Network Working Group	J. Rosenberg	
Internet-Draft	Cisco	
Intended status: Standards Track	February 15, 2008	
Expires: August 18, 2008		

TOC

NICE: Non Session Initiation Protocol (SIP) usage of Interactive Connectivity Establishment (ICE)
draft-rosenberg-mmusic-ice-nonsip-00

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of BCP 79.
Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.
Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as “work in progress.”
The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.
The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.
This Internet-Draft will expire on August 18, 2008.

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 been based on the offer/answer model. This document defines a SIP independent subset of ICE, called NICE, which can be used with any protocol wishing to establish a direct host-to-host relationship through NAT. Protocol specifications need only reference this document, and include the object defined here in their messages, in order to achieve NAT traversal.

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 has needed to define its own usage of ICE. This results in duplicate work and inconsistent solutions for NAT traversal.
To remedy this, this document defines a generic NAT traversal solution based on ICE, called NICE. It does so by referencing the specific parts of the ICE specification that are needed. It also defines a simply object that can be exchanged in other protocols. Consequently, protocols that fit the design pattern for NICE need only reference this document, and provide a way to include the defined object in their messages. With that, they have a solution for NAT traversal.

---

2. Can My Protocol Use NICE?

TOC

Not all protocols can make use of NICE. NICE 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).

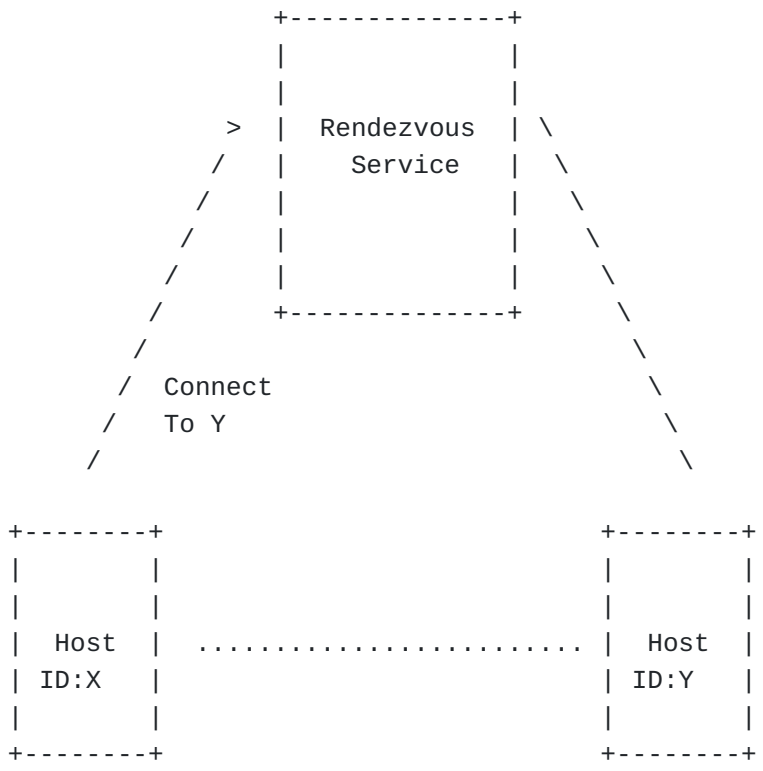

Figure 1: Session Protocols

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 REGISTER method. Connections are established using the INVITE method.

For a protocol to use NICE, it must exhibit the properties of a session protocol as described above. Furthermore, it must provide a mechanism for exchanging MIME objects 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, NICE can be used.

TOC

3. 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 RFC 2119 (Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels," March 1997.) [RFC2119].
In addition, this document introduces the following terms:

Session Initiator: A software or hardware entity on a host that wishes to establish establish communications with another host, called the session responder. A session initiator is also called an initiator.

Initiator: Another term for a session initiator.

Session Responder: A software or hardware entity on a host that receives a request for establishment of communications from the session initiator, and either accepts or declines the request. A session responder is also called a responder.

Responder: Another term for a session responder.

