| | |
|---|---|
| Network Working Group | J. Rosenberg |
| Internet-Draft | Cisco |
| Intended status: Standards Track | July 14, 2008 |
| Expires: January 15, 2009 | |

TOC

**Guidelines for Usage of Interactive Connectivity Establishment (ICE) by non Session Initiation Protocol (SIP) Protocols**
**draft-rosenberg-mmusic-ice-nonsip-01**

**Status of this Memo**



**Abstract**

Interative Connectivity Establishment (ICE) has been specified as a NAT traversal mechanism for protocols based on the offer/answer exchange model. In practice, only the Session Initiation Protocol (SIP) has used ICE. This document provides guidance on how other protocols can make use of ICE.

---

**Table of Contents**

---

## 1. Introduction

TOC

Interactive Connectivity Establishment (ICE) [I-D.ietf-mmusic-ice] (Rosenberg, J., “Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols,” October 2007.) has been specified by the IETF as a mechanism for NAT traversal for protocols based on the offer/answer model [RFC3264] (Rosenberg, J. and H. Schulzrinne, “An Offer/Answer Model with Session Description Protocol (SDP),” June 2002.), which exchanges Session Description Protocol (SDP) [RFC4566] (Handley, M., Jacobson, V., and C. Perkins, “SDP: Session Description Protocol,” July 2006.) objects to negotiate media sessions.
ICE has many benefits. It is automated, relying on very little configuration. It works through an extremely broad range of network and NAT topologies. It is robust, establishing connections in many challenging environments. It is efficient, utilizing relays and intermediaries only when other options will not work. At the time of writing, ICE has seen widespread usage on the Internet for traversal of Voice over IP, primarily based on the Session Initiation Protocol (SIP)

[RFC3261] (Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, “SIP: Session Initiation Protocol,” June 2002.)
However, SIP is not the only protocol that requires the establishment of host-to-host relationships for communications. Consequently, ICE has recently been considered as the NAT traversal technique for other protocols. These include Peer-to-Peer SIP (P2PSIP) [I-D.bryan-p2psip-reload] (Jennings, C., Lowekamp, B., Rescorla, E., Baset, S., and H. Schulzrinne, “REsource LOcation And Discovery (RELOAD),” June 2008.), Host Identity Protocol (HIP) [I-D.manyfolks-hip-sturn] (Nikander, P., Melen, J., Komu, M., and M. Bagnulo, “Mapping STUN and TURN messages on HIP,” November 2007.) and Mobile IP v6 [I-D.tschofenig-mip6-ice] (Tschofenig, H., “Mobile IP Interactive Connectivity Establishment (M-ICE),” February 2008.). In each case, the protocol in question provides a mechanism for two hosts to rendezvous through some intermediary, and then needs a host-to-host connection established. This fits the NAT traversal capability provided by ICE.
Unfortunately, the ICE specification itself is intertwined with SDP and the offer/answer model, and is not immediately usable by protocols that do not utilize offer/answer. For this reason, each of these protocols need to define how to utilize ICE for their specific needs. This document provides guidelines for authors of such specifications. It includes guidance on when ICE can be used by a protocol, describes each of ICE's major functions and how they can be applied.
This document assumes the reader is familiar with ICE and its operation.

---

## 2. Can My Protocol Use ICE?

TOC

Not all protocols can make use of ICE. ICE works only with protocols that fit the pattern of a session protocol. A session protocol is one in which there exists some kind of rendezvous service, typically through a server on the Internet, by which hosts can contact each other. Through the rendezvous service, hosts can exchange information for the purposes of negotiating a direct host to host connection. Each host is assumed to have an identifier by which it is known to the rendezvous service, and by which other hosts can identify it. There is typically some kind of registration operation, by which a host connects to the rendezvous service and identifies itself. This protocol design pattern is shown in Figure 1 (Session Protocols).

