

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: December 19, 2014

E. Iovov  
Jitsi  
E. Marocco  
Telecom Italia  
C. Holmberg  
Ericsson  
June 17, 2014

A Session Initiation Protocol (SIP) usage for Trickle ICE  
draft-iovov-mmusic-trickle-ice-sip-02

## 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).

## 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 December 19, 2014.

## Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

Internet-Draft

Trickle ICE for SIP

June 2014

This document is subject to [BCP 78](http://trustee.ietf.org/license-info) 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.

## Table of Contents

<a href="#">1.</a>	<a href="#">Introduction</a>	<a href="#">2</a>
<a href="#">2.</a>	<a href="#">Terminology</a>	<a href="#">3</a>
<a href="#">3.</a>	<a href="#">Protocol Overview</a>	<a href="#">3</a>
<a href="#">3.1.</a>	<a href="#">Rationale. Why INFO</a>	<a href="#">4</a>
<a href="#">3.2.</a>	<a href="#">Discovery issues</a>	<a href="#">5</a>
<a href="#">3.3.</a>	<a href="#">Relationship with the Offer/Answer Model</a>	<a href="#">6</a>
<a href="#">4.</a>	<a href="#">Incremental signalling of ICE candidates</a>	<a href="#">8</a>
<a href="#">4.1.</a>	<a href="#">Asserting Offer/Answer delivery and dialog state</a>	<a href="#">8</a>
<a href="#">4.2.</a>	<a href="#">Delivering candidates in INFO messages</a>	<a href="#">12</a>
<a href="#">5.</a>	<a href="#">Initial discovery of trickle ICE support</a>	<a href="#">14</a>
<a href="#">5.1.</a>	<a href="#">Provisioning support for trickle ICE</a>	<a href="#">14</a>
<a href="#">5.2.</a>	<a href="#">Trickle ICE discovery with GRUU</a>	<a href="#">15</a>
<a href="#">5.3.</a>	<a href="#">Trickle ICE discovery through other protocols</a>	<a href="#">16</a>
<a href="#">5.4.</a>	<a href="#">Fallback to half trickle</a>	<a href="#">16</a>
<a href="#">6.</a>	<a href="#">Info Package</a>	<a href="#">18</a>
<a href="#">6.1.</a>	<a href="#">Overall Description</a>	<a href="#">18</a>
<a href="#">6.2.</a>	<a href="#">Applicability</a>	<a href="#">18</a>
<a href="#">6.3.</a>	<a href="#">Info Package Name</a>	<a href="#">18</a>
<a href="#">6.4.</a>	<a href="#">Info Package Parameters</a>	<a href="#">18</a>
<a href="#">6.5.</a>	<a href="#">SIP Option-Tags</a>	<a href="#">18</a>
<a href="#">6.6.</a>	<a href="#">Info Message Body Parts</a>	<a href="#">19</a>
<a href="#">7.</a>	<a href="#">Security Considerations</a>	<a href="#">19</a>
<a href="#">8.</a>	<a href="#">Acknowledgements</a>	<a href="#">19</a>
<a href="#">9.</a>	<a href="#">References</a>	<a href="#">19</a>
<a href="#">9.1.</a>	<a href="#">Normative References</a>	<a href="#">19</a>
<a href="#">9.2.</a>	<a href="#">Informative References</a>	<a href="#">20</a>
<a href="#">Appendix A.</a>	<a href="#">Open issues</a>	<a href="#">20</a>
	<a href="#">Authors' Addresses</a>	<a href="#">21</a>

## [1.](#) Introduction

The vanilla specification of the Interactive Connectivity Establishment (vanilla ICE) protocol [[RFC5245](#)] describes a mechanism for NAT traversal that consists of three main phases: a phase where an agent gathers a set of candidate transport addresses (source IP address, port and transport protocol), a second phase where these

candidates are sent to a remote agent and this gathering procedure is repeated and, finally, a third phase where connectivity between all candidates in both sets is checked (connectivity checks). Once these phases have been completed, and only then, can both agents begin communication. According to the Vanilla ICE specification the three phases above happen consecutively, in a blocking way, which may lead to undesirable latency during session establishment.

The trickle ICE extension defined in [[I-D.ietf-mmusic-trickle-ice](#)] defines generic semantics required for these ICE phases to happen simultaneously, in a non-blocking way and hence speed up session establishment.

The present specification defines a usage of trickle ICE with the Session Initiation Protocol (SIP). It describes how ICE candidates are to be incrementally exchanged with SIP INFO requests and how the half and full-trickle modes defined in [[I-D.ietf-mmusic-trickle-ice](#)] 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 [[RFC6086](#)] for use with Trickle ICE.

## [2.](#) Terminology

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

This specification makes use of all terminology defined by the protocol for Interactive Connectivity Establishment in [[RFC5245](#)] and its Trickle ICE extension [[I-D.ietf-mmusic-trickle-ice](#)]. It is assumed that the reader will be familiar with the terminology from both of them.