Client: Either the initiator or responder.

Peer: From the perspective of one of the clients in a session, its peer is the other client. Specifically, from the perspective of the initiator, the peer is the responder. From the perspective of the responder, the peer is the initiator.

Rendezvous Service: A protocol service provided to the clients that allows them to identify each other using a well known identifier, and then send messages back and forth.

Initiate Message: The message in the rendezvous protocol used by an initiator to establish communications. It contains the ICE parameters needed to establish communications.

Accept Message: The message in the rendezvous protocol used by a responder to establish communications. It contains the ICE parameters needed to establish communications.

---

4. Overview of Operation

TOC

To utilize NICE, one host, the INITIATOR, sends a message using the rendezvous protocol. This message is addressed towards another host,

the RESPONDER. This message is called the Initiate message. That message contains a MIME object, specified in Section 15 (The NICE Object), which includes the information needed by NICE. In particular, it contains a set of candidates for the purposes of establishing a single "stream". This stream is a host-to-host UDP association or TCP connection. The rendezvous service delivers the Initiate message to the RESPONDER. It sends a message back to the initiator, called the Accept message. This message also carries the same object, containing information from the Responder for the purposes of establishing the stream.

NICE uses server reflexive and relayed candidates learned from Session Traversal Utilities for NAT (STUN) STUN [I-D.ietf-behave-rfc3489bis] (Rosenberg, J., Mahy, R., Matthews, P., and D. Wing, "Session Traversal Utilities for (NAT) (STUN)," July 2008.) and Traversal Using Relay through NAT (TURN) [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.) servers. These functions can be provided by the rendezvous service, or by traditional STUN and TURN servers in the network. The candidates learned from these servers are what is included in the objects exchanged through the rendezvous protocol.

Once exchanged, the clients perform connectivity checks. These checks probe for connectivity between the pairs of candidates signaled through the rendezvous protocol. Once a match is found, the Initiator sends an updated connectivity check, indicating that a pair has been selected. At that point, packets can flow between initiator and responder.

---

5. Differences between ICE and NICE

TOC

NICE differs from ICE in two fundamental ways. Firstly, it is abstracted from SDP and RTP specifics. Secondly, it is subsetted. This subsetting operation removes many of the features in ICE that are there for reasons having to do with the nuances of SIP, or the need for real-time operation. In particular, the following ICE features are not used in NICE:

- ICE has the notion of a default candidate. This default candidate is used for backwards compatibility with pre-ICE SIP implementations. That mechanism is very specific to SDP backwards compatibility techniques, and is not used here. Instead, if the protocol using NICE requires backwards compatibility, it needs to define its own mechanisms for such.

- ICE supports the notion of updated offers and answers that can modify information. Indeed, it requires such an update when the pair selected by ICE does not match the default. The notion of

default has been removed in NICE, as has the ability to update the ICE information. This update allowed for mid-call changes in connectivity, a frequent occurrence in events like call transfer. If a protocol using NICE requires a connection to a different host, it has to start a totally new and unrelated ICE session. This can result in discontinuous connectivity while the checks re-run. Continuous operation is needed for real-time usage, but not more generally.

*Simllarly, ICE restarts are not supported in NICE. Restarts are an artifact of sending updated offers and answers.

*ICE provides some guidance for handling SIP forking. This is a case where a single offer elicits multiple answers. Forking is specific to SIP, and so this capability is removed from NICE. NICE allows connectivity to be set up only between a pair of hosts.

*ICE defines a lite mode of operation for supporting ease of implementation. Since NICE is already simpler by the removal of several large ICE features (most notably updated offers and answers), this simplified mode seems unneeded. It seems better to simplify NICE overall rather than define complexity in the normal mode in order to introduce a simplified lite mode.

*ICE supports the notion of multiple streams and multiple components per stream. This was done specifically to address the needs of multimedia. NICE provides the ability to establish a single connection between a pair of hosts. Consequently, that capability is not present in NICE.