---

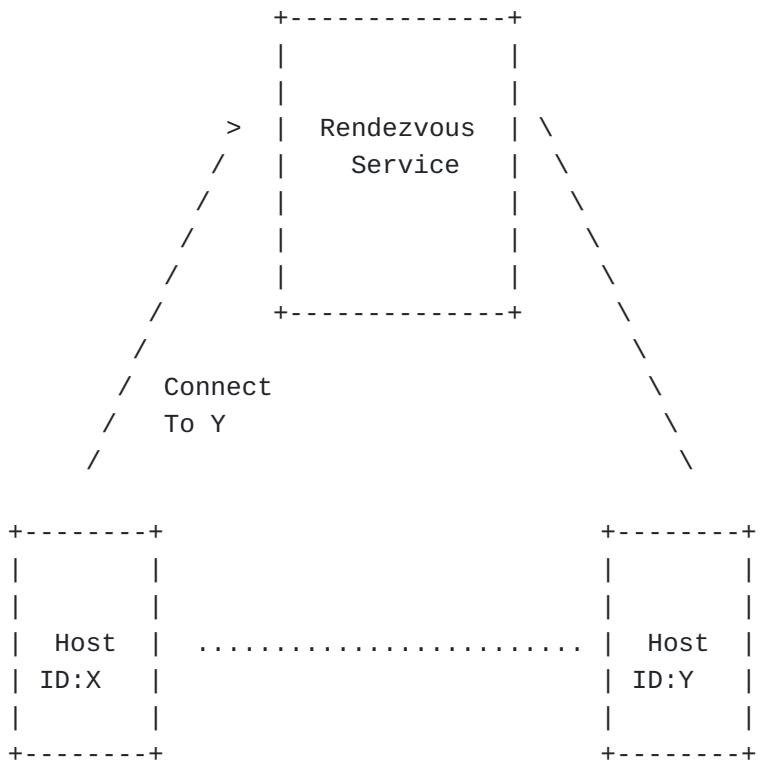


**Figure 1: Session Protocols**

---

If hosts can reach each other through the rendezvous service, why create direct connections? Typically, the rendezvous service provides an indirect connection, and may be very suboptimal in terms of latency and other path metrics. The rendezvous service may also have limited bandwidth, and not be capable of supporting the volume of data required to flow between the hosts.

As an example, in SIP, the rendezvous service is the SIP server. The identifier is the SIP URI. The registration process is supported using the SIP REGISTER method. Connections are established using the INVITE method.

For a protocol to use ICE, it must exhibit the properties of a session protocol as described above. Furthermore, it must provide a mechanism for exchanging information between the hosts for purposes of establishing the connection. It must provide for, at least, one message from the initiator to the other host, and one message back. If all of these criteria are met, ICE can be used.

---

TOC

## 3. Target Architecture

The goal of the recommendations in this document is to enable an architecture for firewall and NAT traversal across many protocols that has two properties:

1. STUN and TURN servers can be used to support multiple applications

2. Gateways can easily be built between ICE-using protocols that are compatible

The second of these requires further discussion. In some cases, two different protocols are ones that provide similar functions, so that it is reasonable to build gateways between them. For example, gateways between SIP and H.323, or between SIP and RTSP, are reasonable things to do. A gateway function between two session protocols needs to concern itself with converting the signaling and converting the media protocol - whether it be RTP or something else. It is highly desirable to avoid actual conversion operations along the direct media path. These greatly increase the cost and complexity of gateway functions. Consequently, the ideal architecture looks like this:

