributes and with default destination set to the IP address values "0.0.0.0"/":::" and port value of "9". This implies that the answering agent is only going to use peer reflexive candidates or that additional candidates would be provided in subsequent signaling messages.

Once the answerer has sent the answer, it can start performing connectivity checks towards the peer candidates that were provided in the offer.

If the offer does not indicate support of ICE Section 4.2.5, the answerer MUST NOT accept the usage of ICE. If the answerer still accepts the offer, the answerer MUST NOT include any ICE-related SDP attributes in the answer. Instead the answerer will generate the answer according to normal offer/answer procedures [RFC3264].

If the answerer detects a possibility of an ICE mismatch, procedures described in Section 4.2.5 are followed.

4.3.3. Receiving the Initial Answer

When an offerer receives an initial answer that indicates that the answerer supports ICE, it can start performing connectivity checks towards the peer candidates that were provided in the answer.

If the answer does not indicate that the answerer supports ICE, or if the answerer included "a=ice-mismatch" attributes for all the active data streams in the answer, the offerer MUST terminate the usage of ICE for the entire session and [RFC3264] procedures MUST be followed instead.

On the other hand, if the answer indicates support for ICE but includes "a=ice-mismatch" in certain active data streams, then the offerer MUST terminate the usage of ICE procedures and [RFC3264] procedures MUST be used instead for only these data streams. Also, ICE procedures MUST be used for data streams where an "a=ice-mismatch" attribute was not included.

If the offerer detects an ICE mismatch for one or more data streams in the answer, as described in Section 4.2.5, the offerer MUST terminate the usage of ICE for the entire session. The subsequent actions taken by the offerer are implementation dependent and are out of the scope of this specification.

4.3.4. Concluding ICE

Once the agent has successfully nominated a pair [RFC8445], the state of the checklist associated with the pair is set to Completed. Once the state of each checklist is set to either Completed or Failed, for each Completed checklist the agent checks whether the nominated pair matches the default candidate pair. If there are one or more pairs that do not match, and the peer did not indicate support for the 'ice2' ice-option, the controlling agent MUST generate a subsequent offer, in which the connection address, port and transport protocol in the "c=" and "m=" lines associated with each data stream match the corresponding local information of the nominated pair for that data stream (Section 4.4.1.2.2). If the peer did indicate support for the 'ice2' ice-option, the controlling agent does not immediately need to generate an updated offer in order to align a connection address, port and protocol with a nominated pair. However, later in the session, whenever the controlling agent does send a subsequent offer, it MUST do the alignment as described above.

If there are one or more checklists with the state set to Failed, the controlling agent MUST generate a subsequent offer in order to remove the associated data streams by setting the port value of the data streams to zero (Section 4.4.1.1.2), even if the peer did indicate support for the 'ice2' ice-option. If needed, such offer is used to align the connection address, port and transport protocol, as described above.

As described in [RFC8445], once the controlling agent has nominated a candidate pair for a checklist, the agent MUST NOT nominate another pair for that checklist during the lifetime of the ICE session (i.e. until ICE is restarted).

[draft-ietf-ice-pac] provides a mechanism for allowing the ICE process to run long enough in order to find working candidate pairs, by waiting for potential peer-reflexive candidates, even though no candidate pairs were received from the peer or all current candidate pairs associated with a checklist have either failed or been discarded. It is OPTIONAL for an ICE agent to support the mechanism.

4.4. 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.

4.4.1. Sending Subsequent Offer

4.4.1.1. Procedures for All Implementations

4.4.1.1.1. ICE Restart

An agent MAY restart ICE processing for an existing data stream [RFC8445].

The rules governing the ICE restart imply that setting the connection address in the "c=" line to "0.0.0.0" (for IPv4)/ ":::" (for IPv6) 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 data stream in an offer. However, 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 MUST NOT be considered as ICE restart.

An agent sets the rest of the ICE-related fields in the SDP for this data stream as it would in an initial offer of this data stream (Section 4.2.1). Consequently, the set of candidates MAY include some, none, or all of the previous candidates for that data stream and MAY include a totally new set of candidates. The agent MAY modify the attribute values of the SDP ice-options and SDP ice-pacing attributes, and it MAY change its role using the SDP ice-lite attribute. The agent MUST NOT modify the SDP ice-options, ice-pacing and ice-lite attributes in a subsequent offer unless the offer is sent in order to request an ICE restart.

4.4.1.1.2. Removing a Data Stream

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

4.4.1.1.3. Adding a Data Stream

If an agent wishes to add a new data stream, it sets the fields in the SDP for this data stream as if this were an initial offer for that data stream (Section 4.2.1). This will cause ICE processing to begin for this data stream.

4.4.1.2. Procedures for Full Implementations

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

4.4.1.2.1. Before Nomination

When an offerer sends a subsequent offer; in each "m=" section for which a candidate pair has not yet been nominated, the offer MUST include the same set of ICE-related information that the offerer included in the previous offer or answer. The agent MAY include additional candidates it did not offer previously, 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.

4.4.1.2.2. After Nomination

Once a candidate pair has been nominated for a data stream, the connection address, port and transport protocol in each "c=" and "m=" line associated with that data stream MUST match the data associated with the nominated pair for that data stream. In addition, the offerer only includes SDP candidates (one per component) representing the local candidates of the nominated candidate pair. The offerer MUST NOT include any other SDP candidate attributes in the subsequent offer.

In addition, if the agent is controlling, it MUST include the "a=remote-candidates" attribute for each data stream whose checklist is in the Completed state. The attribute contains the remote candidates corresponding to the nominated pair in the valid list for each component of that data 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.

4.4.1.3. Procedures for Lite Implementations

If the ICE state is Running, a lite implementation MUST include all of its candidates for each component of each data stream in "a=candidate" attributes in any subsequent offer. The candidates are formed identically to the procedures for initial offers.

A lite implementation MUST NOT add additional host candidates in a subsequent offer, and MUST NOT modify the username fragments and passwords. If an agent needs to offer additional candidates, or modify the username fragments and passwords, it MUST request an ICE restart (Section 4.4.1.1.1) for that data stream.

If ICE has completed for a data stream and if the agent is controlled, the default destination for that data 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 data stream. Additionally, the agent MUST include a candidate attribute for each default destination.

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

4.4.2. Sending Subsequent Answer

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

A controlled agent will receive an offer with the "a=remote-candidates" attribute for a data stream when its peer has concluded ICE processing for that data 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 data stream by:

- o Setting the remote candidate equal to the offerer's default destination for that component (i.e. 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 checklist 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 data 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.

4.4.2.1. ICE Restart

If the offerer in a subsequent offer requested an ICE restart (Section 4.4.1.1.1) for a data stream, and if the answerer accepts the offer, the answerer follows the procedures for generating an initial answer.

For a given data stream, the answerer MAY include the same candidates that were used in the previous ICE session, but it MUST change the SDP ice-pwd and ice-ufrag attribute values.

The answerer MAY modify the attribute values of the SDP ice-options and SDP ice-pacing attributes, and it MAY change its role using the SDP ice-lite attribute. The answerer MUST NOT modify the SDP ice-options, ice-pacing and ice-lite attributes in a subsequent answer unless the answer is sent for an offer that was used to request an ICE restart (Section 4.4.1.1.1). If any of the SDP attributes have been modified in a subsequent offer that is not used to request an ICE restart, the answerer MUST reject the offer.

4.4.2.2. Lite Implementation specific procedures

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

- o Setting the remote candidate equal to the offerer's default destination for that component (i.e., 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 state of the checklist associated with that data stream is set to Completed.

Furthermore, if the agent believed it was controlling, but the offer contained the "a=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 [RFC8445], 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.

4.4.3. Receiving Answer for a Subsequent Offer

4.4.3.1. Procedures for Full Implementations

There may be certain situations where the offerer receives an SDP answer that lacks ICE candidates although the initial answer included them. One example of such an "unexpected" answer might be happen when an ICE-unaware Back-to-Back User Agent (B2BUA) introduces a media server during call hold using 3rd party call-control procedures [RFC3725]. 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.

When the offerer receives an answer indicating support for ICE, the offer performs one of the following actions:

- o If the offer was a restart, the agent MUST perform ICE restart procedures as specified in Section 4.4.3.1.1
- o If the offer/answer exchange removed a data stream, or an answer rejected an offered data stream, an agent MUST flush the Valid list for that data stream. It MUST also terminate any STUN transactions in progress for that data stream.
- o If the offer/answer exchange added a new data stream, the agent MUST create a new checklist for it (and an empty Valid list to start of course) which in turn triggers the candidate processing procedures [RFC8445].

- o If the checklist state associated with a data stream is Running, the agent recomputes the checklist. If a pair on the new checklist was also on the previous checklist, its candidate pair state is copied over. Otherwise, its candidate pair state is set to Frozen. If none of the checklists are active (meaning that the candidate pair states in each checklist are Frozen), appropriate procedures in [RFC8445] are performed to move candidate pair(s) to the Waiting state to further continue ICE processing.
- o If the ICE state is Completed and the SDP answer conforms to Section 4.4.2, the agent MUST remain in the Completed ICE state.

However, if the ICE support is no longer indicated in the SDP answer, the agent MUST fall-back to [RFC3264] procedures and SHOULD NOT drop the dialog because of the missing ICE support or unexpected answer. Once the agent sends a new offer later on, it MUST perform an ICE restart.

4.4.3.1.1. ICE Restarts

The agent MUST remember the nominated pair in the Valid list for each component of the data stream, called the "previous selected pair", prior to the restart. The agent will continue to send media using this pair, as described in section 12 of [RFC8445]. Once these destinations are noted, the agent MUST flush the Valid lists and checklists, and then recompute the checklist and its states, thus triggering the candidate processing procedures [RFC8445]

4.4.3.2. Procedures for Lite Implementations

If ICE is restarting for a data stream, the agent MUST create a new Valid list for that data stream. It MUST remember the nominated pair in the previous Valid list for each component of the data stream, called the "previous selected pairs", and continue to send media there as described in section 12 of [RFC8445]. The state of each checklist for each data stream MUST change to Running, and the ICE state MUST be set 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 cand-extension)

foundation          = 1*32ice-char
component-id        = 1*3DIGIT
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
cand-extension      = extension-att-name SP extension-att-value
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, allowing for IPv4 addresses, IPv6 addresses, and fully qualified domain names (FQDNs). 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 generating local candidates MUST NOT use FQDN addresses. An agent processing remote candidates MUST ignore candidate lines that include candidates with FQDN or IP address versions that are not supported or recognized. The procedures for generation and handling of FQDN candidates, as well

as, how agents indicate support for such procedures, need to be specified in an extension specification.

<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 by extending the sub-registry "ICE Transport Protocols" under "Interactive Connectivity Establishment (ICE)" registry.

<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 [RFC8445]

<component-id>: is a positive integer between 1 and 256 (inclusive) that identifies the specific component of the data 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 data 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 13 in [RFC8445] for additional discussion on extending ICE to new data streams.

<priority>: is a positive integer between 1 and $(2^{31} - 1)$ inclusive. The procedures for computing candidate's priority is described in section 5.1.2 of [RFC8445].

<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. Specifications for new candidate types MUST define how, if at all, various steps in the ICE processing differ from the ones defined by this specification.

<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 Appendix B.3 of [RFC8445] 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 '9'.

The candidate attribute can itself be extended. The grammar allows for new name/value pairs to be added at the end of the attribute. Such extensions MUST be made through IETF Review or IESG Approval [RFC8126] and the assignments MUST contain the specific extension and a reference to the document defining the usage of the extension.

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
203.0.113.141 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 data stream. This attribute MUST be included in an offer by a controlling agent for a data 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 and only reported in the answer. It indicates that the offer arrived with a default destination for a media component that didn't have a corresponding candidate attribute. Inclusion of "a=ice-mismatch" attribute for a given data stream implies that even though both agents support ICE, ICE procedures MUST NOT be used for this data stream and [RFC3264] procedures MUST be used instead.

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 data 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 data stream. If two data 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 same applies when ICE is restarting for an agent).

[RFC8445] requires the ice-ufrag attribute to contain at least 24 bits of randomness, and the ice-pwd attribute to 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" is a session level attribute that indicates the desired connectivity check pacing (Ta interval), in milliseconds, that the sender wishes to use. See section 14.2 of [RFC8445] for more information regarding selecting a pacing value. The syntax is:

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

If absent in an offer or answer the default value of the attribute is 50 ms, which is the recommended value specified in [RFC8445].

Once both agents have indicated the pacing value they wish to use, both agents MUST use the larger of the indicated values.

Example shows an ice-pacing SDP line with value '50':

```
a=ice-pacing:50
```

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
                      *(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 it is willing to use it, if the peer agent also includes the same extension in the answer. There might be further extension specific negotiation needed between the agents that determine how the extension gets used in a given session. The details of the negotiation procedures, if present, MUST be defined by the specification defining the extension (Section 10.2).

Example shows an ice-options SDP line with 'ice2' and 'rtp+ecn' [RFC6679] values :

```
a=ice-options:ice2 rtp+ecn
```

6. Keepalives

All the ICE agents MUST follow the procedures defined in section 11 of [RFC8445] for sending keepalives. The keepalives MUST be sent regardless of whether the data 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. SIP Considerations

Note that ICE is not intended for NAT traversal for SIP signaling, 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, checklists, states, and so on.

7.1. 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.