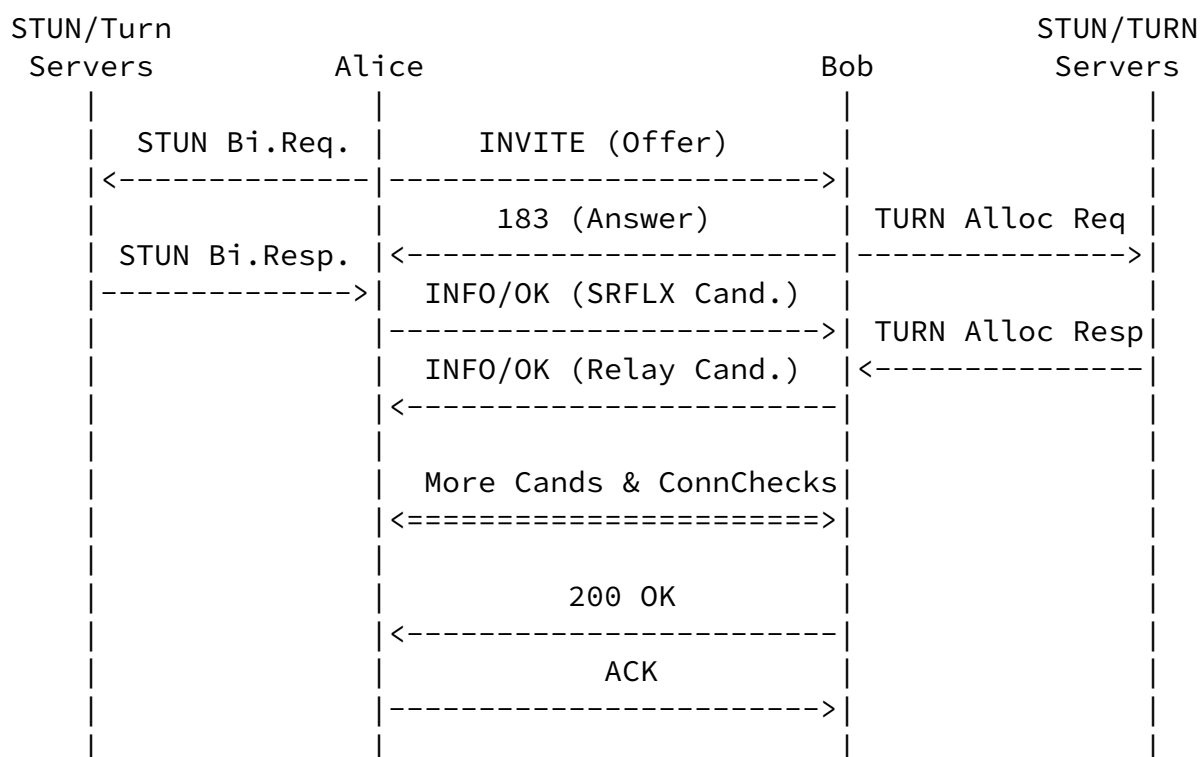
## [3.](#) Protocol Overview

The semantics that vanilla ICE [[RFC5245](#)] defines for exchanging ICE candidates are exclusively based on use of Offers and Answers as per [[RFC3264](#)] over the Session Description Protocol (SDP) [[RFC4566](#)]. This specification extends these mechanism by allowing ICE candidates to also be sent after the completion of Offer/Answer negotiation, through the use of SIP INFO messages and a newly defined Info Package [[RFC6086](#)].

Specifically, in cases where trickle ICE is fully supported, a typical exchange would happen along the following lines: The offerer would send an INVITE containing a subset of candidates and then wait

for an early dialog to be established. Once that happens, it will be able to continue sending candidates through in INFO requests and within the same dialog.

Similarly, once it has sent an answer (though not earlier) an answerer can continue "trickling" ICE candidates using INFO messages within the dialog established by its 18x provisional response. Figure 1 shows such a sample exchange:



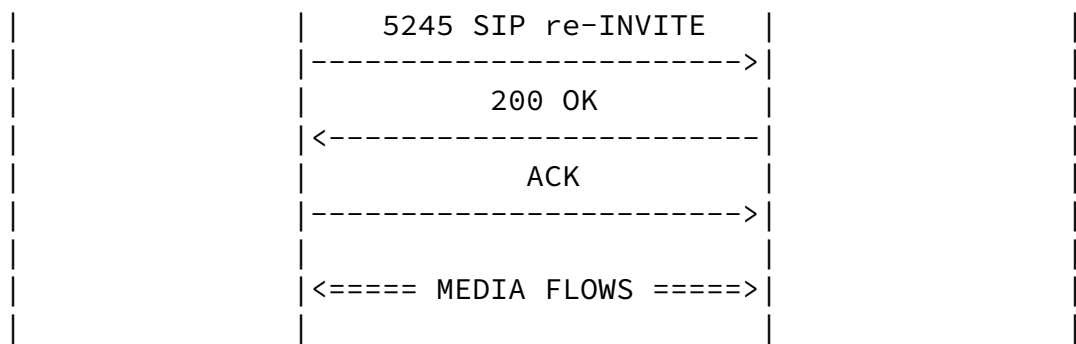


Figure 1: Sample trickle ICE scenario with SIP

### [3.1.](#) 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 messages can be exchanged both ways with no restrictions on timing and frequency and no risk of collision.

On the other hand, using offer/answer and UPDATE requests, which from an [\[RFC5245\]](#) perspective is the traditional way of transporting ICE candidates, introduces the following complications:

Need for a turn-based mechanism: [\[RFC3264\]](#) defines Offer/Answer as a strictly sequential mechanism. There can only be a maximum of one exchange at any point of time. Both sides cannot simultaneously send offers nor can they generate multiple offers prior to receiving an answer. Using UPDATES 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 backoff intervals would imply trickle ICE processing duration would not only fail to improve but actually exceed those of vanilla ICE.