---

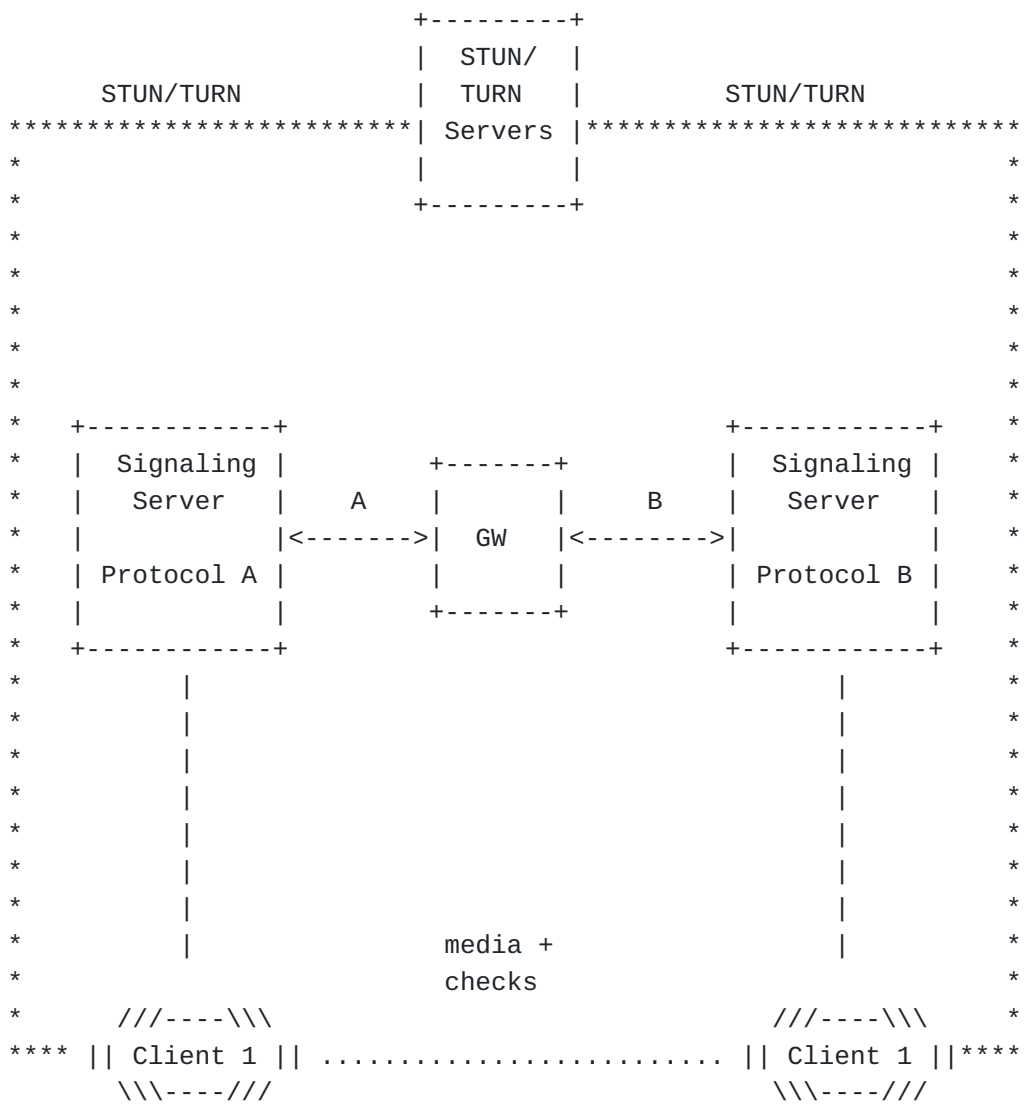


**Figure 2: Ideal Multi Protocol ICE Architecture**

---

In this architecture, clients of two different protocols (A and B) make use of signaling servers for their respective protocols. There is a gateway function between them, but this function ONLY concerns itself with the signaling. The content of the established sessions - which includes the media and the path-based connectivity checks that ICE uses - do not require any protocol conversion.
Of course, implementations can choose to gateway the media and checks if they want, but it is a strong objective of the recommendations here that they don't HAVE TO.

The architecture also shows that the goal is to have a common set of TURN and STUN functions that serve all applications using ICE.

---

## 4. General Considerations

TOC

There are some general considerations for the using protocol.

---

### 4.1. Lite Implementation

TOC

The lite mode of operation for ICE allows for usage by agents which are always reachable by any other agent, both now and in the future. The using protocol needs to decide whether this mode of operation is supported or not. If not, all agents will be full implementations. If the mode is supported, agents can either be lite or full.
The principal consideration is the likelihood of agents being always publicly reachable, vs. the cost of an ICE implementation. ICE itself provides strong caution against the lite mode of implementation. It is very easy for protocol designers to envision specific scenarios for deployment of their protocol, and then for the reality to be different. Furthermore, the full mode provides important security benefits. It ensures that an ICE implementation cannot be used to launch DoS attacks. Consequently, that same guidance is given here: using protocols should only use ICE's lite mode if there is a belief that implementors absolutely will not implement the full mode, and that those implementations will always be publicly reachable by every other agent for the lifetime of deployment of that implementation, and that the security benefits of full mode are not worth the implementation complexity.