7.1.1. Offer in INVITE

To reduce post-dial delays, it is RECOMMENDED that the caller begin gathering candidates prior to actually sending its initial INVITE, so that the candidates can be provided in the 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 as soon as it has completed gathering the candidates. 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 with the transport address matching the candidate address for one of the data 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 since there will be no Binding request sent and the number four is arbitrarily chosen to limit the number of 18x retransmits.

Once the answer has been sent, the agent SHOULD begin its connectivity checks. Once candidate pairs for each component of a data stream enter the valid list, the answerer can begin sending media on that data 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 data 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.

7.1.2. 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.

7.2. 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.

7.3. 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 data streams, it cannot determine which data 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.

7.4. 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 7.1 describe their own type of "preconditions", albeit with less functionality than those provided by the explicit preconditions in [RFC5898].

7.5. 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 data stream and go through the process of gathering new candidates. Furthermore, that list of candidates SHOULD include the ones currently being used for media.

8. 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 connection 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.

9. Security Considerations

The generic ICE security considerations are defined in [RFC8445], and the generic SDP offer/answer security considerations are defined in [RFC3264]. These security considerations also apply to implementations of this document.

9.1. IP Address Privacy

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 '9'.

9.2. 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 data 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.

9.3. The Voice Hammer Attack

The voice hammer attack is an amplification attack, and can be triggered even if the attacker is an authenticated and valid participant in a session. In this attack, the attacker initiates sessions to other agents, and maliciously includes the connection 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). The use of ICE can help to prevent against this attack.

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.

SIP User Agents (UA) [RFC3261] that are not willing to receive non-ICE answers MUST include an "ice" Option Tag [RFC5768] in the SIP Require Header Field in their offer. UAs that reject non-ICE offers will generally use a 421 response code, together with an Option Tag "ice" in the Require Header Field in the response.

10. IANA Considerations

10.1. SDP Attributes

The original ICE specification defined seven new SDP attributes per the procedures of Section 8.2.4 of [RFC4566]. The registration information from the original specification is included here with modifications to include Mux Category and also defines a new SDP attribute 'ice-pacing'.

10.1.1. candidate Attribute

Attribute Name: candidate

Type of Attribute: media-level

Subject to charset: No

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.

Contact Name: IESG

Contact Email: iesg@ietf.org

Reference: RFCXXXX

Mux Category: TRANSPORT

10.1.2. remote-candidates Attribute

Attribute Name: remote-candidates

Type of Attribute: media-level

Subject to charset: No

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.

Contact Name: IESG

Contact Email: iesg@ietf.org

Reference: RFCXXXX

Mux Category: TRANSPORT

10.1.3. ice-lite Attribute

Attribute Name: ice-lite

Type of Attribute: session-level

Subject to charset: No

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.

Contact Name: IESG

Contact Email: iesg@ietf.org

Reference: RFCXXXX

Mux Category: NORMAL

10.1.4. ice-mismatch Attribute

Attribute Name: ice-mismatch

Type of Attribute: media-level

Subject to charset: No

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.

Contact Name: IESG

Contact e-mail: iesg@ietf.org

Reference: RFCXXXX

Mux Category: NORMAL

10.1.5. ice-pwd Attribute

Attribute Name: ice-pwd

Type of Attribute: session- or media-level

Subject to charset: No

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.

Contact Name: IESG

Contact e-mail: iesg@ietf.org

Reference: RFCXXXX

Mux Category: TRANSPORT

10.1.6. ice-ufrag Attribute

Attribute Name: ice-ufrag

Type of Attribute: session- or media-level

Subject to charset: No

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.

Contact Name: IESG

Contact e-mail: iesg@ietf.org

Reference: RFCXXXX

Mux Category: TRANSPORT

10.1.7. ice-options Attribute

Attribute Name: ice-options

Long Form: ice-options

Type of Attribute: session-level

Subject to charset: No

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.

Contact Name: IESG

Contact e-mail: iesg@ietf.org

Reference: RFCXXXX

Mux Category: NORMAL

10.1.8. ice-pacing Attribute

This specification also defines a new SDP attribute, "ice-pacing" according to the following data:

Attribute Name: ice-pacing

Type of Attribute: session-level

Subject to charset: No

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.

Contact Name: IESG

Contact e-mail: iesg@ietf.org

Reference: RFCXXXX

Mux Category: NORMAL

10.2. 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" [RFC8126].

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. ICE options are defined at the session level.

A registration request MUST include the following information:

- o The ICE option identifier to be registered
- 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

10.3. Candidate Attribute Extension Subregistry Establishment

This section creates a new sub-registry, "Candidate Attribute Extensions", under the sdp-parameters registry:
<http://www.iana.org/assignments/sdp-parameters>.

The purpose of the sub-registry is to register SDP candidate attribute extensions.

When a candidate extension is registered in the sub-registry, it needs to meet the "Specification Required" policies defined in [RFC8126].

Candidate attribute extensions MUST follow the 'cand-extension' syntax. The attribute extension name MUST follow the 'extension-att-name' syntax, and the attribute extension value MUST follow the 'extension-att-value' syntax.

A registration request MUST include the following information:

- o The name of the attribute extension.
- o A short description of the attribute extension.
- o A reference to a specification that describes the semantics, usage and possible values of the attribute extension.

11. 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.

Many thanks to Flemming Andreasen for shepherd review feedback.

Thanks to following experts for their reviews and constructive feedback: Thomas Stach, Adam Roach, Peter Saint-Andre, Roman Danyliw, Alissa Cooper, Benjamin Kaduk, Mirja Kuhlewind, Alexey Melnikov, Eric Vyncke for their detailed reviews.

12. Changes from RFC 5245

[RFC8445] describes the changes that were done to the common SIP procedures, including removal of aggressive nomination, modifying the procedures for calculating candidate pair states and scheduling connectivity checks and the calculation of timer values.

This document defines the following SDP offer/answer specific changes:

- o SDP offer/answer realization and usage of of 'ice2' option.
- o Definition and usage of SDP 'ice-pacing' attribute.
- o Explicit text that an ICE agent must not generate candidates with FQDNs, and must discard such candidates if received from the peer agent.
- o Relax requirement to include SDP 'rtcp' attribute.
- o Generic clarifications of SDP offer/answer procedures.

13. References

13.1. Normative References

[draft-ietf-ice-pac]

Holmberg, C. and J. Uberti, "Interactive Connectivity Establishment Patiently Awaiting Connectivity (ICE PAC)", draft-ietf-ice-pac-02 (work in progress), July 2019, <<http://www.ietf.org/internet-drafts/draft-ietf-ice-pac-02.txt>>.

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://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, <<https://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, <<https://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, <<https://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, <<https://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, <<https://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, <<https://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, <<https://www.rfc-editor.org/info/rfc4032>>.
- [RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, DOI 10.17487/RFC4566, July 2006, <<https://www.rfc-editor.org/info/rfc4566>>.

- [RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <<https://www.rfc-editor.org/info/rfc5234>>.
- [RFC5768] Rosenberg, J., "Indicating Support for Interactive Connectivity Establishment (ICE) in the Session Initiation Protocol (SIP)", RFC 5768, DOI 10.17487/RFC5768, April 2010, <<https://www.rfc-editor.org/info/rfc5768>>.
- [RFC6336] Westerlund, M. and C. Perkins, "IANA Registry for Interactive Connectivity Establishment (ICE) Options", RFC 6336, DOI 10.17487/RFC6336, July 2011, <<https://www.rfc-editor.org/info/rfc6336>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8445] Keranen, A., Holmberg, C., and J. Rosenberg, "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal", RFC 8445, DOI 10.17487/RFC8445, July 2018, <<https://www.rfc-editor.org/info/rfc8445>>.

13.2. 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, <<https://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, <<https://www.rfc-editor.org/info/rfc3960>>.
- [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, <<https://www.rfc-editor.org/info/rfc5245>>.

- [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, <<https://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, <<https://www.rfc-editor.org/info/rfc5898>>.
- [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, <<https://www.rfc-editor.org/info/rfc6679>>.
- [RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <<https://www.rfc-editor.org/info/rfc8126>>.

Appendix A. Examples

For the example shown in section 15 of [RFC8445] 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-options:ice2
a=ice-pacing:50
a=ice-pwd:asd88fgpdd777uzjYhagZg
a=ice-ufraq: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:db8:8101:3a55:4858:a2a9:22ff:99b9
t=0 0
a=ice-options:ice2
a=ice-pacing:50
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:db8:8101:3a55:4858:a2a9:22ff:99b9
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-options:ice2
a=ice-pacing:50
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-options:ice2
a=ice-pacing:50
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 data 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.

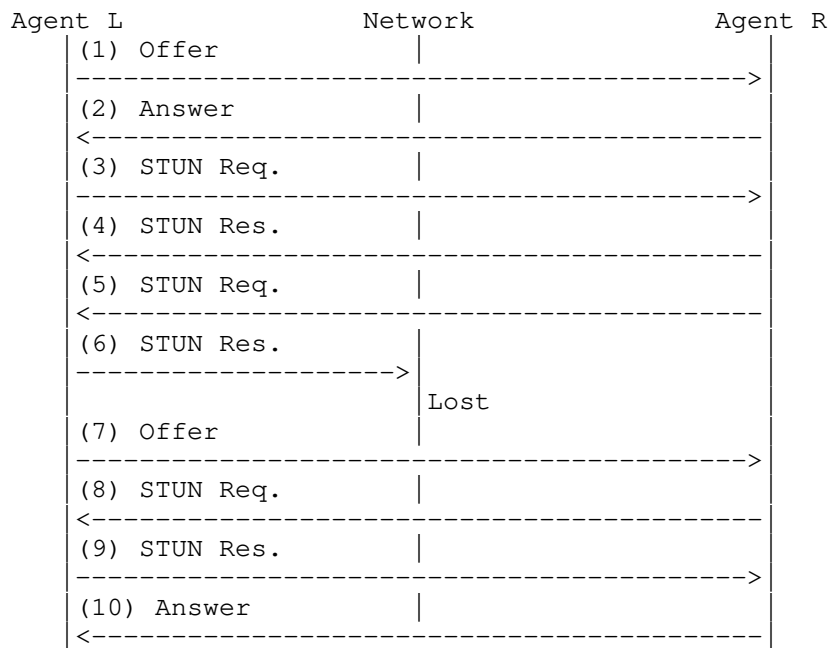


Figure 1: Race Condition Flow

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:

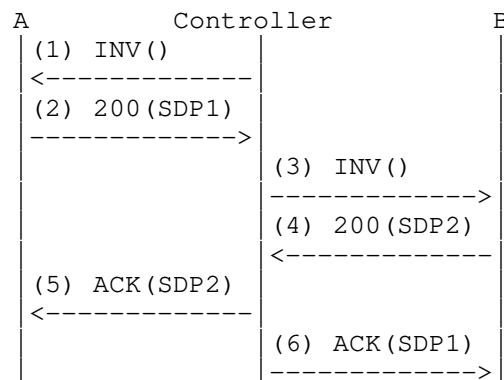


Figure 2: Role Conflict Flow

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 raises 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.

Appendix E. Contributors

Following experts have contributed textual and structural improvements for this work

1. Thomas Stach

* thomass.stach@gmail.com

Authors' Addresses

Marc Petit-Huguenin
Impedance Mismatch

Email: marc@petit-huguenin.org

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

Email: snandaku@cisco.com

Christer Holmberg
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

Email: christer.holmberg@ericsson.com

Ari Keranen
Ericsson
Jorvas 02420
Finland

Email: ari.keranen@ericsson.com

Roman Shpount
TurboBridge
4905 Del Ray Avenue, Suite 300
Bethesda, MD 20814
USA

Phone: +1 (240) 292-6632
Email: rshpount@turbobridge.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 24, 2018

E. Iovov
Jitsi
T. Stach
Unaffiliated
E. Marocco
Telecom Italia
C. Holmberg
Ericsson
June 22, 2018

A Session Initiation Protocol (SIP) Usage for Incremental Provisioning
of Candidates for the Interactive Connectivity Establishment (Trickle
ICE)
draft-ietf-mmusic-trickle-ice-sip-18

Abstract

The Interactive Connectivity Establishment (ICE) protocol describes a Network Address Translator (NAT) traversal mechanism for UDP-based multimedia sessions established with the Offer/Answer model. The ICE extension for Incremental Provisioning of Candidates (Trickle ICE) defines a mechanism that allows ICE Agents to shorten session establishment delays by making the candidate gathering and connectivity checking phases of ICE non-blocking and by executing them in parallel.

This document defines usage semantics for Trickle ICE with the Session Initiation Protocol (SIP). The document also defines a new SIP Info Package to support this usage together with the corresponding media type. Additionally, a new SDP 'end-of-candidates' attribute and a new SIP Option Tag 'trickle-ice' are defined.