INFO messages have no impact on 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. The alternative 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 constraint to the adoption of trickle ICE.

### 3.2. Discovery issues

In order for 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 actually using it.

Contrary to other protocols, like XMPP [[RFC6120](#)], where "in advance" capability discovery is widely implemented, the mechanisms that allow this for sip (i.e., a combination of UA Capabilities [[RFC3840](#)] and 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.

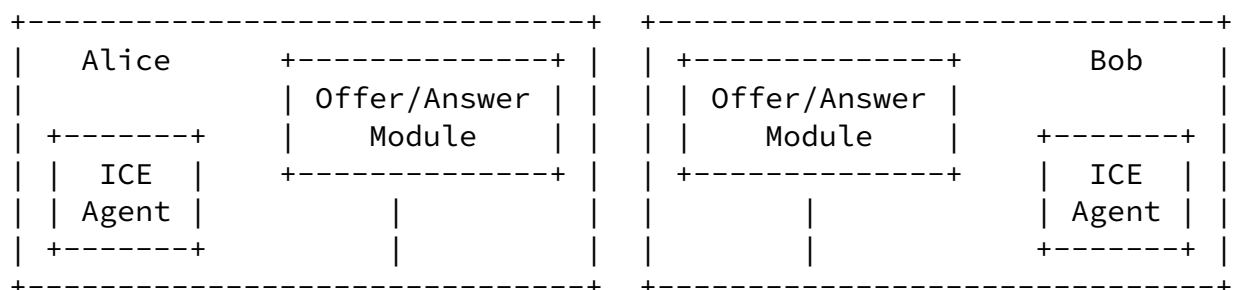
The "half trickle" mode of operation defined in the trickle ICE specification [[I-D.ietf-mmusic-trickle-ice](#)] provides one way around this, by requiring first offers to contain a complete set of ICE candidates and only using incremental provisioning for the rest of

the sessions.

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 5.4](#) discusses the implementation and use of "half trickle" by SIP UAs where none of the above are an option.

### [3.3](#). Relationship with the Offer/Answer Model

It is important to note that this specification does not require, define, or even assume any mechanisms that would have an impact on the Offer/Answer model beyond the way it is already used by vanilla ICE [[RFC5245](#)]. From the perspective of all SIP middle boxes and proxies, and with the exception of the actual INFO messages, signalling in general and Offer/Answer exchanges in particular would look the same way for trickle as they would for vanilla ICE.



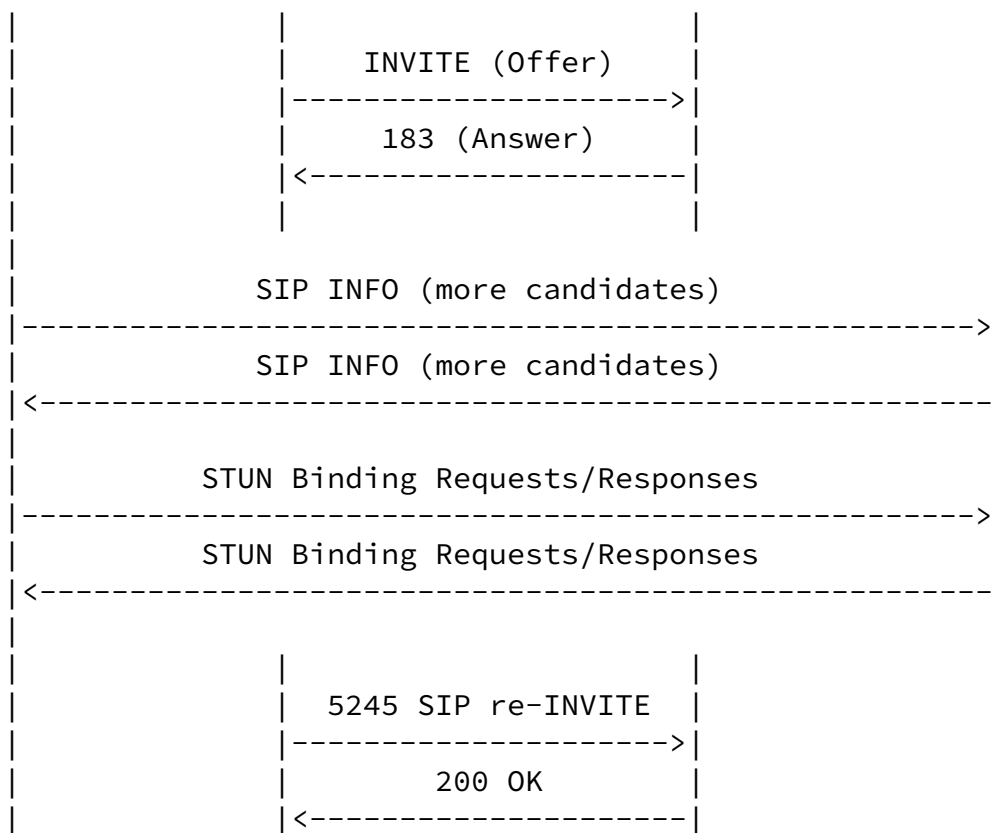


Figure 2: Distinguishing between trickle ICE and traditional signalling.

It is important to note that, as displayed on Figure 2, exchanging candidates through SIP INFO messages are best represented as signalling between ICE agents and not between the traditional SIP and Offer/Answer modules of SIP User Agents. Such INFO requests do not impact the state of Offer/Answer, nor do they have an impact on the version number in the "o=" line. In that regard they are actually comparable to Peer Reflexive candidates that ICE agents can discover during ICE processing.