---

### 4.2. Multiple Components

TOC

ICE introduces the concept of multiple components for a single media stream. ICE attempts to provide atomic processing across components, such that a set of candidates (one for each component) are only used if all of them succeeded. This grouping is useful when it is desirable for path characteristics to be identical across multiple IP addresses and ports that make up a connection of some sort.
Using protocols should indicate whether this functionality is needed or not. If not, the procedures defined for ICE are used as is, but the implementation for the using protocol just assumes there is always a single component per stream.

### 4.3. Multiple Media Streams

TOC

ICE allows for multiple media streams. ICE largely runs independently for each stream, with a few important exceptions. First, ICE will perform pacing across all of the streams, thus providing aggregate congestion control. Secondly, ICE will utilize results from one stream to speed up the results of candidate gathering for another stream. Using protocols should decide whether the concept of multiple streams applies or not. If it does, the using protocol can elect to run ICE on each stream completely independently (in which case its effectively a separate offer/answer exchange and ICE state machine for each stream), or together. The primary consideration, as noted above, is whether aggregate congestion control and rapid convergence are desired. This document recommends that, if a using protocol has multiple streams, it runs ICE jointly across them, as defined by the ICE specification (in other words, there is one instance of the ICE state machine, not one for each media stream).

## 5. ICE Functions

TOC

ICE processing can be broken six distinct steps:

1. Gathering of candidates
2. Initial exchange of candidates
3. Connectivity checks
4. Conclusion of ICE
5. Subsequent signaling
6. Media and Keepalives

Each of these steps requires consideration by the designer of the protocol that intends to use ICE (called the using protocol).

### 5.1. Gathering of Candidates

TOC

This phase of operation involves the gathering of candidates by the agent. Any using protocol will need to perform this step. The

specification for the using protocol should point to Section 4.1 of ICE, and dictate that the procedures there be followed. However, there are several aspects of the gathering operation which are subject to considerations by the using protocol, and the using protocol should provide additional guidance on whether any of these behaviors change or not.

---

#### 5.1.1. Candidate types

TOC

ICE allows an agent, as a matter of policy, to gather candidates of a particular type - host, server reflexive, and relayed. Consequently, a using protocol needs to define whether its agents will support all three, or just a subset. ICE recommends strongly that all three types be gathered and supported. This is because reliability of connection establishment cannot be provided unless all three mechanisms are implemented. Using protocols should only utilize a subset if their deployment topologies are limited to cases where one of the agents will always be behind NATs with endpoint independent mapping properties.

---

#### 5.1.2. Pacing

TOC

ICE defines a pacing algorithm for rate limiting the traffic it generates during the gathering phase. When used in conjunction with the parameter computations in Section 16.2, those algorithms are applicable to any protocol. However, they may be overly conservative for certain applications. Consequently, using protocols can define alternative mechanisms for pacing ICE.
However, using protocols should be aware that there are two issues that drove the design of the pacing. One of them is network congestion control. The using protocol has to ensure that its pacing remains TCP friendly whenever possible. The second issue is NAT overload. Testing of NAT devices at the time of writing showed that some of them went into an 'overload' mode when too many mappings were created within a short interval of time. Keeping the creation of new mappings to a rate less than one every 50ms seemed to address this problem. Using protocols should follow a similar design goal.

---

#### 5.1.3. Number and Discovery of Servers

TOC

ICE only defines operations for a single STUN server
[I-D.ietf-behave-rfc3489bis] (Rosenberg, J., Mahy, R., Matthews, P.,

and D. Wing, “Session Traversal Utilities for (NAT) (STUN),” July 2008.), or for a single TURN server [I-D.ietf-behave-turn] (Rosenberg, J., Mahy, R., and P. Matthews, “Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN),” July 2009.). It does not consider cases where there are multiple STUN and/or multiple TURN servers used by the agent. However, this is an omission for the sake of simplicity. If a using protocol has a need to highly optimize the connection paths in multi-layer natting environments, multiple STUN servers - ideally one behind each NAT - can provide an optimal path. A using protocol can elect to specify that multiple STUN servers be used in these cases.

Of course, the using protocol will need to specify how a client discovers or is configured with those additional STUN servers. The usage of multiple STUN servers affects pacing; the overall rate of candidate gathering across all servers needs to be congestion controlled and stay below the rate of a new allocation every 50ms.