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 <https://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 December 24, 2018.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Terminology	4
3. Protocol Overview	4
3.1. Discovery issues	5
3.2. Relationship with the Offer/Answer Model	6
4. Incremental Signaling of ICE candidates	7
4.1. Initial Offer/Answer Exchange	8
4.1.1. Sending the Initial Offer	8
4.1.2. Receiving the Initial Offer	9
4.1.3. Sending the Initial Answer	9
4.1.4. Receiving the Initial Answer	10
4.2. Subsequent Offer/Answer Exchanges	10
4.3. Establishing the Dialog	10
4.3.1. Establishing Dialog State through Reliable Offer/Answer Delivery	11
4.3.2. Establishing Dialog State through Unreliable Offer/Answer Delivery	12
4.3.3. Initiating Trickle ICE without an SDP Answer	14
4.4. Delivering Candidates in INFO Requests	16
5. Initial Discovery of Trickle ICE Support	20
5.1. Provisioning Support for Trickle ICE	20
5.2. Trickle ICE Discovery with Globally Routable User Agent URIs (GRUU)	20
5.3. Fall-back to Half Trickle	21
6. Considerations for RTP and RTCP Multiplexing	23
7. Considerations for Media Multiplexing	26
8. SDP 'end-of-candidates' Attribute	28
8.1. Definition	28
8.2. Offer/Answer Procedures	29

9.	Content Type 'application/trickle-ice-sdpfrag'	29
9.1.	Overall Description	29
9.2.	Grammar	29
10.	Info Package	32
10.1.	Rationale - Why INFO?	32
10.2.	Overall Description	33
10.3.	Applicability	33
10.4.	Info Package Name	34
10.5.	Info Package Parameters	34
10.6.	SIP Option Tags	34
10.7.	Info Request Body Parts	34
10.8.	Info Package Usage Restrictions	34
10.9.	Rate of INFO Requests	34
10.10.	Info Package Security Considerations	35
11.	Deployment Considerations	35
12.	IANA Considerations	35
12.1.	SDP 'end-of-candidates' Attribute	35
12.2.	Media Type 'application/trickle-ice-sdpfrag'	36
12.3.	SIP Info Package 'trickle-ice'	38
12.4.	SIP Option Tag 'trickle-ice'	38
13.	Security Considerations	38
14.	Acknowledgements	39
15.	Change Log	39
16.	References	43
16.1.	Normative References	43
16.2.	Informative References	46
	Authors' Addresses	46

1. Introduction

The Interactive Connectivity Establishment (ICE) protocol [I-D.ietf-ice-rfc5245bis] describes a mechanism for Network Address Translator (NAT) traversal that consists of three main phases.

During the first phase an agent gathers a set of candidate transport addresses (source IP address, port and transport protocol). This is followed by a second phase where these candidates are sent to a remote agent within the Session Description Protocol (SDP) body of a SIP message. At the remote agent the gathering procedure is repeated and candidates are sent to the first agent. Once the candidate information is available, a third phase starts in parallel where connectivity between all candidates in both sets is checked (connectivity checks). Once these phases have been completed, and only then, both agents can begin communication.

According to [I-D.ietf-ice-rfc5245bis] the three phases above happen consecutively, in a blocking way, which can introduce undesirable setup delay during session establishment. The Trickle ICE extension

[I-D.ietf-ice-trickle] defines generic semantics required for these ICE phases to happen in a parallel, non-blocking way and hence speed up session establishment.

This specification defines a usage of Trickle ICE with the Session Initiation Protocol (SIP)[RFC3261]. It describes how ICE candidates are to be exchanged incrementally using SIP INFO requests [RFC6086] and how the Half Trickle and Full Trickle modes defined in [I-D.ietf-ice-trickle] are to be used by SIP User Agents (UAs) depending on their expectations for support of Trickle ICE by a remote agent.

This document defines a new Info Package as specified in [RFC6086] for use with Trickle ICE together with the corresponding media type, SDP attribute and SIP option tag.

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 BCP 14 [RFC2119], [RFC8174] when, and only when, they appear in all capitals, as shown here.

This specification makes use of terminology defined by the protocol for Interactive Connectivity Establishment in [I-D.ietf-ice-rfc5245bis] and its Trickle ICE extension [I-D.ietf-ice-trickle]. It is assumed that the reader is familiar with the terminology from both documents.

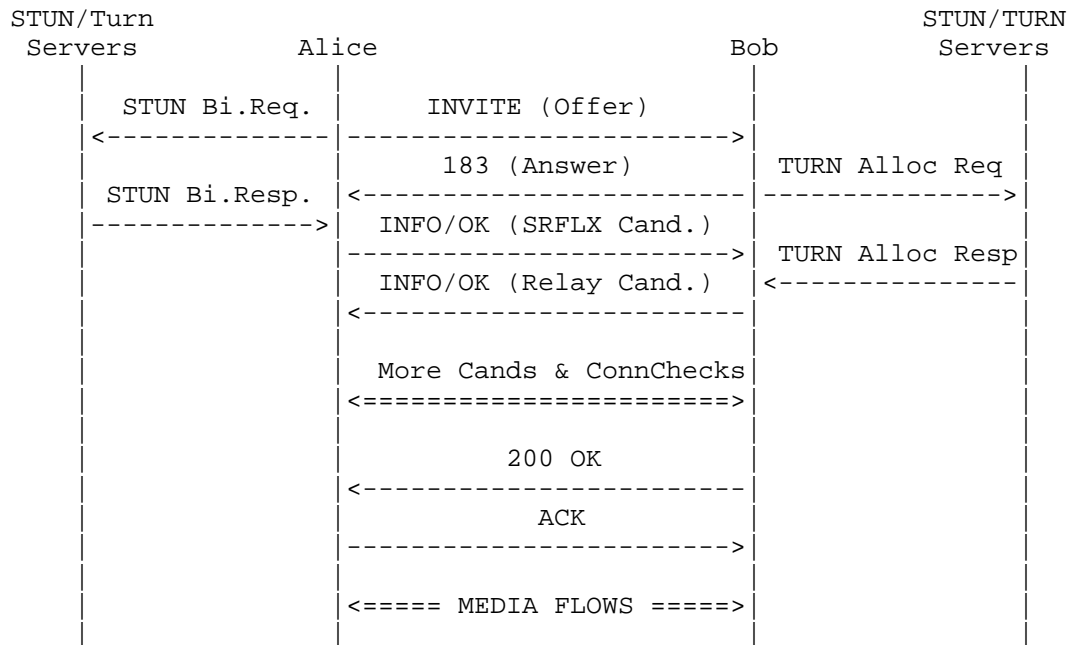
[I-D.ietf-ice-rfc5245bis] also describes how ICE makes use of the Session Traversal Utilities for NAT (STUN) protocol [RFC5389] and its extension Traversal Using Relay NAT (TURN) [RFC5766].

3. Protocol Overview

When using ICE for SIP according to [I-D.ietf-mmusic-ice-sip-sdp] the ICE candidates are exchanged solely via SDP Offer/Answer as per [RFC3264]. This specification defines an additional mechanism where candidates can be exchanged using SIP INFO messages and a newly defined Info Package [RFC6086]. This allows ICE candidates also to be sent in parallel to an ongoing Offer/Answer negotiation and/or after the completion of the Offer/Answer negotiation.

Typically, in cases where Trickle ICE is fully supported, the Offerer sends an INVITE request containing a subset of candidates. Once an early dialog is established the Offerer can continue sending candidates in INFO requests within that dialog.

Similarly, an Answerer can send ICE candidates using INFO requests within the dialog established by its 18x provisional response. Figure 1 shows such a sample exchange:



Note: SRFLX denotes server-reflexive candidates

Figure 1: Sample Trickle ICE scenario with SIP

3.1. Discovery issues

In order to benefit from Trickle ICE’s full potential and reduce session establishment latency to a minimum, Trickle ICE agents need to generate SDP Offers and Answers that contain incomplete, potentially empty sets of candidates. Such Offers and Answers can only be handled meaningfully by agents that actually support incremental candidate provisioning, which implies the need to confirm such support before using it.

Contrary to other protocols, where "in advance" capability discovery is widely implemented, the mechanisms that allow this for SIP (i.e., a combination of UA Capabilities [RFC3840] and Globally Routable User Agent URIs (GRUU) [RFC5627]) have only seen low levels of adoption. This presents an issue for Trickle ICE implementations as SIP UAs do not have an obvious means of verifying that their peer will support incremental candidate provisioning.

The Half Trickle mode of operation defined in the Trickle ICE specification [I-D.ietf-ice-trickle] provides one way around this, by requiring the first Offer to contain a complete set of local ICE candidates and only using incremental provisioning of remote candidates for the rest of the session.

While using Half Trickle does provide a working solution it also comes at the price of increased latency. Section 5 therefore makes several alternative suggestions that enable SIP UAs to engage in Full Trickle right from their first Offer: Section 5.1 discusses the use of on-line provisioning as a means of allowing use of Trickle ICE for all endpoints in controlled environments. Section 5.2 describes anticipatory discovery for implementations that actually do support GRUU and UA Capabilities and Section 5.3 discusses the implementation and use of Half Trickle by SIP UAs where none of the above are an option.

3.2. Relationship with the Offer/Answer Model

From the perspective of SIP middle boxes and proxies the Offer/Answer exchange for Trickle ICE looks partly similar to the Offer/Answer exchange for regular ICE for SIP [I-D.ietf-mmusic-ice-sip-sdp]. However, in order to have the full picture of the candidate exchange, the newly introduced INFO messages need to be considered as well.

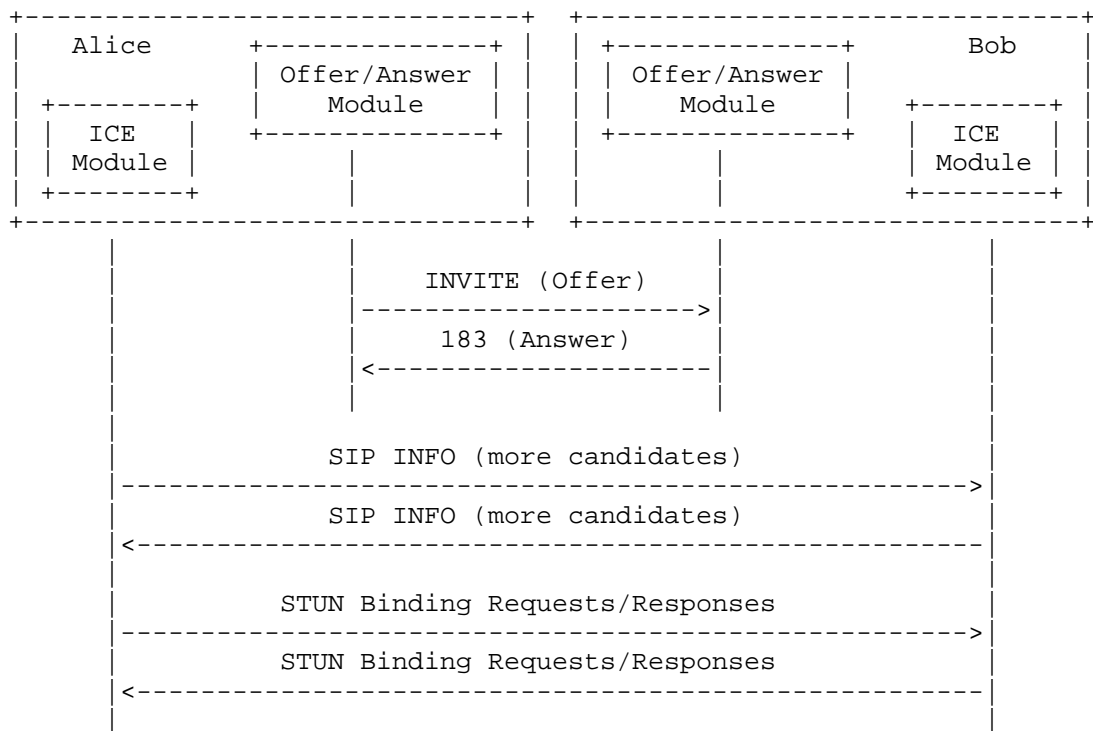


Figure 2: Distinguishing between Trickle ICE and traditional signaling.

From an architectural viewpoint, as displayed in Figure 2, exchanging candidates through SIP INFO requests could be represented as signaling between ICE modules and not between Offer/Answer modules of SIP User Agents. Then, such INFO requests do not impact the state of the Offer/Answer transaction other than providing additional candidates. Consequently, INFO requests are not considered Offers or Answers. Nevertheless, candidates that have been exchanged using INFO requests SHALL be included in subsequent Offers or Answers. The version number in the "o=" line of that subsequent Offer needs to be incremented by 1 per the rules in [RFC3264].

4. Incremental Signaling of ICE candidates

Trickle ICE Agents will exchange ICE descriptions compliant to [I-D.ietf-ice-trickle] via Offer/Answer procedures and/or INFO request bodies. This requires the following SIP-specific extensions:

1. Trickle ICE Agents MUST indicate support for Trickle ICE by including the SIP option-tag 'trickle-ice' in a SIP Supported: header field within all SIP INVITE requests and responses.
2. Trickle ICE Agents MUST indicate support for Trickle ICE by including the ice-option 'trickle' within all SDP Offers and Answers in accordance to [I-D.ietf-ice-trickle].
3. Trickle ICE Agents MAY include any number of ICE candidates, i.e. from zero to the complete set of candidates, in their initial Offer or Answer. If the complete candidate set is included already in the initial Offer, this is called Half-Trickle.
4. Trickle ICE Agents MAY exchange additional ICE candidates using INFO requests within an existing INVITE dialog usage (including an early dialog) as specified in [RFC6086]. The INFO requests carry an Info-Package: trickle-ice. Trickle ICE Agents MUST be prepared to receive INFO requests within that same dialog usage, containing additional candidates and/or an indication that trickling of such candidates has ended.
5. Trickle ICE Agents MAY exchange additional ICE candidates before the Answerer has sent the Answer provided that an invite dialog usage is established at both Trickle ICE Agents. Note that in case of forking multiple early dialogs may exist.