Trickle ICE agents will construct offers and answers the same way a [\[RFC5245\]](#) compatible agent would, with the following two main differences:

- o First, all offers and answers sent by the trickle ICE capable agents MUST indicate support for trickle ICE through the "trickle" ice-options tag defined in [\[I-D.ietf-mmusic-trickle-ice\]](#):

a=ice-options:trickle

- o Second, offers and answers may contain all, some, or no ICE candidates at all.

Once an agent has sent an offer or an answer indicating support for trickle ICE, it MUST be prepared to receive SIP INFO requests within that same dialog, containing additional candidates or an indication for the end of such candidates.

Similarly, as soon as a SIP UA has established a dialog (including an early state one) it MAY begin sending INFO requests containing additional candidates or end-of-candidates indications. Such INFO requests MUST be sent within that same dialog. This is necessary so that the candidates from these INFO messages could be delivered to the same entities that initiated the session.

#### [4.1.](#) Asserting Offer/Answer delivery and dialog state

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

- o ICE support in the peer agent MUST be confirmed.
- o The SIP dialog at both sides MUST be at least in the early state.

[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 needs to happen anyway for ICE processing to commence (see Figure 3).

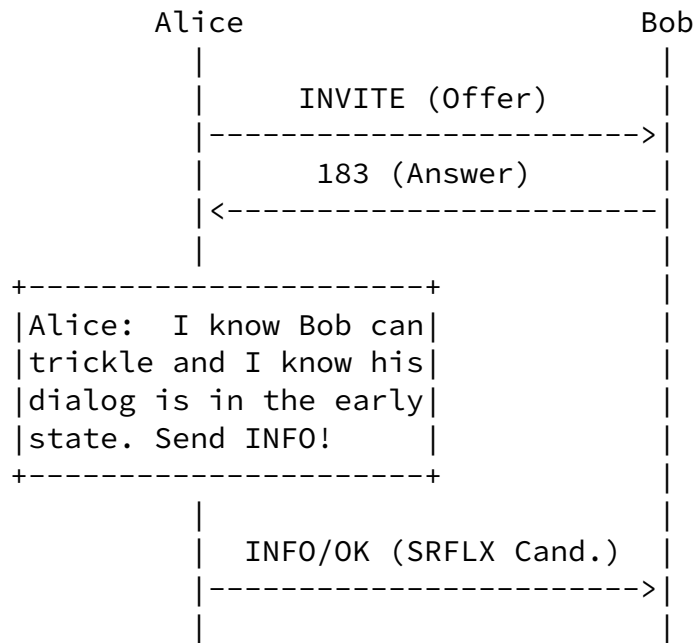


Figure 3: SIP Offerer can freely trickle as soon as it receives an answer.

Satisfying both conditions is also relatively trivial for ICE agents that have sent an offer in an INVITE and that have received an answer. Regardless of how that answer was delivered, it is guaranteed to have confirmed support for trickle ICE within the answerer (or lack thereof) and to have fully initialized the SIP dialog at both ends. Offerers in the above situation can therefore freely commence trickling within the newly established dialog (see Figure 4).

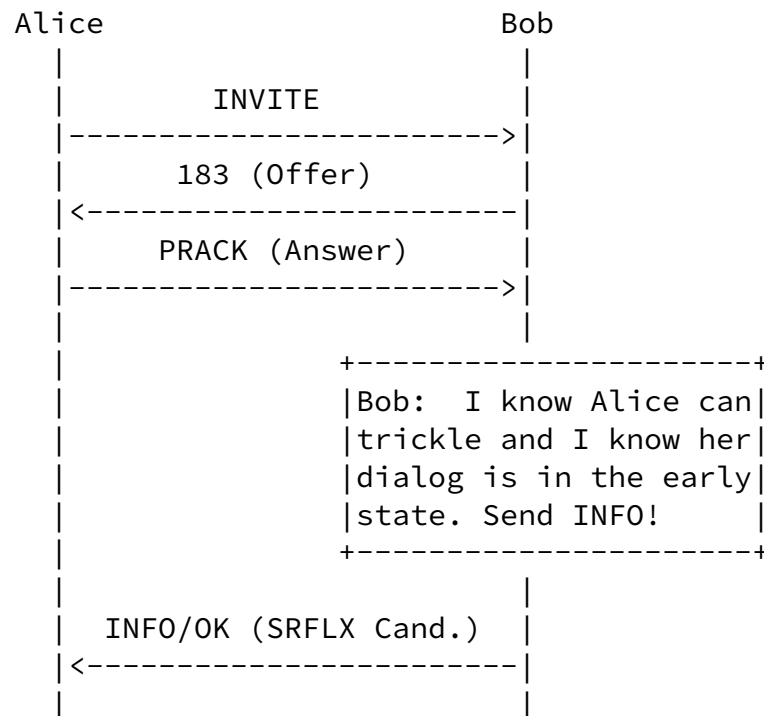


Figure 4: A SIP Offerer in a 3PCC scenario can also freely start trickling as soon as it receives an answer.

Agents that have sent an offer in a reliable provisional response or in a 200 OK and that receive an answer in a PRACK or in an ACK are also in a similar situation because, by the time the offer and the answer are exchanged, support for trickle ICE will be confirmed and the SIP dialog is guaranteed to be in a state that would allow in-dialog INFO requests.