---

#### 5.1.4. Other Protocols

TOC

It is possible that the using protocol can define or utilize other mechanisms for gathering candidates. For example, a mechanism may be built into the rendezvous protocol itself. Indeed, this is the primary reason for using something besides STUN and TURN. If a using protocol is not building such functionality into the rendezvous server itself, it is highly recommended that it reuse the STUN and TURN protocols. The primary reason for this is that it allows a domain to deploy STUN and TURN servers just once, and then reuse them for multiple protocols that require NAT traversal functionality. This reuse is highly desirable, and would likely outweigh any minor protocol improvements that could come from 'rolling your own' mechanism for gathering candidates. This is why an exception is called out for building the gathering protocol into the rendezvous server itself; that server needs to be deployed anyway.

However, there are pitfalls to building a candidate gathering mechanism into the rendezvous protocol and server. In particular, obtaining relayed candidates from a rendezvous protocol can be problematic. TURN servers are ideally deployed throughout the network, in points that are topologically close to clients. Since the whole purpose of ICE is to allow two clients to connect directly to each other without sending data through the rendezvous server, building TURN-like functionality into the rendezvous server defeats much of the purpose of ICE itself. Such a move only makes sense if it is believed that, for the using protocol, the likelihood of usage of relayed candidates is particularly low.

---

### 5.1.5. Prioritization

TOC

ICE allows the prioritization of candidates to be a matter of local policy. Using protocols may define their own policy for how candidates are prioritized. However, protocols absolutely must utilize the same range of priority values (0 to 2^32 - 1), and must use the concepts of foundations and bases, along with the procedures for eliminating redundant candidates. Utilizing those ensures that ICE can be interoperated easily between different using protocols with only a gateway function on the signaling, not the media.

---

### 5.1.6. Default Candidates

TOC

The concept of default candidates is primarily to support backwards compatibility, and may not be required for a using protocol. Firstly, if a using protocol is being defined for the first time, and ICE is being used as a mandatory-to-implement part of the protocol, then clearly there are no backwards compatibility issues, and the default candidate mechanism is not needed.
Even in cases where there are older, non-ICE implementations, there are several basic mechanisms that can be used to deal with it:

> **Capability Query:** An ICE-compliant agent can query the target agent, prior to an ICE exchange, to determine if they support ICE. If they do, the agent proceeds with the ICE exchange, otherwise, they do not. If a using protocol utilizes this basic technique, the default candidate mechanism is not needed.
>
> **Fail-and-Retry:** An ICE-compliant agent sends an initial message with ICE parameters, along with some kind of flag which tells the recipient to reject the message if it doesn't support ICE. If such a rejection is received, the agent retries without ICE. If a using protocol utilizes this technique, the default candidate mechanism is not needed.
>
> **Fallback:** An ICE-compliant agent sends an initial message with ICE parameters, but they are encoded in such a way that they will be ignored by a non-ICE implementation. If a non-ICE implementation receives them, it sends back an answer without ICE, and the offerer notices this and proceeds without ICE. This technique requires the default candidate mechanism defined by ICE.

Of these three approaches, the first two require potentially two round trips to setup a session, whereas the third can do it in a single round trip regardless of the capabilities of the other agent. When latency

for establishment is an important concern, the fallback approach is preferable.

---

### 5.2. Initial Exchange of Candidates

TOC

ICE specifies the usage of the offer/answer protocol and the Session Description Protocol for exchanging ICE parameters. This mechanism is clearly ICE specific, and the using protocol should define something appropriate. However, in all cases, the protocol exchange has to allow for a two-phase exchange where one side offers ICE information to the other, and the other offers ICE information back in response. Though it is possible for protocols to utilize mechanism other than a two-phase exchange, this is not recommended, since it significantly complicates the construction of gateways between protocols that utilize ICE.

---

#### 5.2.1. ICE Mismatch

TOC

The ICE mismatch feature is very specific to SIP. It is a consequence of the existence of intermediaries which routinely modify the media destination in the SDP, but are not ICE aware and will just ignore (and pass on) any ICE attributes that are present. The ICE mismatch mechanism detects these cases and falls back to non-ICE operation. A using protocol should only utilize this mechanism if it happens to have similar deployment constraints.