The following sections provide further details on how Trickle ICE Agents perform the initial Offer/Answer exchange (Section 4.1), perform subsequent Offer/Answer exchanges (Section 4.2) and establish the INVITE dialog usage (Section 4.3) such that they can incrementally trickle candidates (Section 4.4).

4.1. Initial Offer/Answer Exchange

4.1.1. Sending the Initial Offer

If the Offerer includes candidates in its initial Offer, it MUST encode these candidates as specified in [I-D.ietf-mmusic-ice-sip-sdp].

If the Offerer wants to send its initial Offer before knowing any candidate for one or more media descriptions, it MUST set the port to the default value '9' for these media descriptions. If the Offerer does not want to include the host IP address in the corresponding c-line, e.g. due to privacy reasons, it SHOULD include a default address in the c-line, which is set to the IPv4 address 0.0.0.0 or to the IPv6 equivalent ::.

In this case, the Offerer obviously cannot know the RTCP transport address and, thus, MUST NOT include the "a=rtcp" attribute [RFC6086]. This avoids potential ICE mismatch (see [I-D.ietf-mmusic-ice-sip-sdp]) for the RTCP transport address.

If the Offerer wants to use RTCP multiplexing [RFC5761] and/or exclusive RTCP multiplexing [I-D.ietf-mmusic-mux-exclusive], it still will include the "a=rtcp-mux" and/or "a=rctp-mux-only" attribute in the initial Offer.

In any case, the Offerer MUST include the attribute "a=ice-options:trickle" in accordance to [I-D.ietf-ice-trickle] and MUST include in each "m="-line a "a=mid:" attribute in accordance to [RFC5888]. The "a=mid:" attribute identifies the "m="-line to which a candidate belongs and helps in case of multiple "m="-lines, when candidates gathering could occur in a order different from the order of the "m="-lines.

4.1.2. Receiving the Initial Offer

If the initial Offer included candidates, the Answerer uses these candidates to start ICE processing as specified in [I-D.ietf-ice-trickle].

If the initial Offer included the attribute a=ice-options:trickle, the Answerer MUST be prepared for receiving trickled candidates later on.

In case of a "m/c=" line with default values none of the eventually trickled candidates will match the default destination. This situation MUST NOT cause an ICE mismatch (see [I-D.ietf-mmusic-ice-sip-sdp]).

4.1.3. Sending the Initial Answer

If the Answerer includes candidates in its initial Answer, it MUST encode these candidates as specified in [I-D.ietf-mmusic-ice-sip-sdp].

If the Answerer wants to send its initial Answer before knowing any candidate for one or more media descriptions, it MUST set the port to the default value '9' for these media descriptions. If the Answerer does not want to include the host IP address in the corresponding c-line, e.g. due to privacy reasons, it SHOULD include a default address in the c-line, which is set to the IPv4 address 0.0.0.0 or to the IPv6 equivalent ::.

In this case, the Answerer obviously cannot know the RTCP transport address and, thus, MUST NOT include the "a=rtcp" attribute [RFC6086]. This avoids potential ICE mismatch (see [I-D.ietf-mmusic-ice-sip-sdp]) for the RTCP transport address.

If the Answerer accepts to use RTCP multiplexing [RFC5761] and/or exclusive RTCP multiplexing [I-D.ietf-mmusic-mux-exclusive], it will include the "a=rtcp-mux" attribute in the initial Answer.

In any case, the Answerer MUST include the attribute "a=ice-options:trickle" in accordance to [I-D.ietf-ice-trickle] and MUST include in each "m="-line a "a=mid:" attribute in accordance to [RFC5888].

4.1.4. Receiving the Initial Answer

If the initial Answer included candidates, the Offerer uses these candidates to start ICE processing as specified in [I-D.ietf-ice-trickle].

In case of a "m/c=" line with default values none of the eventually trickled candidates will match the default destination. This situation MUST NOT cause an ICE mismatch (see [I-D.ietf-mmusic-ice-sip-sdp]).

4.2. Subsequent Offer/Answer Exchanges

Subsequent Offer/Answer exchanges are handled as for regular ICE (see section 4.2 of [I-D.ietf-mmusic-ice-sip-sdp]).

If an Offer or Answer needs to be sent while the ICE agents are in the middle of trickling section 3.2 of [I-D.ietf-mmusic-ice-sip-sdp]) applies. This means that an ICE agent includes candidate attributes for all local candidates it had trickled previously for a specific media stream.

[RFC EDITOR NOTE: The section 3.2 in above sentence is correct for version 20 of said I-D. Authors need to cross-check during Auth48 since it could have have changed in the meantime.]

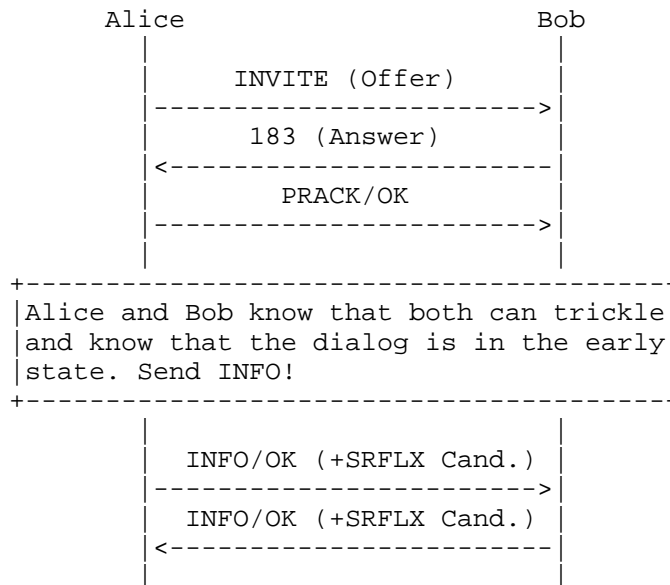
4.3. Establishing the Dialog

In order to be able to start trickling, the following two conditions need to be satisfied at the SIP UAs:

- o Trickle ICE support at the peer agent MUST be confirmed.
- o A dialog MUST have been created between the peers.

Section 5 discusses in detail the various options for satisfying the first of the above conditions. Regardless of those mechanisms, however, agents are certain to have a clear understanding of whether their peers support trickle ICE once an Offer and an Answer have been exchanged, which also allows for ICE processing to commence (see Figure 3).

4.3.1. Establishing Dialog State through Reliable Offer/Answer Delivery



Note: SRFLX denotes server-reflexive candidates

Figure 3: SIP Offerer can freely trickle as soon as it receives an Answer.

As shown in Figure 3 satisfying both conditions is relatively trivial for ICE Agents that have sent an Offer in an INVITE and that have received an Answer in a reliable provisional response. It is guaranteed to have confirmed support for Trickle ICE at the Answerer (or lack thereof) and to have fully initialized the SIP dialog at both ends. Offerers and Answerers (after receipt of the PRACK request) in the above situation can therefore freely commence trickling within the newly established dialog.

4.3.2. Establishing Dialog State through Unreliable Offer/Answer Delivery

The situation is a bit more delicate for agents that have received an Offer in an INVITE request and have sent an Answer in an unreliable provisional response because, once the response has been sent, the Answerer does not know when or if it has been received (Figure 4).

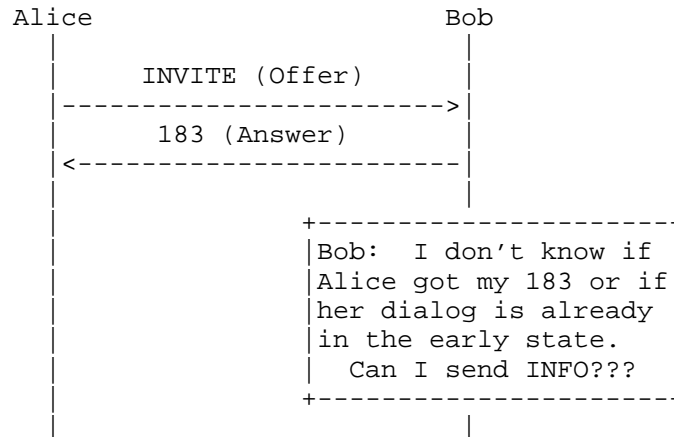


Figure 4: A SIP UA that sent an Answer in an unreliable provisional response does not know if it was received and if the dialog at the side of the Offerer has entered the early state

In order to clear this ambiguity as soon as possible, the Answerer needs to retransmit the provisional response with the exponential back-off timers described in [RFC3262]. These retransmissions MUST cease on receipt of an INFO request carrying a 'trickle-ice' Info Package body, on receipt of any other in-dialog request from the offerer or on transmission of the Answer in a 2xx response. The offerer cannot send in-dialog requests until it receives a response, so the arrival of such a request proves that the response has arrived. Using the INFO request for dialog confirmation is similar to the procedure described in section 6.1.1 of [I-D.ietf-mmusic-ice-sip-sdp] except that the STUN binding Request is replaced by the INFO request.

[RFC EDITOR NOTE: The section 6.1.1 in above sentence is correct for version 20 of said I-D. Authors need to cross-check during Auth48 since it could have have changed in the meantime.]

The Offerer MUST send a Trickle ICE INFO request as soon as it receives an SDP Answer in an unreliable provisional response. This INFO request MUST repeat the candidates that were already provided in

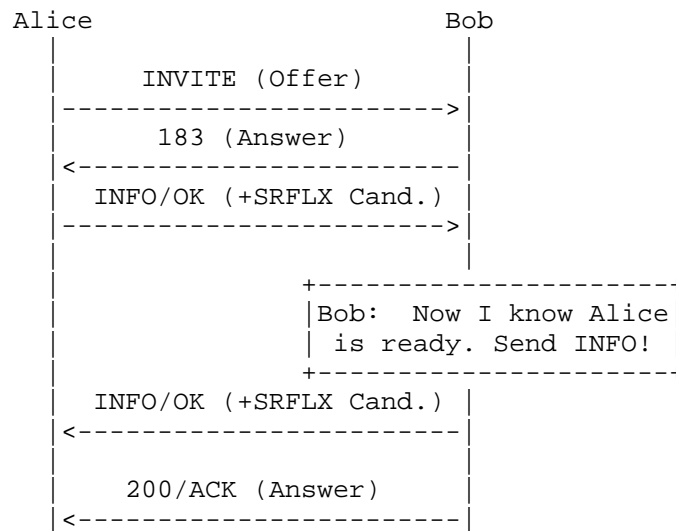
the Offer (as would be the case when Half Trickle is performed or when new candidates have not been learned since then). The first case could happen when Half Trickle is used and all candidate are already in the initial offer. The second case could happen when Full Trickle is used and the offerer is currently gathering additional candidates, but did not yet get them. Also, if the initial Offer did not contain any candidates, depending on how the Offerer gathers its candidates and how long it takes to do so, this INFO could still contain no candidates.

When Full Trickle is used and if newly learned candidates are available, the Offerer SHOULD also deliver these candidates in said INFO request, unless it wants to hold back some candidates in reserve, e.g. in case that these candidates are expensive to use and would only be trickled if all other candidates failed.

The Offerer SHOULD include an end-of-candidates attribute in case candidate discovery has ended in the mean time and no further candidates are to be trickled.

As soon as an Answerer has received such an INFO request, the Answerer has an indication that a dialog is established at both ends and can begin trickling (Figure 5).

Note: The +SRFLX in Figure 5 indicates that additionally newly learned server-reflexive candidates are included.



Note: SRFLX denotes server-reflexive candidates

Figure 5: A SIP UA that received an INFO request after sending an unreliable provisional response knows that the dialog at the side of the receiver has entered the early state

When sending the Answer in the 200 OK response to the INVITE request, the Answerer needs to repeat exactly the same Answer that was previously sent in the unreliable provisional response in order to fulfill the corresponding requirements in [RFC3264]. Thus, the Offerer needs to be prepared for receiving a different number of candidates in that repeated Answer than previously exchanged via trickling and MUST ignore the candidate information in that 200 OK response.

4.3.3. Initiating Trickle ICE without an SDP Answer

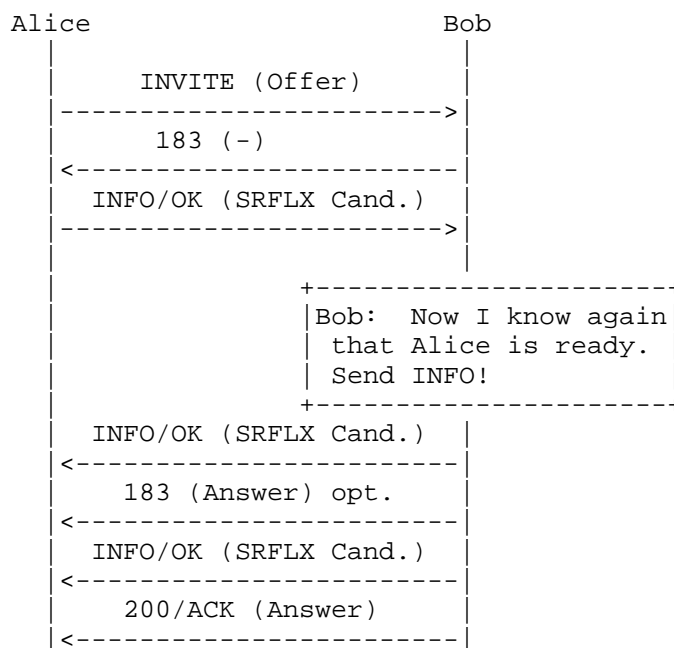
The ability to convey arbitrary candidates in INFO message bodies allows ICE Agents to initiate trickling without actually sending an Answer. Trickle ICE Agents can therefore respond to an INVITE request with provisional responses without an SDP Answer [RFC3261]. Such provisional responses serve for establishing an early dialog.