*ICE defines an algorithm called the Frozen algorithm. This algorithm exists to speed up completion of ICE in cases where multiple candidates share similar properties. For example, when an audio and video candidate are on the same host IP address. Since NICE only supports a single candidate and a single component, the use cases for the Frozen algorithm diminish significantly. Furthermore, the Frozen algorithm is entirely about speed and is not as much an issue for more general non-real tiem protocols . Thus, this algorithm is not used by NICE. It falls out by using the algorithm defined in ICE, but by setting each foundation to a unique value.

*ICE defines SDP attributes for "remote-candidates". These are used to resolve a race condition between a subsequent offer/ answer and the ICE checks. Since NICE does not use any subsequent rendezvous signaling, this attribute and its procedures are not used in NICE.

*ICE defines an SDP attribute called "ice-mismatch". This detects an ICE failure case due to the presence of signaling intermediaries that don't support ICE. This problem is specific to SIP and thus this attribute and associated procedures are not used in NICE.

---

6. Gathering Candidates

TOC

When a client wishes to establish a connection, it follows the process of gathering candidates as described in Section 4.1 of 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.). However, the client MUST follow those rules under the assumption of a single media stream and a single component for that stream. This simplification means that component ID for an ICE candidate is always one. In addition, the rules in Section 4.1.1.3 MUST be ignored; instead, each candidate MUST have a unique foundation, assigned arbitrarily by the client.

If the client wishes to establish a TCP connection and not a UDP stream, or wishes to try both, the client MUST implement ICE-tcp [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.), and MUST follow the procedures defined there for gathering of TCP candidates, again assuming a single component.

The default candidate selection described in Section 4.1.3 of ICE MUST be ignored; defaults are not signaled or utilized here.

The ICE specification assumes that an ICE agent is configured with, or somehow knows of, TURN and STUN servers. Protocols using ICE need to describe how such information is learned by clients.

---

7. Sending an Initiate Message

TOC

Section 4.3 of ICE describes procedures for encoding the SDP. Instead of actually encoding an SDP, the candidate information (IP address and port and transport protocol, priority, foundation, component ID, type and related address) is carried within the object defined in Section 15 (The NICE Object). Similarly, the username fragment and password are carried in this object. This object does not contain any default candidates or the ice-lite attribute, as these features of ICE are not used in NICE. The object does contain a Next-Protocol field. This field is a string that contains the protocol name that is to be run over the

TCP or UDP association created by ICE. These names are drawn from the list of protocols defined by IANA at http://www.iana.org/assignments/port-numbers. Note that, since NICE will cause STUN and this protocol to be multiplexed on the same port, NICE can only be used to negotiate protocols that can be differentiated from STUN by inspection. If the desired protocol cannot be differentiated, it MUST be shimmed with a field that allows such differentiation, and the resulting protocol MUST be given a new name.

---

8. Receiving an Initiate Message

TOC

A responder MUST take the role of controlled. The role determination mechanism in Section 5.2 of ICE is not used with NICE. The ICE verification step in Section 5.1 is not used either. Instead, protocols using this specification will need to describe how to handle interoperability between clients which are using it, and ones which are not.

The responder follows the procedures in Section 6 (Gathering Candidates) to gather candidates. It then forms an Accept message and includes the object defined in Section 15 (The NICE Object).

The responder MUST follow the procedures in Section 5.7 and 5.8 of ICE, following the full implementation requirements and behaving as if there was a single media stream with a single component. Because there is only a single media stream and single component in NICE, the states described in Section 5.7.4 will become simplified. There will only be a single check list, and none of the candidate pairs will ever have the state of Frozen; all pairs will start in the Waiting state.

---

9. Receiving an Accept Message

TOC

When the initiator receives a response message, it extracts and NICE object from the message. The initiator MUST take the role of controlled, and then MUST follow the procedures of Section 5.7 and 5.8 of ICE, following the full implementation requirements and behaving as if there was a single media stream with a single component.

---

10. Connectivity Checks

TOC

The process of performing connectivity checks, as described in Section 7 of ICE, is used here without change. This means that STUN connectivity checks will contain the ICE-CONTROLLED and ICE-CONTROLLING

attributes. Strictly speaking, these are not needed. However, they are retained here to allow for the possibility of gatewaying between NICE and ICE (for example, in the event that H.323 decided to utilize NICE).