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, there is no way for the answerer to know when or if it has been received (Figure 5).

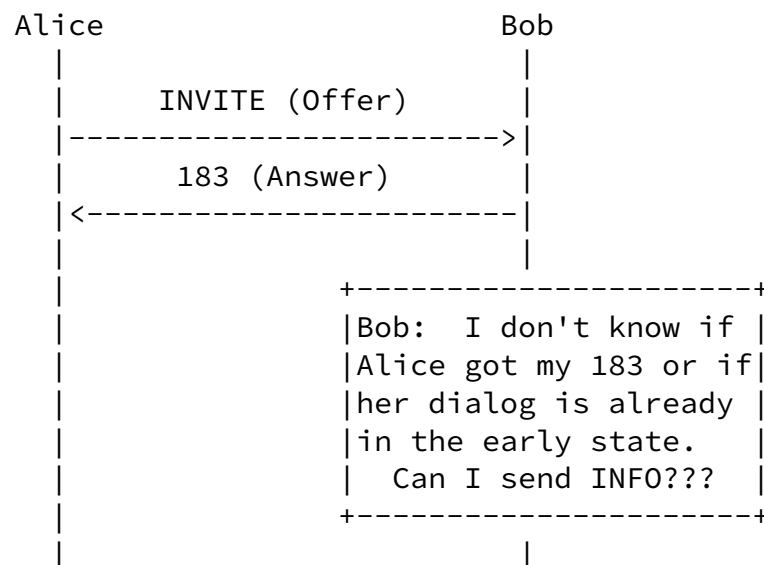


Figure 5: A SIP UA that has answer-ed in an unreliable provisional response cannot know exactly when it is received and when the dialog at the side of the receiver has entered the early state

In order to clear this ambiguity as soon as possible, trickle ICE SIP UAs MUST send a trickle ICE INFO request as soon as they receive an SDP Answer in an unreliable provisional response. This INFO message can only contain the candidates that were already provided in the offer (as would be the case when half trickle is performed or when no new candidates have been learned since then) or they can also deliver new information, such as new candidates (if available) or an end-of-candidates indication in case candidate discovery has ended in the mean time.

As soon as answerers have received such INFO requests, they would have an indication that a dialog is well established at both ends and

they MAY begin trickling (Figure 6).

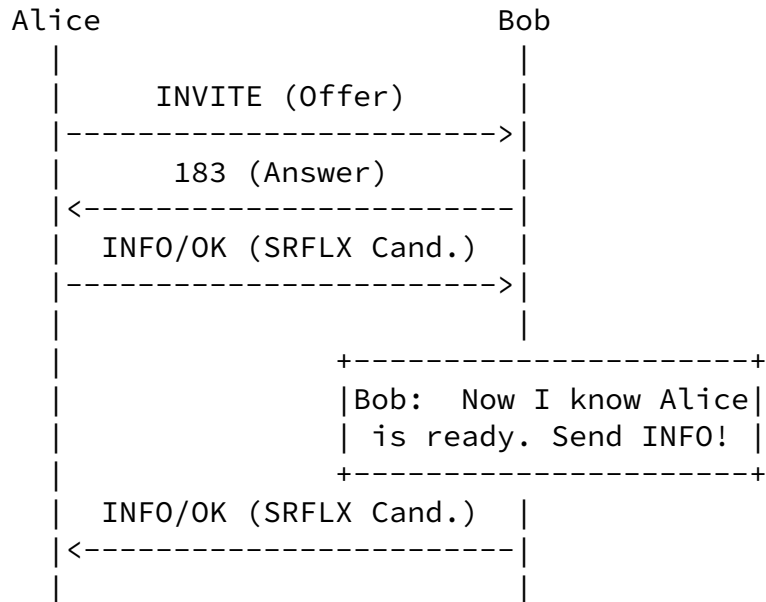


Figure 6: A SIP UA that has answered in an unreliable provisional response cannot know exactly when it is received and when the dialog at the side of the receiver has entered the early state

Obviously, if PRACK [[RFC3262](#)] requests are supported and used, there is no need for the above as the PRACKs themselves would provide sufficient indication for the state of the dialog.

## [4.2.](#) Delivering candidates in INFO messages

Whenever new ICE candidates become available for sending, agents would encode them in "a=candidate" lines as described by [\[I-D.ietf-mmusic-trickle-ice\]](#). For example:

```
a=candidate:2 1 UDP 1694498815 192.0.2.3 5000 typ srflx
      raddr 10.0.1.1 rport 8998
```

The use of SIP INFO requests happens within the context of an Info Package specifically defined for the purpose ([Section 6](#)).

Such INFO requests MUST be sent within the existing SIP dialog. The MIME type for their payload MUST be set to 'application/sdpfrag' as defined in [\[I-D.ivov-dispatch-sdpfrag\]](#).

Since neither the "a=candidate" nor the "a=end-of-candidates" lines contain information that would allow correlating them to a specific "m=" line, this is handled through the use of MID [\[RFC3388\]](#). Agents MUST include the corresponding "a=mid" line for every "m=" line whose candidate list they intend to update. Such "a=mid" lines MUST

immediately precede the list of candidates for that specific "m=" line. All "a=candidate" or "a=end-of-candidates" lines following an "a=mid" line, up until (and excluding) the next occurrence of an "a=mid" line, pertain to the "m=" line identified by that MID. "a=end-of-candidates" lines preceding any "a=mid" lines indicate end of all trickling from that agent (as opposed to end of trickling for a specific "m=" line).