Agents that choose to establish the dialog in this way, MUST retransmit these responses with the exponential back-off timers described in [RFC3262]. These retransmissions MUST cease on receipt of an INFO request carrying a 'trickle-ice' Info Package body, on receipt any in-dialog request from the offerer or on transmission of

the Answer in a 2xx response. The offerer cannot send in-dialog requests until it receives a response, so the arrival of such a request proves that the response has arrived. This is again similar to the procedure described in section 6.1.1 of [I-D.ietf-mmusic-ice-sip-sdp] except that an Answer is not yet provided.

[RFC EDITOR NOTE: The section 6.1.1 in above sentence is correct for version 20 of said I-D. Authors need to cross-check during Auth48 since it could have have changed in the meantime.]

Note: The +SRFLX in Figure 6 indicates that additionally newly learned server-reflexive candidates are included.



Note: SRFLX denotes server-reflexive candidates

Figure 6: A SIP UA sends an unreliable provisional response without an Answer for establishing an early dialog

When sending the Answer, the agent MUST repeat all currently known and used candidates, if any, and MAY include all newly gathered candidates since the last INFO request was sent. However, if that Answer was already sent in a unreliable provisional response, the Answerers MUST repeat exactly the same Answer in the 200 OK response

to the INVITE request in order to fulfill the corresponding requirements in [RFC3264]. In case that trickling continued, an Offerer needs to be prepared for receiving fewer candidates in that repeated Answer than previously exchanged via trickling and MUST ignore the candidate information in that 200 OK response.

4.4. Delivering Candidates in INFO Requests

Whenever new ICE candidates become available for sending, agents encode them in "a=candidate:" attributes as described by [I-D.ietf-mmusic-ice-sip-sdp]. For example:

```
a=candidate:1 1 UDP 2130706432 200a0b:12f0::1 5000 typ host
```

The use of SIP INFO requests happens within the context of the Info Package as defined Section 10. The Media Type [RFC6838] for their payload MUST be set to 'application/trickle-ice-sdpfrag' as defined in Section 9. The Info request body adheres to the grammar as specified in Section 9.2.

Since neither the "a=candidate:" nor the "a=end-of-candidates" attributes contain information that would allow correlating them to a specific "m=" line, this is handled through the use of pseudo "m=" lines.

Pseudo "m=" lines follow the SDP syntax for "m=" lines as defined in [RFC4566] and are linked to the corresponding "m=" line in the SDP Offer or Answer via the identification tag in a "a=mid:" attribute [RFC5888]. A pseudo "m=" line does not provide semantics other than indicating to which "m=" line a candidate belongs. Consequently, the receiving agent MUST ignore any remaining content of the pseudo "m=" line, which is not defined in this document. This guarantees that the 'application/trickle-ice-sdpfrag' bodies do not interfere with the Offer/Answer procedures as specified in [RFC3264].

When sending the INFO request, the agent MAY, if already known to the agent, include the same content into the pseudo "m=" line as for the "m=" line in the corresponding Offer or Answer. However, since Trickle-ICE might be decoupled from the Offer/Answer negotiation this content might be unknown to the agent. In this case, the agent MUST include the following default values.

- o The media field is set to 'audio'.
- o The port value is set to '9'.

- o The proto value is set to 'RTP/AVP'.
- o The fmt field MUST appear only once and is set to '0'

Agents MUST include a pseudo "m=" line and an identification tag in a "a=mid:" attribute for every "m=" line whose candidate list they intend to update. Such "a=mid:" attributes MUST immediately precede the list of candidates for that specific "m=" line.

All "a=candidate:" or "a=end-of-candidates" attributes following an "a=mid:" attribute, up until (and excluding) the next occurrence of a pseudo "m=" line, pertain to the "m=" line identified by that identification tag.

Note, that there is no requirement that the Info request body contains as many pseudo m= lines as the Offer/Answer contains m=lines, nor that the pseudo m= lines be in the same order as the m=lines that they pertain to. The correspondence can be made via the "a=mid:" attributes since candidates are grouped in sections headed by "pseudo" m=lines. These sections contain "a=mid:" attribute values which point back to the true m=line.

An "a=end-of-candidates" attribute, preceding the first pseudo "m=" line, indicates the end of all trickling from that agent, as opposed to end of trickling for a specific "m=" line, which would be indicated by a media level "a=end-of-candidates" attribute.

Refer to Figure 7 for an example of the INFO request content.

The use of pseudo "m=" lines allows for a structure similar to the one in SDP Offers and Answers where separate media-level and session-level sections can be distinguished. In the current case, lines preceding the first pseudo "m=" line are considered to be session-level. Lines appearing in between or after pseudo "m=" lines will be interpreted as media-level.

Note that while this specification uses the "a=mid:" attribute from [RFC5888], it does not define any grouping semantics.

All INFO requests MUST carry the "a=ice-pwd:" and "a=ice-ufrag:" attributes that allow mapping them to a specific ICE generation. An agent MUST discard any received INFO requests containing "a=ice-pwd:" and "a=ice-ufrag:" attributes that do not match those of the current ICE Negotiation Session.

The "a=ice-pwd:" and "a=ice-ufrag:" attributes MUST appear at the same level as the ones in the Offer/Answer exchange. In other words, if they were present as session-level attributes, they will also

appear at the beginning of all INFO request payloads, i.e. preceding the first pseudo "m=" line. If they were originally exchanged as media level attributes, potentially overriding session-level values, then they will also be included in INFO request payloads following the corresponding pseudo "m=" lines.

Note that [I-D.ietf-ice-trickle] requires that when candidates are trickled, each candidate must be delivered to the receiving Trickle ICE implementation not more than once and in the same order as it was conveyed. If the signaling protocol provides any candidate retransmissions, they need to be hidden from the ICE implementation. This requirement is fulfilled as follows.

Since the agent is not fully aware of the state of the ICE Negotiation Session at its peer it MUST include all currently known and used local candidates in every INFO request. I.e. the agent MUST repeat in the INFO request body all candidates that were previously sent under the same combination of "a=ice-pwd:" and "a=ice-ufrag:" in the same order as they were sent before. In other words, the sequence of a previously sent list of candidates MUST NOT change in subsequent INFO requests and newly gathered candidates MUST be added at the end of that list. Although repeating all candidates creates some overhead, it also allows easier handling of problems that could arise from unreliable transports, like e.g. loss of messages and reordering, which can be detected through the CSeq: header field in the INFO request.

In addition, an ICE agent needs to adhere to section 17 of [I-D.ietf-ice-trickle] on preserving candidate order while trickling.

When receiving INFO requests carrying any candidates, agents MUST therefore first identify and discard the attribute lines containing candidates they have already received in previous INFO requests or in the Offer/Answer exchange preceding them.

Such candidates are considered to be equal if their IP address port, transport and component ID are the same. After identifying and discarding the known candidates, the agents MUST forward the actually new candidates to the ICE Agents in the same order as they were received in the INFO request body. The ICE Agents will then process the new candidates according to the rules described in [I-D.ietf-ice-trickle].

Receiving an "a=end-of-candidates" attribute in an INFO request body - with the "a=ice-ufrag" and "a=ice-pwd" attributes matching the current ICE generation - is an indication from the peer agent that it will not send any further candidates. When included at session level, i.e. before any pseudo "m=" line, this indication applies to

the whole session; when included at media level the indication applies only to the corresponding "m=" line. Handling of such end-of-candidates indications is defined in [I-D.ietf-ice-trickle].

The example in Figure 7 shows the content of a candidate delivering INFO request. In the example the "a=end-of-candidates" attributes indicate that the candidate gathering is finished and that no further INFO requests follow.

```
INFO sip:alice@example.com SIP/2.0
...
Info-Package: trickle-ice
Content-type: application/trickle-ice-sdpfrag
Content-Disposition: Info-Package
Content-length: 862

a=ice-pwd:asd88fgpdd777uzjYhagZg
a=ice-ufrag:8hhY
m=audio 9 RTP/AVP 0
a=mid:1
a=candidate:1 1 UDP 2130706432 2001:db8:a0b:12f0::1 5000 typ host
a=candidate:1 2 UDP 2130706432 2001:db8:a0b:12f0::1 5001 typ host
a=candidate:1 1 UDP 2130706431 192.0.2.1 5010 typ host
a=candidate:1 2 UDP 2130706431 192.0.2.1 5011 typ host
a=candidate:2 1 UDP 1694498815 192.0.2.3 5010 typ srflx
  raddr 192.0.2.1 rport 8998
a=candidate:2 2 UDP 1694498815 192.0.2.3 5011 typ srflx
  raddr 192.0.2.1 rport 8998
a=end-of-candidates
m=audio 9 RTP/AVP 0
a=mid:2
a=candidate:1 1 UDP 2130706432 2001:db8:a0b:12f0::1 6000 typ host
a=candidate:1 2 UDP 2130706432 2001:db8:a0b:12f0::1 6001 typ host
a=candidate:1 1 UDP 2130706431 192.0.2.1 6010 typ host
a=candidate:1 2 UDP 2130706431 192.0.2.1 6011 typ host
a=candidate:2 1 UDP 1694498815 192.0.2.3 6010 typ srflx
  raddr 192.0.2.1 rport 9998
a=candidate:2 2 UDP 1694498815 192.0.2.3 6011 typ srflx
  raddr 192.0.2.1 rport 9998
a=end-of-candidates
```

Note: In a real INFO request there will be no line breaks in the a=candidate: attributes

Figure 7: An Example for the Content of an INFO Request

5. Initial Discovery of Trickle ICE Support

SIP User Agents (UAs) that support and intend to use trickle ICE are required by [I-D.ietf-ice-trickle] to indicate that in their Offers and Answers using the attribute "a=ice-options:trickle" and MUST include the SIP option-tag "trickle-ice" in a SIP Supported: or Require: header field. This makes discovery fairly straightforward for Answerers or for cases where Offers need to be generated within existing dialogs (i.e., when sending UPDATE or re-INVITE requests). In both scenarios prior SDP bodies will have provided the necessary information.

Obviously, such information is not available at the time a first Offer is being constructed and it is therefore impossible for ICE Agents to determine support for incremental provisioning that way. The following options are suggested as ways of addressing this issue.

5.1. Provisioning Support for Trickle ICE

In certain situations it may be possible for integrators deploying Trickle ICE to know in advance that some or all endpoints reachable from within the deployment will support Trickle ICE. This is the case, for example, if Session Border Controllers (SBC) with support for this specification are used to connect to UAs that do not support Trickle ICE.

While the exact mechanism for allowing such provisioning is out of scope here, this specification encourages trickle ICE implementations to allow the option in the way they find most appropriate.

However, an Offerer assuming Trickle ICE support MUST include a SIP Require: trickle-ice header field. That way, if the provisioned assumption of Trickle ICE support ends up being incorrect, the failure is (a) operationally easy to track down, and (b) recoverable by the client, i.e., they can re-send the request without the SIP Require: header field and without the assumption of Trickle ICE support.

5.2. Trickle ICE Discovery with Globally Routable User Agent URIs (GRUU)

[RFC3840] provides a way for SIP User Agents to query for support of specific capabilities using, among others, OPTIONS requests. Support for GRUU according to [RFC5627] on the other hand allows SIP requests to be addressed to specific UAs (as opposed to arbitrary instances of an address of record). Combining the two and using the "trickle-ice" option tag defined in Section 10.6 provides SIP UAs with a way of learning the capabilities of specific SIP UA instances and then

addressing them directly with INVITE requests that require Trickle ICE support.

Such learning of capabilities may happen in different ways. One option for a SIP UA is to learn the GRUU instance ID of a peer through presence and then to query its capabilities with an OPTIONS request. Alternatively, it can also just send an OPTIONS request to the Address of Record (AOR) it intends to contact and then inspect the returned response(s) for support of both GRUU and Trickle ICE (Figure 8). It is noted that using the GRUU means that the INVITE request can go only to that particular device. This prevents the use of forking for that request.

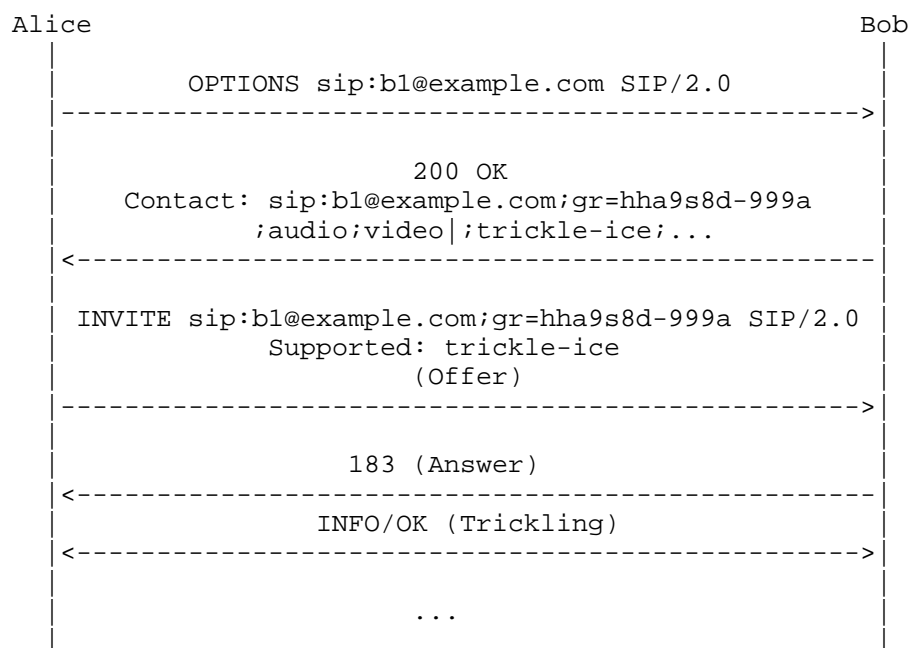


Figure 8: Trickle ICE support discovery with OPTIONS and GRUU

Confirming support for Trickle ICE through [RFC3840] gives SIP UAs the options to engage in Full Trickle negotiation (as opposed to the more lengthy Half Trickle) from the very first Offer they send.