---

#### 5.2.2. Parameter Encoding

TOC

The syntax for the messages is entirely a matter of convenience for the using protocol. However, the following parameters and their data types needs to be conveyed in the initial exchange:

> **Candidate attribute** There will be one or more of these for each "media stream". Each candidate is composed of:
>
> > **Foundation:** A sequence of up to 32 characters.
> >
> > **Component-ID:** This would be present only if the using protocol were utilizing the concept of components. If it

is, it would be a positive integer that indicates the component ID for which this is a candidate.

**Transport:** An indicator of the transport protocol for this candidate. This need not be present if the using protocol will only ever run over a single transport protocol. If it runs over more than one, or if others are anticipated to be used in the future, this should be present.

**Priority:** An encoding of the 32 bit priority value.

**Candidate Type:** The candidate type, as defined in ICE.

**Related Address and Port:** The related IP address and port for this candidate, as defined by ICE.

**Extensibility Parameters:** The using protocol should define some means for adding new per-candidate ICE parameters in the future.

**Lite Flag:** If ICE lite is used by the using protocol, it needs to convey a boolean parameter which indicates whether the implementation is lite or not.

**Ufrag and Password:** The using protocol has to convey a username fragment and password. It must allow up to 256 characters for the ufrag and 256 for the password.

**ICE extensions:** In addition to the per-candidate extensions above, the using protocol should allow for new media-stream or session-level attributes.

If the using protocol is using the ICE mismatch feature, a way is needed to convey this parameter in answers. If is a boolean flag.
The exchange of parameters is symmetric; both agents need to send the same set of attributes as defined above.
The using protocol may (or may not) need to deal with backwards compatibility with older implementations that do not support ICE. If the fallback mechanism is being used, then presumably the using protocol already provides a way of conveying the default candidate (its IP address and port) in addition to the ICE parameters.

---

### 5.2.3. Role Determination

TOC

The role determination mechanism must be used by the using protocol. However, the conflict resolution algorithm in Section 5.2 of ICE is almost entirely an artifact of the fact that SIP separates its

signaling exchange from the offer/answer exchange. In using protocols that lack this separation, the conflict resolution algorithm itself will never get used.

---

### 5.3. Connectivity Checks

TOC

The core of the ICE algorithm is the connectivity checks. After both sides have gathered candidates and have exchanged them with each other, the check process begins. Here, it is very important that the using protocol simply follow the mechanisms already defined by ICE. Implementations should directly utilize the functionality defined in Section 5.7 to compute pairs and priorities, prune, form the check lists, and compute states. If a using protocol has elected not to use the concepts of multiple components or multiple streams, these algorithms simplify. However, the using protocol must not specify a different algorithm; it can only reuse what is there and constrain its behavior by mandating constrained inputs (only one component, or only one media stream).
The actual connectivity checks themselves must also be performed exactly as defined in Section 7 of ICE. The using protocol should just reference that section directly. Note that, even if a using protocol does not need to use the role conflict detection mechanism, it must include the ICE-CONTOLLED and ICE-CONTROLLING attributes in its connectivity checks as described in Section 7 of ICE. This ensures that it is possible to easily build gateways between different protocols using ICE.

---

#### 5.3.1. Scheduling Checks

TOC

The primary area where using protocols can alter the behavior defined in ICE is in the area of pacing. The using protocol can define different mechanisms for computing Ta and RTO, and may even define a different mechanism entirely for interleaving scheduled and triggered checks.
As with the pacing of candidate gathering, the pacing of connectivity checks needs to take congestion control and NAT overload into consideration.

---

TOC

### 5.4. Conclusion of ICE

The procedures for concluding ICE as defined in Section 8 should be used as defined for the using protocol, with only a few areas of flexibility.

---

#### 5.4.1. Regular vs. Aggressive Nomination

TOC

The primary area of flexibility is around regular vs. aggressive nomination. A using protocol can mandate that all implementations use one or the other or allow for both. The considerations for this choice are identical for the using protocol as they are for ICE in general. Aggressive nomination is faster but can introduce glitches; regular nomination is slower but is more stable. Regular nomination is recommended if at all possible.