The use of "a=mid" lines allows for a structure similar to the one in SDP offers and answers where one can distinguish separate media-level and session-level sections. In the current case lines preceding any "a=mid" lines are considered to be session-level. Lines appearing in between or after "a=mid" lines will be interpreted as media-level.

All INFO requests MUST carry the "ice-pwd" and "ice-ufrag" attributes that would allow mapping them to a specific ICE generation. INFO requests containing ice-ufrag and ice-pwd values that do not match those of the current ICE processing session MUST be discarded. The

"ice-pwd" and "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 there, they will also appear at the beginning of all INFO message payloads, preceding all "a=mid" lines. If they were originally exchanged as media level attributes, potentially overriding session-level values, then they will also be included in INFO message payloads, following the corresponding "a=mid" line.

In every INFO request agents MUST include all local candidates they have previously signalled. This is necessary in order to more easily avoid problems that would arise from misordering and unreliability.

When receiving INFO requests carrying any candidates, agents will therefore first identify and discard the SDP lines containing candidates they have already received in previous INFO requests or in the Offer/Answer exchange preceding them. Two candidates are considered to be equal if their IP address port, transport and component ID are the same. After identifying and discarding known candidates, agents will then process the remaining ones (the actual new candidates) according to the rules described in [\[I-D.ietf-mmusic-trickle-ice\]](#).

The following example shows the content of one sample candidate delivering INFO request:

```
INFO sip:alice@example.com SIP/2.0
```

```
...
```

```
Info-Package: trickle-ice
```

```
Content-type: application/sdp
```

```
Content-Disposition: Info-Package
```

```
Content-length: ...
```

```
a=ice-pwd:asd88fgpdd777uzjYhagZg
```

```
a=ice-ufrag:8hhY
```

```
a=mid:1
```

```
a=candidate:1 1 UDP 1658497328 192.168.100.33 5000 typ host
```

```
a=candidate:2 1 UDP 1658497328 96.1.2.3 5000 typ srflx
      raddr 10.0.1.1 rport 8998
a=mid:2
a=candidate:2 1 UDP 1658497328 96.1.2.3 5002 typ srflx
      raddr 10.0.1.1 rport 9000
a=end-of-candidates
```

## [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-mmusic-trickle-ice](#)] to indicate that in their offers and answers using the following attribute: "a=ice-options:trickle". This makes discovery fairly straightforward for answerers or for cases where offers need to be generated within existing dialogs (i.e., when sending re-INVITE requests). In both scenarios prior SDP would have provided the necessary information.

Obviously, prior SDP 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 likely to be the case, for example, for WebRTC clients that will always be communicating with other WebRTC clients or known Session Border Controllers (SBC) with support for this specification.

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.

### [5.2.](#) Trickle ICE discovery with GRUU

[RFC3840] provides a way for SIP user agents to query for support of specific capabilities using, among others, OPTIONS requests. GRUU



support 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 6.5](#) provides SIP UAs with a way of learning the capabilities of specific US instances and then addressing them directly with INVITE requests that require SIP support.

Such targeted trickling may happen in different ways. One option would be for a SIP UA to learn the GRUU instance ID of a peer through presence and to then query its capabilities direction with an OPTIONS request. Alternately, it can also just send an OPTIONS request to the AOR it intends to contact and then inspect the returned response(s) for support of both GRUU and trickle ICE (Figure 7).

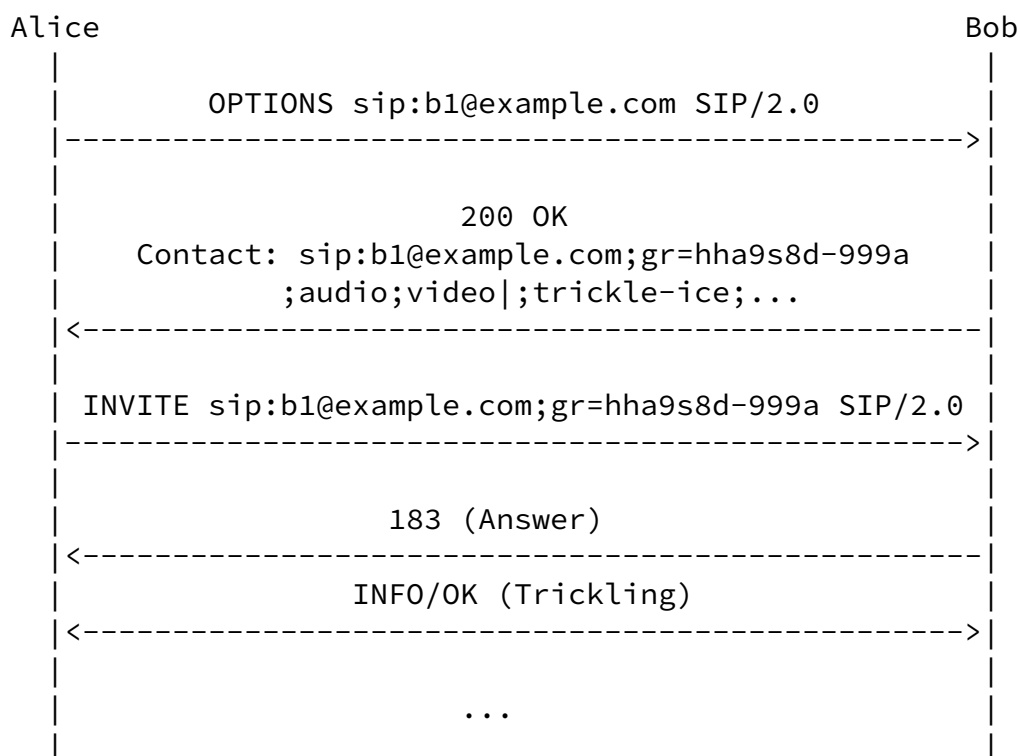


Figure 7: Trickle ICE support discovery with OPTIONS and GRUU

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.](#) Trickle ICE discovery through other protocols