5.3. Fall-back to Half Trickle

In cases where none of the other mechanisms in this section are acceptable, SIP UAs should use the Half Trickle mode defined in [I-D.ietf-ice-trickle]. With Half Trickle, agents initiate sessions

the same way they would when using ICE for SIP [I-D.ietf-mmusic-ice-sip-sdp]. This means that, prior to actually sending an Offer, agents first gather ICE candidates in a blocking way and then send them all in that Offer. The blocking nature of the process implies that some amount of latency will be accumulated and it is advised that agents try to anticipate it where possible, for example, when user actions indicate a high likelihood for an imminent call (e.g., activity on a keypad or a phone going off-hook).

Using Half Trickle results in Offers that are compatible with both ICE SIP endpoints and legacy [RFC3264] endpoints.

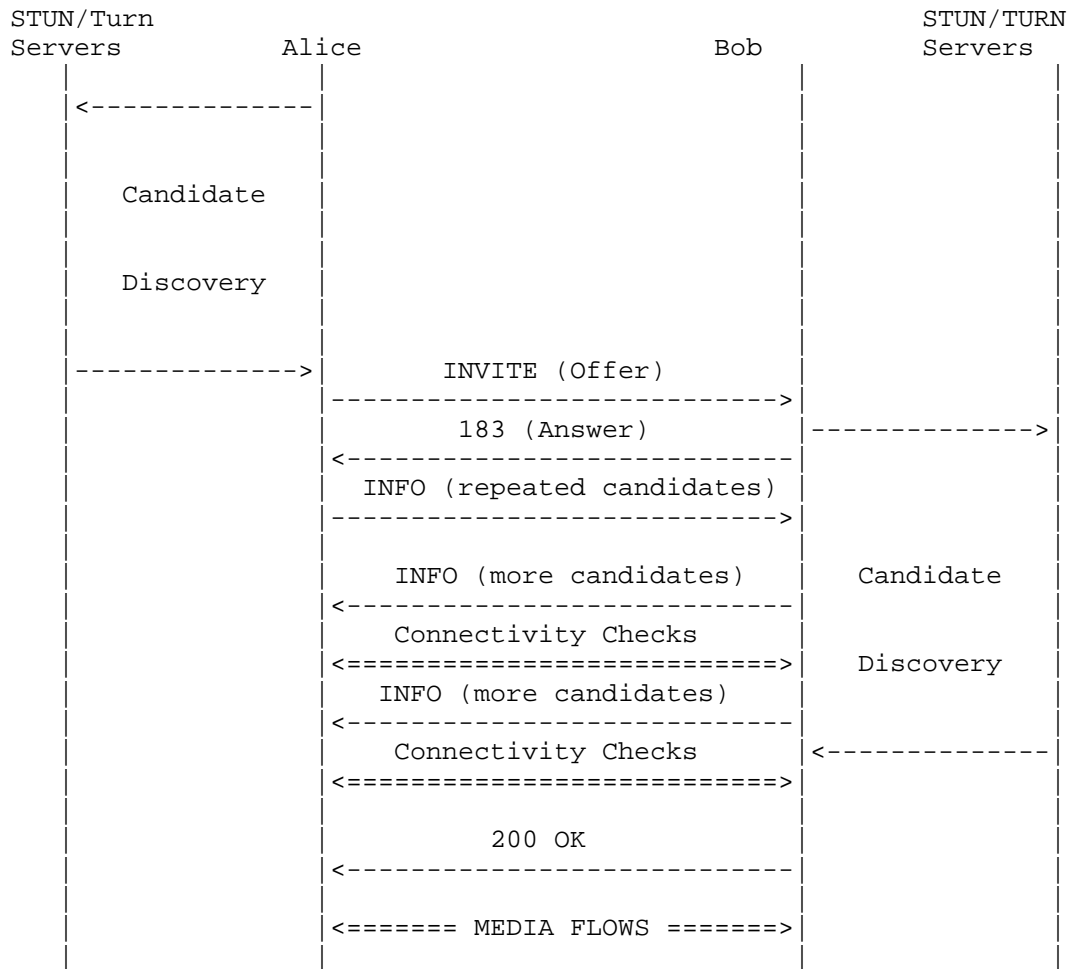


Figure 9: Example - A typical (Half) Trickle ICE exchange with SIP

It is worth reminding that once a single Offer or Answer had been exchanged within a specific dialog, support for Trickle ICE will have been determined. No further use of Half Trickle will therefore be necessary within that same dialog and all subsequent exchanges can use the Full Trickle mode of operation.

6. Considerations for RTP and RTCP Multiplexing

The following consideration describe options for Trickle-ICE in order to give some guidance to implementors on how trickling can be optimized with respect to providing RTCP candidates.

Handling of the "a=rtcp" attribute [RFC3605] and the "a=rtcp-mux" attribute for RTP/RTCP multiplexing [RFC5761] is already considered in section 5.1.1.1. of [I-D.ietf-ice-rfc5245bis] and as well in [RFC5761] itself. These considerations are still valid for Trickle ICE, however, trickling provides more flexibility for the sequence of candidate exchange in case of RTCP multiplexing.

[RFC EDITOR NOTE: The section 5.1.1.1 in above sentence is correct for version 17 of said I-D. Authors need to cross-check during Auth48 since it could have have changed in the meantime.]

If the Offerer supports RTP/RTCP multiplexing exclusively as specified in [I-D.ietf-mmusic-mux-exclusive], the procedures in that document apply for the handling of the "a=rtcp-mux-only", "a=rtcp" and the "a=rtcp-mux" attributes.

While a Half Trickle Offerer has to send an Offer compliant to [I-D.ietf-mmusic-ice-sip-sdp] and [RFC5761] including candidates for all components, the flexibility of a Full Trickle Offerer allows to send only RTP candidates (component 1) in the initial Offer assuming that RTCP multiplexing is supported by the Answerer. A Full Trickle Offerer would need to start gathering and trickling RTCP candidates (component 2) only after having received an indication in the Answer that the Answerer unexpectedly does not support RTCP multiplexing.

A Trickle Answerer MAY include an "a=rtcp-mux" attribute [RFC5761] in the application/trickle-ice-sdpfrag body if it supports and uses RTP and RTCP multiplexing. The Trickle Answerer needs to follow the guidance on the usage of the "a=rtcp" attribute as given in [I-D.ietf-mmusic-ice-sip-sdp] and [RFC3605]. Receipt of this attribute at the Offerer in an INFO request prior to the Answer indicates that the Answerer supports and uses RTP and RTCP multiplexing. The Offerer can use this information e.g. for stopping gathering of RTCP candidates and/or for freeing corresponding resources.

This behavior is illustrated by the following example Offer that indicates support for RTP and RTCP multiplexing.

```
v=0
o=alice 2890844526 2890844526 IN IP6 atlanta.example.com
s=
c=IN IP6 2001:db8:a0b:12f0::3
t=0 0
a=ice-pwd:777uzjYhagZgasd88fgpdd
a=ice-ufrag:Yhh8
m=audio 5000 RTP/AVP 0
a=mid:1
a=rtcp-mux
a=candidate:1 1 UDP 1658497328 2001:db8:a0b:12f0::3 5000 typ host
```

Once the dialog is established as described in section Section 4.3 the Answerer sends the following INFO request.

```
INFO sip:alice@example.com SIP/2.0
...
Info-Package: trickle-ice
Content-type: application/trickle-ice-sdpfrag
Content-Disposition: Info-Package
Content-length: 161

a=ice-pwd:asd88fgpdd777uzjYhagZg
a=ice-ufrag:8hhY
m=audio 9 RTP/AVP 0
a=mid:1
a=rtcp-mux
a=candidate:1 1 UDP 1658497382 2001:db8:a0b:12f0::4 6000 typ host
```

This INFO request indicates that the Answerer supports and uses RTP and RTCP multiplexing as well. It allows the Offerer to omit gathering of RTCP candidates or releasing already gathered RTCP candidates. If the INFO request did not contain the `a=rtcp-mux` attribute, the Offerer has to gather RTCP candidates unless it wants to wait until receipt of an Answer that eventually confirms support or non-support for RTP and RTCP multiplexing. In case the Offerer had sent RTCP candidates in a previous INFO request, it still needs to repeat them in subsequent INFO requests, even in case that support for RTCP multiplexing was confirmed by the Answerer and the Offerer has released its RTCP candidates.

7. Considerations for Media Multiplexing

The following considerations describe options for Trickle-ICE in order to give some guidance to implementors on how trickling can be optimized with respect to providing candidates in case of Media Multiplexing [I-D.ietf-mmusic-sdp-bundle-negotiation]. It is assumed that the reader is familiar with [I-D.ietf-mmusic-sdp-bundle-negotiation].

ICE candidate exchange is already considered in section 11 of [I-D.ietf-mmusic-sdp-bundle-negotiation]. These considerations are still valid for Trickle ICE, however, trickling provides more flexibility for the sequence of candidate exchange, especially in Full Trickle mode.

Except for bundle-only "m=" lines, a Half Trickle Offerer has to send an Offer with candidates for all bundled "m=" lines. The additional flexibility, however, allows a Full Trickle Offerer to initially send only candidates for the "m=" line with the suggested Offerer BUNDLE address.

On receipt of the Answer, the Offerer will detect if BUNDLE is supported by the Answerer and if the suggested Offerer BUNDLE address was selected. In this case, the Offerer does not need to trickle further candidates for the remaining "m=" lines in a bundle. However, if BUNDLE is not supported, the Full Trickle Offerer needs to gather and trickle candidates for the remaining "m=" lines as necessary. If the Answerer selects an Offerer BUNDLE address different from the suggested Offerer BUNDLE address, the Full Trickle Offerer needs to gather and trickle candidates for the "m=" line that carries the selected Offerer BUNDLE address.

A Trickle Answerer SHOULD include an "a=group:BUNDLE" attribute [I-D.ietf-mmusic-sdp-bundle-negotiation] at session level in the application/trickle-ice-sdpfrag body if it supports and uses bundling. When doing so, the Answerer MUST include all identification-tags in the same order that is used or will be used in the Answer.

Receipt of this attribute at the Offerer in an INFO request prior to the Answer indicates that the Answerer supports and uses bundling. The Offerer can use this information e.g. for stopping the gathering of candidates for the remaining "m=" lines in a bundle and/or for freeing corresponding resources.

This behaviour is illustrated by the following example Offer that indicates support for Media Multiplexing.

In case the Offerer had sent already candidates for "m="-lines in a bundle in a previous INFO request, it still needs to repeat them in subsequent INFO requests, even in case that support for bundling was confirmed by the Answerer and the Offerer has released no longer needed candidates.

```
v=0
o=alice 2890844526 2890844526 IN IP6 atlanta.example.com
s=
c=IN IP6 2001:db8:a0b:12f0::3
t=0 0
a=group:BUNDLE foo bar
a=ice-pwd:777uzjYhagZgasd88fgpdd
a=ice-frag:Yhh8
m=audio 10000 RTP/AVP 0
a=mid:foo
a=rtcp-mux
a=rtpmap:0 PCMU/8000
a=extmap 1 urn:ietf:params:rtp-hdext:sdes:mid
a=candidate:1 1 UDP 1658497328 2001:db8:a0b:12f0::3 10000 typ host
m=video 10002 RTP/AVP 31
a=mid:bar
a=rtcp-mux
a=rtpmap:31 H261/90000
a=extmap 1 urn:ietf:params:rtp-hdext:sdes:mid
```

The example Offer indicates support for RTP and RTCP multiplexing and contains a "a=candidate:" attribute only for the "m="-line with the suggested Offerer bundle address. Once the dialog is established as described in Section 4.3 the Answerer sends the following INFO request.


```
INFO sip:alice@example.com SIP/2.0
...
Info-Package: trickle-ice
Content-type: application/trickle-ice-sdpfrag
Content-Disposition: Info-Package
Content-length: 219

a=group:BUNDLE foo bar
a=ice-pwd:asd88fgpdd777uzjYhagZg
a=ice-ufrag:8hhY
m=audio 9 RTP/AVP 0
a=mid:foo
a=rtcp-mux
a=candidate:1 1 UDP 1658497328 2001:db8:a0b:12f0::3 5000 typ host
```

This INFO request indicates that the Answerer supports and uses Media Multiplexing as well. Note that the Answerer only includes a single pseudo "m="-line since candidates matching those from the second "m="-line in the offer are not needed from the Answerer.