---

#### 5.4.2. Updated Signaling and Remote Candidates

TOC

ICE defines conditions on which an updated offer is required to be sent after ICE concludes - namely, if the candidates selected by ICE are not a match for the default candidates, an updated exchange is sent.
This function of ICE is primarily an artifact of the realities of SIP deployments. It is not at all needed for correctness of ICE operation. In the case of SIP, signaling intermediaries that are inspecting the offer/answer exchanges, but are not ICE aware, will be confused unless there is an updated exchange. This same consideration applies to using protocols. If the using protocol has deployments with intermediaries that inspect messages, and will be confused if the actual connections/ media are established to something different than any defaults that were signaled, the updated exchange should be used. If not, it can be avoided.
If it is used, the remote-candidates attribute has to be conveyed in the updated offer, and the agents need to implement the algorithms described in Section 9 of ICE for setting the answer based on this attribute. Furthermore, the signaling protocols require a way to encode it.

---

### 5.5. Subsequent Signaling

TOC

ICE defines procedures for performing subequent offer/answer exchanges that have an affect of updating the state of ICE. Support for

subsequent exchanges is needed if the using protocol requires any of the following capabilities:

- The ability to add a new candidate to a set while ICE is already in progress, without abandoning the progress so far.

- The ability to add a new media stream, or remove a new media stream, without redoing ICE processing for all of the media streams.

- The ability to change the IP address or port for a media stream, but to do so with a "make before break" property - so that the new destination begins to be used only once checks for the new destination have completed.

If any of these properties are important, ICE's capabilites for subsequent signaling should be utilized.
One use case where these functions are not needed is when the using protocol fundamentally doesn't allow any kind of updating of connection addresses. If it requires the previous connection to be closed, and a new one to be opened starting from scratch, ICE's subsequent signaling feature is not needed.
If subsequent signaling is used, ICE restarts must be supported.

---

### 5.6. Media and Keepalives

TOC

The keepalive procedures in Section 10 must be used as defined. The media handling rules in Section 11 apply as well, with the exception of the RTP-specific guidelines.

---

## 6. Security Considerations

TOC

Several ICE features exist to provide security, including the message integrity mechanism. Using protocols must use these in the same way ICE does.
The guidelines defined here do allow a using protocol to support the ICE lite mode of operation. The lite mode is less secure than full mode, as it allows an implementation to be used as a source of DoS traffic. For this reason, using protocols must address, in their security considerations, why they have elected to allow the lite implementation in cases where it is being supported.

---

## 7. IANA Considerations

TOC

There are no IANA considerations associated with this specification.

---

## 8. Informative References

TOC


| | |
|---|---|
| [I-D.ietf-mmusic-ice] | Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols," draft-ietf-mmusic-ice-19 (work in progress), October 2007 (TXT). |
| [RFC3264] | Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)," RFC 3264, June 2002 (TXT). |
| [RFC4566] | Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol," RFC 4566, July 2006 (TXT). |
| [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 (TXT). |
| [I-D.bryan-p2psip-reload] | Jennings, C., Lowekamp, B., Rescorla, E., Baset, S., and H. Schulzrinne, "REsource LOcation And Discovery (RELOAD)," draft-bryan-p2psip-reload-04 (work in progress), June 2008 (TXT). |
| [I-D.manyfolks-hip-sturn] | Nikander, P., Melen, J., Komu, M., and M. Bagnulo, "Mapping STUN and TURN messages on HIP," draft-manyfolks-hip-sturn-01 (work in progress), November 2007 (TXT). |
| [I-D.tschofenig-mip6-ice] | Tschofenig, H., "Mobile IP Interactive Connectivity Establishment (M-ICE)," draft-tschofenig-mip6-ice-02 (work in progress), February 2008 (TXT). |
| [I-D.ietf-behave-rfc3489bis] | Rosenberg, J., Mahy, R., Matthews, P., and D. Wing, "Session Traversal Utilities for (NAT) (STUN)," draft-ietf-behave-rfc3489bis-18 (work in progress), July 2008 (TXT). |
| [I-D.ietf-behave-turn] | Rosenberg, J., Mahy, R., and P. Matthews, "Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN)," draft-ietf-behave-turn-16 (work in progress), July 2009 (TXT). |


---

## Author's Address

TOC


| | |
|---|---|
| | Jonathan Rosenberg |
| | Cisco |
| | Edison, NJ |
| | US |
| Phone: | +1 973 952-5000 |
| Email: | jdrosen@cisco.com |
| URI: | http://www.jdrosen.net |


---

## Full Copyright Statement

TOC



## Intellectual Property