Protocols like XMPP [[RFC6120](#)] define advanced discovery mechanisms that allow specific features to be queried prior to actually attempting to use them. Solutions like [[RFC7081](#)] define ways of using SIP and XMPP together which also provides a way for dual stack SIP+XMPP endpoints to make use of such features and verify trickle ICE support for a specific SIP endpoint through XMPP. [TODO expand on a specific way to do this]

### [5.4.](#) Fallback 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-mmusic-trickle-ice](#)]. With half trickle, agents initiate sessions the same way they would when using vanilla ICE [[RFC5245](#)]. This means that, prior to actually sending an offer, agents would first gather ICE candidates in a blocking way and then send them all in that offer. The blocking nature of the process would likely imply that some amount of latency will be accumulated and it is advised that agents try to anticipate it where possible, like for example, when user actions indicate a high likelihood for an imminent call (e.g., activity on a keypad or a phone going offhook).

Using half trickle would result in offers that are compatible with both vanilla ICE and legacy [[RFC3264](#)] endpoints both.

A typical (half) trickle ICE exchange with SIP would look this way:

Internet-Draft

Trickle ICE for SIP

June 2014

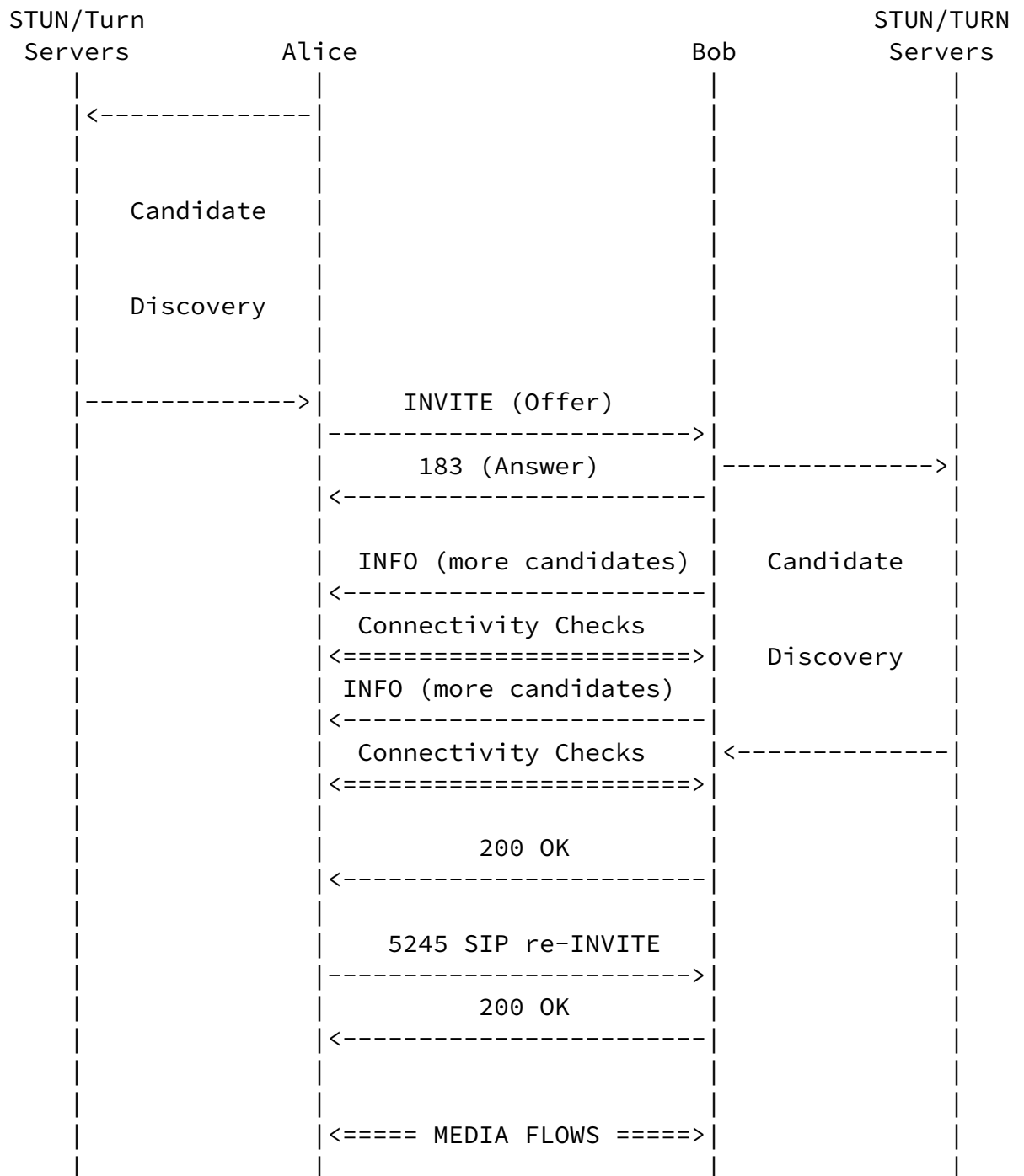


Figure 8: Example

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.](#) Info Package

### [6.1.](#) Overall Description

This specification defines an Info Package meant for use by SIP user agents implementing Trickle ICE. Typically INFO requests would carry ICE candidates discovered after the user agent has sent or received a trickle-ice offer.