The INFO request also indicates that the Answerer accepted the suggested Offerer Bundle Address. This allows the Offerer to omit gathering of RTP and RTCP candidates for the other "m=" lines or releasing already gathered candidates. If the INFO request did not contain the a=group:BUNDLE attribute, the Offerer has to gather RTP and RTCP candidates for the other "m=" lines unless it wants to wait until receipt of an Answer that eventually confirms support or non-support for Media Multiplexing.

Independent of using Full Trickle or Half Trickle mode, the rules from [I-D.ietf-mmusic-sdp-mux-attributes] apply to both, Offerer and Answerer, when putting attributes as specified in Section 9.2 in the application/trickle-ice-sdpfrag body.

8. SDP 'end-of-candidates' Attribute

8.1. Definition

This section defines a new SDP media-level and session-level attribute [RFC4566] 'end-of-candidates'. 'end-of-candidates' is a property attribute [RFC4566], and hence has no value. By including this attribute in an Offer or Answer the sending agent indicates that it will not trickle further candidates. When included at session level this indication applies to the whole session, when included at media level the indication applies only to the corresponding media description.

Name: end-of-candidates

Value: N/A

Usage Level: media and session-level

Charset Dependent: no

Mux Category: IDENTICAL

Example: a=end-of-candidates

8.2. Offer/Answer Procedures

The Offerer or Answerer MAY include an "a=end-of-candidates" attribute in case candidate discovery has ended and no further candidates are to be trickled. The Offerer or Answerer MUST provide the "a=end-of-candidates" attribute together with the "a=ice-ufrag" and "a=ice-pwd" attributes of the current ICE generation as required by [I-D.ietf-ice-trickle]. When included at session level this indication applies to the whole session; when included at media level the indication applies only to the corresponding media description.

Receipt of an "a=end-of-candidates" attribute at an Offerer or Answerer - with the "a=ice-ufrag" and "a=ice-pwd" attributes matching the current ICE generation - indicates that gathering of candidates has ended at the peer, either for the session or only for the corresponding media description as specified above. The receiving agent forwards an end-of-candidates indication to the ICE Agent, which in turn acts as specified in [I-D.ietf-ice-trickle].

9. Content Type 'application/trickle-ice-sdpfrag'

9.1. Overall Description

A application/trickle-ice-sdpfrag body is used exclusively by the 'trickle-ice' Info Package. Other SDP related applications need to define their own media type. The INFO request body uses a subset of the possible SDP lines as defined by the grammar defined in [RFC4566]. A valid body uses only pseudo "m=" lines and certain attributes that are needed and/or useful for trickling candidates. The content adheres to the following grammar.

9.2. Grammar

The grammar of an 'application/trickle-ice-sdpfrag' body is based on the following ABNF [RFC5234]. It specifies the subset of existing SDP attributes, that is needed or useful for trickling candidates.

The grammar uses the indicator for case-sensitivity %s as defined in [RFC7405], but also imports grammars for other SDP attributes that precede the production of [RFC7405]. A sender SHOULD use lower-case for attributes from such earlier grammars, but a receiver MUST treat them case-insensitively.

```

; Syntax
trickle-ice-sdpfrag = session-level-fields
                      pseudo-media-descriptions
session-level-fields = *(session-level-field CRLF)

session-level-field = ice-lite-attribute /
                     ice-pwd-attribute /
                     ice-ufrag-attribute /
                     ice-options-attribute /
                     ice-pacing-attribute /
                     end-of-candidates-attribute /
                     bundle-group-attribute /
                     extension-attribute-fields
                               ; for future extensions

ice-lite-attribute      = %s"a" "=" ice-lite
ice-pwd-attribute       = %s"a" "=" ice-pwd-att
ice-ufrag-attribute     = %s"a" "=" ice-ufrag-att
ice-pacing-attribute    = %s"a" "=" ice-pacing-att
ice-options-attribute   = %s"a" "=" ice-options
end-of-candidates-attribute = %s"a" "=" end-of-candidates
end-of-candidates      = %s"end-of-candidates"
bundle-group-attribute = %s"a" "=" %s"group:" bundle-semantic
                               *(SP identification-tag)
bundle-semantic        = "BUNDLE"
extension-attribute-fields = attribute-fields

pseudo-media-descriptions = *( media-field
                               trickle-ice-attribute-fields )
trickle-ice-attribute-fields = *(trickle-ice-attribute-field CRLF)
trickle-ice-attribute-field = mid-attribute /
                             candidate-attributes /
                             ice-pwd-attribute /
                             ice-ufrag-attribute /
                             remote-candidate-attribute /
                             end-of-candidates-attribute /
                             rtcp-attribute /
                             rtcp-mux-attribute /
                             rtcp-mux-only-attribute /
                             extension-attribute-fields
                                       ; for future extensions

rtcp-attribute          = %s"a" "=" %s"rtcp"
rtcp-mux-attribute      = %s"a" "=" %s"rtcp-mux"
rtcp-mux-only-attribute = %s"a" "=" %s"rtcp-mux-only"
candidate-attributes    = %s"a" "=" candidate-attribute
remote-candidate-attribute = %s"a" "=" remote-candidate-att

```

with ice-lite, ice-pwd-att, remote-candidate-att, ice-ufrag-att, ice-pacing-att, ice-options, candidate-attribute remote-candidate-att from [I-D.ietf-mmusic-ice-sip-sdp], identification-tag, mid-attribute ; from [RFC5888], media-field, attribute-fields from [RFC4566]. The "a=rtcp" attribute is defined in [RFC3605], the "a=rtcp-mux" attribute in [RFC5761] and the "a=rtcp-mux-only" attribute in [I-D.ietf-mmusic-mux-exclusive]. The latter attributes lack a formal grammar in their corresponding RFC and are reproduced here.

The "a=ice-pwd:" and "a=ice-ufrag:" attributes MUST appear at the same level as the ones in the Offer/Answer exchange. In other words, if they were present as session-level attributes, they will also appear at the beginning of all INFO request payloads, i.e. preceding all pseudo "m=" lines. If they were originally exchanged as media level attributes, potentially overriding session-level values, then they will also be included in INFO request payloads following the corresponding pseudo "m=" lines.

An Agent MUST ignore any received unknown extension-attribute-fields.

10. Info Package

10.1. Rationale - Why INFO?

The decision to use SIP INFO requests as a candidate transport method is based primarily on their lightweight nature. Once a dialog has been established, INFO requests can be exchanged both ways with no restrictions on timing and frequency and no risk of collision.

A critical fact is that the sending of Trickle ICE candidates in one direction is entirely uncoupled from sending candidates in the other direction. Thus, the sending of candidates in each direction can be done by a stream of INFO requests that is not correlated with the stream of INFO requests in the other direction. And since each INFO request cumulatively includes the contents of all previous INFO requests in that direction, ordering between INFO requests need not be preserved. All of this permits using largely-independent INFO requests.

Contrarily, UPDATE or other offer/answer mechanisms assume that the messages in each direction are tightly coupled with messages in the other direction. Using Offer/Answer and UPDATE requests [RFC3311] would introduce the following complications:

Blocking of messages: [RFC3264] defines Offer/Answer as a strictly sequential mechanism. There can only be a maximum of one active exchange at any point of time. Both sides cannot simultaneously send Offers nor can they generate multiple Offers prior to

receiving an Answer. Using UPDATE requests for candidate transport would therefore imply the implementation of a candidate pool at every agent where candidates can be stored until it is once again that agent's "turn" to emit an Answer or a new Offer. Such an approach would introduce non-negligible complexity for no additional value.

Elevated risk of glare: The sequential nature of Offer/Answer also makes it impossible for both sides to send Offers simultaneously. What's worse is that there are no mechanisms in SIP to actually prevent that. [RFC3261], where the situation of Offers crossing on the wire is described as "glare", only defines a procedure for addressing the issue after it has occurred. According to that procedure both Offers are invalidated and both sides need to retry the negotiation after a period between 0 and 4 seconds. The high likelihood for glare to occur and the average two second back-off intervals implies that the duration of Trickle ICE processing would not only fail to improve but actually exceed those of regular ICE.

INFO messages decouple the exchange of candidates from the Offer/Answer negotiation and are subject to none of the glare issues described above, which makes them a very convenient and lightweight mechanism for asynchronous delivery of candidates.

Using in-dialog INFO messages also provides a way of guaranteeing that candidates are delivered end-to-end, between the same entities that are actually in the process of initiating a session. Out-of-dialog alternatives would have implied requiring support for Globally Routable UA URI (GRUU) [RFC5627] which, given GRUUs relatively low adoption levels, would have constituted too strong of a constraint to the adoption of Trickle ICE.

10.2. Overall Description

This specification defines an Info Package for use by SIP User Agents implementing Trickle ICE. INFO requests carry ICE candidates discovered after the peer user agents have confirmed mutual support for Trickle ICE.

10.3. Applicability

The purpose of the ICE protocol is to establish a media path in the presence of NAT and firewalls. The candidates are transported in INFO requests and are part of this establishment.

Candidates sent by a Trickle ICE Agent after the Offer, follow the same signaling path and reach the same entity as the Offer itself.

While it is true that GRUUs can be used to achieve this, one of the goals of this specification is to allow operation of Trickle ICE in as many environments as possible including those without GRUU support. Using out-of-dialog SUBSCRIBE/NOTIFY requests would not satisfy this goal.

10.4. Info Package Name

This document defines a SIP Info Package as per [RFC6086]. The Info Package token name for this package is "trickle-ice"

10.5. Info Package Parameters

This document does not define any Info Package parameters.

10.6. SIP Option Tags

[RFC6086] allows Info Package specifications to define SIP option-tags. This specification extends the option-tag construct of the SIP grammar as follows:

```
option-tag /= "trickle-ice"
```

SIP entities that support this specification MUST place the 'trickle-ice' option-tag in a SIP Supported: or Require: header field within all SIP INVITE requests and responses.

When responding to, or generating a SIP OPTIONS request a SIP entity MUST also include the 'trickle-ice' option-tag in a SIP Supported: or Require: header field.

10.7. Info Request Body Parts

Entities implementing this specification MUST include a payload of type 'application/trickle-ice-sdpfrag' as defined in Section 9.2 in SIP INFO requests. The payload is used to convey SDP-encoded ICE candidates.

10.8. Info Package Usage Restrictions

This document does not define any Info Package Usage Restrictions.

10.9. Rate of INFO Requests

Given that IP addresses may be gathered rapidly a Trickle ICE Agent with many network interfaces might create a high rate of INFO requests if every newly detected candidate is trickled individually without aggregation. An implementation MUST aggregate ICE candidates

in case that an unreliable transport protocol such as UDP is used. A Trickle ICE agent MUST NOT have more than one INFO request pending at any one time. When INFO messages are sent over an unreliable transport, they are retransmitted according to the rules specified in [RFC3261] section 17.1.2.1."

If the INFO requests are sent on top of TCP, which is probably the standard way, this is not an issue for the network anymore, but it can remain one for SIP proxies and other intermediaries forwarding the SIP INFO messages. Also, an endpoint may not be able to tell that it has congestion controlled transport all the way.

10.10. Info Package Security Considerations

See Section 13

11. Deployment Considerations

Trickle ICE uses two mechanisms for exchange of candidate information. This imposes new requirements to certain middleboxes that are used in some networks, e.g. for monitoring purposes. While the first mechanism, SDP Offers and Answers, is already used by regular ICE and is assumed to be supported, the second mechanism, INFO request bodies, needs to be considered by such middleboxes as well when trickle ICE is used. Such middleboxes need to make sure that they remain in the signaling path of the INFO requests and need to understand the INFO request body.

12. IANA Considerations

[RFC EDITOR NOTE: Please replace RFCXXXX with the RFC number of this document.]

12.1. SDP 'end-of-candidates' Attribute

This section defines a new SDP media-level and session-level attribute [RFC4566] , 'end-of-candidates'. 'end-of-candidates' is a property attribute [RFC4566] , and hence has no value.

Name: end-of-candidates

Value: N/A

Usage Level: media and session

Charset Dependent: no

Purpose: The sender indicates that it will not trickle further ICE candidates.

O/A Procedures: RFCXXX defines the detailed SDP Offer/Answer procedures for the 'end-of-candidates' attribute.

Mux Category: IDENTICAL

Reference: RFCXXXX

Example:

a=end-of-candidates

12.2. Media Type 'application/trickle-ice-sdpfrag'

This document defines a new Media Type 'application/trickle-ice-sdpfrag' in accordance with [RFC6838].

Type name: application

Subtype name: trickle-ice-sdpfrag

Required parameters: None.

Optional parameters: None.

Encoding considerations:

The media contents follow the same rules as SDP, except as noted in this document. The media contents are text, with the grammar specified in Section 9.2.

Although the initially defined content of a trickle-ice-sdpfrag body does only include ASCII characters, UTF-8 encoded content might be introduced via extension attributes. The "a=charset:"

attribute may be used to signal the presence of other character sets in certain parts of a trickle-ice-sdpfrag body (see [RFC4566]). Arbitrary binary content cannot be directly represented in SDP or a trickle-ice-sdpfrag body.

Security considerations:

See [RFC4566] and RFCXXXX

Interoperability considerations:

See RFCXXXX

Published specification:

See RFCXXXX

Applications which use this Media Type:

Trickle-ICE

Fragment identifier considerations: N/A

Additional information:

Deprecated alias names for this type: N/A

Magic number(s): N/A

File extension(s): N/A

Macintosh File Type Code(s): N/A

Person and email address to contact for further information:

The IESG (iesg@ietf.org)

Intended usage:

Trickle-ICE for SIP as specified in RFCXXXX.

Restrictions on usage: N/A

Author/Change controller:

The IESG (iesg@ietf.org)

Provisional registration? (standards tree only): N/A

12.3. SIP Info Package 'trickle-ice'

This document defines a new SIP Info Package named 'trickle-ice' and updates the Info Packages Registry with the following entry.

Name	Reference
trickle-ice	[RFCXXXX]

12.4. SIP Option Tag 'trickle-ice'

This specification registers a new SIP option tag 'trickle-ice' as per the guidelines in Section 27.1 of [RFC3261] and updates the "Option Tags" section of the SIP Parameter Registry with the following entry:

Name	Description	Reference
trickle-ice	This option tag is used to indicate that a UA supports and understands Trickle-ICE.	[RFCXXXX]

13. Security Considerations

The Security Considerations of [I-D.ietf-mmusic-ice-sip-sdp], [RFC6086] and [I-D.ietf-ice-trickle] apply. This document clarifies

how the above specifications are used together for trickling candidates and does not create additional security risks.

The new Info Package 'trickle-ice' and the new Media Type 'application/trickle-ice-sdpfrag' do not introduce additional security considerations when used in the context of Trickle ICE. Both are not intended to be used for other applications, so any security considerations for its use in other contexts is out of the scope of this document

14. Acknowledgements

The authors like to thank Flemming Andreasen, Ayush Jain, Paul Kyzivat, Jonathan Lennox, Simon Perreault, Roman Shpount and Martin Thomson for reviewing and/or making various suggestions for improvements and optimizations.

The authors also like to thank Flemming Andreasen for shepherding this document and Ben Campbell for his AD review and suggestions. In addition, the author like to thank Benjamin Kaduk, Adam Roach, Mirja Kuehlewind and Eric Rescorla for their comments and/or text proposals for improving the document during IESG review.

Many thanks to Dale Worley for Gen-Art review and proposed enhancements for several sections.

Many thanks to Joerg Ott for TSV-Art review and suggested improvements.

The authors thank Shawn Emery for Security Directorate review.

15. Change Log

[RFC EDITOR NOTE: Please remove this section when publishing].

Changes from draft-ietf-mmusic-trickle-ice-sip-01

- o Editorial Clean up
- o IANA Consideration added
- o Security Consideration added
- o RTCP and BUNDLE Consideration added with rules for including "a=rtcp-mux" and "a=group: BUNDLLE" attributes
- o 3PCC Consideration added

- o Clarified that 18x w/o answer is sufficient to create a dialog that allows for trickling to start
- o Added remaining Info Package definition sections as outlined in section 10 of [RFC6086]
- o Added definition of application/sdpfrag making draft-ivov-mmusic-sdpfrag obsolete
- o Added pseudo m-lines as additional separator in sdpfrag bodies for Trickle ICE
- o Added ABNF for sdp-frag bodies and Trickle-ICE package

Changes from draft-ietf-mmusic-trickle-ice-sip-02

- o Removed definition of application/sdpfrag
- o Replaced with new type application/trickle-ice-sdpfrag
- o RTCP and BUNDLE Consideration enhanced with some examples
- o draft-ietf-mmusic-sdp-bundle-negotiation and RFC5761 changed to normative reference
- o Removed reference to 4566bis
- o Addressed review comment from Simon Perreault

Changes from draft-ietf-mmusic-trickle-ice-sip-03

- o replaced reference to RFC5245 with draft-ietf-mmusic-rfc5245bis and draft-ietf-mmusic-ice-sip-sdp
- o Corrected Figure 10, credits to Ayush Jain for finding the bug
- o Referencing a=rtcp and a=rtcp-mux handling from draft-ietf-mmusic-ice-sip-sdp
- o Referencing a=rtcp-mux-exclusive handling from draft-ietf-mmusic-mux-exclusive, enhanced ABNF to support a=rtcp-mux-exclusive
- o Clarifying that draft-ietf-mmusic-sdp-mux-attributes applies for the application/trickle-ice-sdpfrag body

Changes from draft-ietf-mmusic-trickle-ice-sip-04

- o considered comments from Christer Holmberg

- o corrected grammar for INFO package, such that ice-ufrag/pwd are also allowed on media-level as specified in [I-D.ietf-mmusic-ice-sip-sdp]
- o Added new ice-pacing-attribute fom [I-D.ietf-mmusic-ice-sip-sdp]
- o Added formal definition for the end-of-candidates attribute

Changes from draft-ietf-mmusic-trickle-ice-sip-05

- o considered further comments from Christer Holmberg
- o editorial comments on section 3 addressed
- o moved section 3.1 to section 10.1 and applied some edits
- o replaced the term "previously sent candidates" with "currently known and used candidates".

Changes from draft-ietf-mmusic-trickle-ice-sip-06

- o editorial fixes
- o additional text on the content of the INFO messages.
- o recommendation on what to do if a previously sent candidate is unexpectedly missing in a subsequent INFO
- o terminology alignment with draft-ietf-ice-trickle-07

Changes from draft-ietf-mmusic-trickle-ice-sip-07

- o editorial fixes
- o clarification on ordering of candidates for alignment with draft-ietf-ice-trickle-12
- o O/A procedures for end-of-candidates attribute described here after corresponding procedures have been removed from draft-ietf-ice-trickle-11
- o using IPv6 addresses in examples

Changes from draft-ietf-mmusic-trickle-ice-sip-08

- o editorial fixes/clarification based on Flemmings review

- o Description of Trickle specifics in O/A procedures for initial O/A exchange and specification of ICE mismatch exception

Changes from draft-ietf-mmusic-trickle-ice-sip-09

- o editorial fixes/correction of references
- o adding missing Ref to RFC3605 in section 6, 5th para
- o replaced remaining IPv4 addresses with IPv6
- o Added text for handling a=rtcp in case of default RTP address 0.0.0.0:9 based on comment from Roman Shpount.

Changes from draft-ietf-mmusic-trickle-ice-sip-10

- o editorial fixes due to idnits output

Changes from draft-ietf-mmusic-trickle-ice-sip-11

- o addressing comments from Ben Campell's AD review and Christer's review
- o Numerous editorial improvements/corrections
- o Added [RFC8174] boiler plate and adapted usage of normative language
- o Clarified terminology ICE modules .vs. ICE agent
- o Added more detailed OA procedures
- o Corrected default values in m-line and usage of "a=mid:" attribute explicitly mentioned for offer/answer
- o Removed explicit mentioning of XMPP
- o Added Deployment Considerations section
- o Fixed ref for rfc5245bis

Changes from draft-ietf-mmusic-trickle-ice-sip-12

- o addressing comments from Gen-Art review, TSV-Art review and Security Directorate review
- o Numerous editorial improvements/corrections/clarifications

Changes from draft-ietf-mmusic-trickle-ice-sip-13

- o added expansions for SDP, GRUU, AOR, STUN, TURN
- o some editorial corrections

Changes from draft-ietf-mmusic-trickle-ice-sip-14

Addressing comments from IESG review

- o Clarification/enhancement in section 5 and Fig. 10 based on comments from Benjamin Kaduk
- o Clarification on sequence for sending candidates, definition of pseudo m-lines, usage of a=mid attribute, usage of INFO as ACK for receipt of 18x based on comments from Eric Rescorla
- o Removal of 3PCC Section 3.4, removal of NATted IPv6 addresses, adding more flexibility to in the grammar, explicit mentioning of Require: header field, usage of Require: header field in case of provisioning, text on repetition of candidates in case of RTCP mux and Bundle, various other editorial improvements/corrections based on comments from Adam Roach
- o Modified text on rate limitation of INFO requests based on comments of Mirja Kuehlewind, Adam Roach and Roman Shpount
- o some editorial corrections

Changes from draft-ietf-mmusic-trickle-ice-sip-15

- o Corrections in section 7 on Media Multiplexing

Changes from draft-ietf-mmusic-trickle-ice-sip-16

- o some editorial corrections

Changes from draft-ietf-mmusic-trickle-ice-sip-16

- o Changed IPv6 candidate example from srflx to host

16. References

16.1. Normative References

- [I-D.ietf-ice-rfc5245bis]
Keranen, A., Holmberg, C., and J. Rosenberg, "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal", draft-ietf-ice-rfc5245bis-20 (work in progress), March 2018.
- [I-D.ietf-ice-trickle]
Ivov, E., Rescorla, E., Uberti, J., and P. Saint-Andre, "Trickle ICE: Incremental Provisioning of Candidates for the Interactive Connectivity Establishment (ICE) Protocol", draft-ietf-ice-trickle-21 (work in progress), April 2018.
- [I-D.ietf-mmusic-ice-sip-sdp]
Petit-Huguenin, M., Nandakumar, S., and A. Keranen, "Session Description Protocol (SDP) Offer/Answer procedures for Interactive Connectivity Establishment (ICE)", draft-ietf-mmusic-ice-sip-sdp-20 (work in progress), April 2018.
- [I-D.ietf-mmusic-mux-exclusive]
Holmberg, C., "Indicating Exclusive Support of RTP/RTCP Multiplexing using SDP", draft-ietf-mmusic-mux-exclusive-12 (work in progress), May 2017.
- [I-D.ietf-mmusic-sdp-bundle-negotiation]
Holmberg, C., Alvestrand, H., and C. Jennings, "Negotiating Media Multiplexing Using the Session Description Protocol (SDP)", draft-ietf-mmusic-sdp-bundle-negotiation-52 (work in progress), May 2018.
- [I-D.ietf-mmusic-sdp-mux-attributes]
Nandakumar, S., "A Framework for SDP Attributes when Multiplexing", draft-ietf-mmusic-sdp-mux-attributes-17 (work in progress), February 2018.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://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, <<https://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, <<https://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, <<https://www.rfc-editor.org/info/rfc3264>>.
- [RFC3605] Huitema, C., "Real Time Control Protocol (RTCP) attribute in Session Description Protocol (SDP)", RFC 3605, DOI 10.17487/RFC3605, October 2003, <<https://www.rfc-editor.org/info/rfc3605>>.
- [RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, DOI 10.17487/RFC4566, July 2006, <<https://www.rfc-editor.org/info/rfc4566>>.
- [RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <<https://www.rfc-editor.org/info/rfc5234>>.
- [RFC5761] Perkins, C. and M. Westerlund, "Multiplexing RTP Data and Control Packets on a Single Port", RFC 5761, DOI 10.17487/RFC5761, April 2010, <<https://www.rfc-editor.org/info/rfc5761>>.
- [RFC5888] Camarillo, G. and H. Schulzrinne, "The Session Description Protocol (SDP) Grouping Framework", RFC 5888, DOI 10.17487/RFC5888, June 2010, <<https://www.rfc-editor.org/info/rfc5888>>.
- [RFC6086] Holmberg, C., Burger, E., and H. Kaplan, "Session Initiation Protocol (SIP) INFO Method and Package Framework", RFC 6086, DOI 10.17487/RFC6086, January 2011, <<https://www.rfc-editor.org/info/rfc6086>>.
- [RFC6838] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", BCP 13, RFC 6838, DOI 10.17487/RFC6838, January 2013, <<https://www.rfc-editor.org/info/rfc6838>>.
- [RFC7405] Kyzivat, P., "Case-Sensitive String Support in ABNF", RFC 7405, DOI 10.17487/RFC7405, December 2014, <<https://www.rfc-editor.org/info/rfc7405>>.

- [RFC8085] Eggert, L., Fairhurst, G., and G. Shepherd, "UDP Usage Guidelines", BCP 145, RFC 8085, DOI 10.17487/RFC8085, March 2017, <<https://www.rfc-editor.org/info/rfc8085>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

16.2. Informative References

- [RFC3311] Rosenberg, J., "The Session Initiation Protocol (SIP) UPDATE Method", RFC 3311, DOI 10.17487/RFC3311, October 2002, <<https://www.rfc-editor.org/info/rfc3311>>.
- [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, <<https://www.rfc-editor.org/info/rfc3725>>.
- [RFC3840] Rosenberg, J., Schulzrinne, H., and P. Kyzivat, "Indicating User Agent Capabilities in the Session Initiation Protocol (SIP)", RFC 3840, DOI 10.17487/RFC3840, August 2004, <<https://www.rfc-editor.org/info/rfc3840>>.
- [RFC5389] Rosenberg, J., Mahy, R., Matthews, P., and D. Wing, "Session Traversal Utilities for NAT (STUN)", RFC 5389, DOI 10.17487/RFC5389, October 2008, <<https://www.rfc-editor.org/info/rfc5389>>.
- [RFC5627] Rosenberg, J., "Obtaining and Using Globally Routable User Agent URIs (GRUUs) in the Session Initiation Protocol (SIP)", RFC 5627, DOI 10.17487/RFC5627, October 2009, <<https://www.rfc-editor.org/info/rfc5627>>.
- [RFC5766] Mahy, R., Matthews, P., and J. Rosenberg, "Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN)", RFC 5766, DOI 10.17487/RFC5766, April 2010, <<https://www.rfc-editor.org/info/rfc5766>>.

Authors' Addresses

Emil Ivov
Jitsi
Strasbourg 67000
France

Phone: +33 6 72 81 15 55
Email: emcho@jitsi.org

Thomas Stach
Unaffiliated
Vienna 1130
Austria

Email: thomass.stach@gmail.com

Enrico Marocco
Telecom Italia
Via G. Reiss Romoli, 274
Turin 10148
Italy

Email: enrico.marocco@telecomitalia.it

Christer Holmberg
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

Email: christer.holmberg@ericsson.com