### [6.2.](#) Applicability

The purpose of the ICE protocol is to establish a media path. The candidates that this specification transports in INFO requests are part of this establishment. There is hence no way for them to be transported through the not yet existing media path.

Candidates sent by a trickle ICE agent after the offer, are meant to follow the same signalling 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 with no GRUU support. Using out-of-dialog SUBSCRIBE/NOTIFY requests would not satisfy this goal.

### [6.3.](#) Info Package Name

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

### [6.4.](#) Info Package Parameters

This document does not define any Info package parameters.

## [6.5.](#) SIP Option-Tags

[RFC6086] allows Info Package specifications to define SIP option-tags. This document therefore stipulates that SIP entities that support trickle ICE and this specification MUST place the 'trickle-ice' option-tag in a SIP Supported header field.

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

Ivov, et al.

Expires December 19, 2014

[Page 18]

---

Internet-Draft

Trickle ICE for SIP

June 2014

## [6.6.](#) Info Message Body Parts

Entities implementing this specification MUST include SDP encoded ICE candidates in all SIP INFO requests. The MIME type for the payload MUST be of type 'application/sdp' as defined in [Section 4.2](#) and [\[I-D.ietf-mmusic-trickle-ice\]](#).

## [7.](#) Security Considerations

[TODO]

## [8.](#) Acknowledgements

The authors would like to thank Thomas Stack for suggesting the INFO acknowledgements used in the specification as a way of avoiding making PRACKs mandatory, Paul Kyzivat and Jonathan Lennox for making various suggestions for improvements and optimisations.

## [9.](#) References

### [9.1.](#) Normative References

[I-D.ietf-mmusic-trickle-ice]

Ivov, E., Rescorla, E., and J. Uberti, "Trickle ICE: Incremental Provisioning of Candidates for the Interactive Connectivity Establishment (ICE) Protocol", [draft-ietf-](#)

[mmusic-trickle-ice-01](#) (work in progress), February 2014.

[I-D.ivov-dispatch-sdpfrag]

Ivov, E. and A. Roach, "Internet Media Type application/sdpfrag", [draft-ivov-dispatch-sdpfrag-03](#) (work in progress), October 2013.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", [RFC 3261](#), June 2002.

[RFC3262] Rosenberg, J. and H. Schulzrinne, "Reliability of Provisional Responses in Session Initiation Protocol (SIP)", [RFC 3262](#), June 2002.

[RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", [RFC 3264](#), June 2002.

Ivov, et al.

Expires December 19, 2014

[Page 19]

---

Internet-Draft

Trickle ICE for SIP

June 2014

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", [RFC 4566](#), July 2006.

[RFC5245] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", [RFC 5245](#), April 2010.

[RFC6086] Holmberg, C., Burger, E., and H. Kaplan, "Session Initiation Protocol (SIP) INFO Method and Package Framework", [RFC 6086](#), January 2011.

[RFC6120] Saint-Andre, P., "Extensible Messaging and Presence Protocol (XMPP): Core", [RFC 6120](#), March 2011.

## [9.2.](#) Informative References

[RFC3388] Camarillo, G., Eriksson, G., Holler, J., and H. Schulzrinne, "Grouping of Media Lines in the Session

Description Protocol (SDP)", [RFC 3388](#), December 2002.

- [RFC3840] Rosenberg, J., Schulzrinne, H., and P. Kyzivat, "Indicating User Agent Capabilities in the Session Initiation Protocol (SIP)", [RFC 3840](#), August 2004.
- [RFC5627] Rosenberg, J., "Obtaining and Using Globally Routable User Agent URIs (GRUUs) in the Session Initiation Protocol (SIP)", [RFC 5627](#), October 2009.
- [RFC7081] Iovov, E., Saint-Andre, P., and E. Marocco, "CUSAX: Combined Use of the Session Initiation Protocol (SIP) and the Extensible Messaging and Presence Protocol (XMPP)", [RFC 7081](#), November 2013.

## [Appendix A](#). Open issues

At the time of writing of this document the authors have no clear view on how and if the following list of issues should be address here:

1. Should we allow for full trickle if support can be verified automatically and confirmed for a gruu with [\[RFC3840\]](#).
2. Can we pick between MID and stream indices for stream identification.

Iovov, et al.	Expires December 19, 2014	[Page 20]
---------------	---------------------------	-----------

---

Internet-Draft	Trickle ICE for SIP	June 2014
----------------	---------------------	-----------

## Authors' Addresses

Emil Iovov  
Jitsi  
Strasbourg 67000  
France

Phone: +33 6 72 81 15 55  
Email: [emcho@jitsi.org](mailto:emcho@jitsi.org)

Enrico Marocco

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

Email: [enrico.marocco@telecomitalia.it](mailto:enrico.marocco@telecomitalia.it)

Christer Holmberg  
Ericsson  
Hirsalantie 11  
Jorvas 02420  
Finland

Email: [christer.holmberg@ericsson.com](mailto:christer.holmberg@ericsson.com)