---

11. Concluding ICE

TOC

The controlling client MUST utilize regular nomination. This is to ensure consistent state on the final selected pairs without the need for additional signaling.
The procedures in Section 8 of ICE are followed to conclude ICE, with the following exceptions:

- The controlling agent MUST NOT attempt to send an updated offer once the state of its single media stream reaches Completed.

- Once the state of ICE reaches Completed, the agent can immediately free all unused candidates. This is because the concept of forking is not used here, and thus the three second delay in Section 8.3 of ICE does not apply.

---

12. Subsequent Messaging

TOC

A client MUST NOT send additional Initiate or Accept messages. Thus, the procedures in Section 9 of ICE MUST be ignored. A client that needs to modify its connection parameters in some way MUST establish a completely new connection by starting a totally new Initiate/Accept exchange and ICE connectivity checks.

---

13. Keepalives

TOC

A NICE client MUST utilize STUN for the keepalives described in Section 10 of ICE.

---

14. Sending and Receiving Data

TOC

A client follows the procedures of Section 11.1.1 of ICE to determine when it can proceed to send data. However, in this case, the "media" takes the form of application layer protocols. The concept of a

previous selected pair for a component does not apply to NICE, since ICE restarts are not used. A client MUST be prepared to receive data at any time.

---

15. The NICE Object

TOC

NICE operates by exchaning a MIME object, called the NICE object, in an initiate and response message. The syntax of that object is described here using the BNF defined in [RFC5234] (Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF," January 2008.).

```
NICE-obj    =     nice-ufrag CRLF
                  nice-pwd CRLF
                  nice-proto CRLF
                  1*(nice-cand CRLF)
                  *(nice-opts CRLF)
                  *(nice-ext CRLF)
nice-ufrag  =     ice-pwd-att
nice-pwd    =     ice-ufrag-att
nice-cand   =     candidate-attribute
nice-opts   =     ice-options
nice-proto  =     "nextproto:" token
nice-ext    =     ext-name ":" ext-value
ext-name    =     token
ext-value   =     byte-string
```

The BNF productions for ice-pwd-att, ice-ufrag-att, candidate-attribute and ice-options are defined in [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.). The NICE object also contains an extensibility mechanism, allowing new parameters to be defined which follow the form of name:value. The grammar for the name and its value follow those for SDP attributes. This allows for a direct copying of any future ICE-related SDP extensions into NICE without translations or specifications; the attribute is simply placed into the bottom of the NICE object using the grammar defined for it in the ICE extension.
The nextproto field contains an indication of the protocol that is to be multiplexed with STUN over the established connection. In some cases there is only one choice, based on the rendezvous protocol.
STUN connectivity checks between agents are authenticated using the short term credential mechanism defined for STUN [I-D.ietf-behave-rfc3489bis] (Rosenberg, J., Mahy, R., Matthews, P., and D. Wing, "Session Traversal Utilities for (NAT) (STUN)," July 2008.). This mechanism relies on a username and password that are exchanged through protocol machinery between the client and server.

With NICE, the Initiate and Accept exchange is used to exchange them. The username part of this credential is formed by concatenating a username fragment from each agent, separated by a colon. Each agent also provides a password, used to compute the message integrity for requests it receives. The username fragment and password are exchanged in the nice-ufrag and nice-pwd attributes, respectively. In addition to providing security, the username provides disambiguation and correlation of checks to media streams.

---

16. Security Considerations

TOC

ICE provides an extensive discussion on security considerations. That discussion applies here as well.
In particular, ICE security depends in part on message integrity and confidentiality of the offer/answer exchange. In the case of NICE, the rendezvous protocol carrying the ICE object needs to provide confidentiality and message integrity. Rendezvous protocols utilizing ICE MUST implement and SHOULD use some kind of mechanism to achieve that.

---

17. IANA Considerations

TOC

This specification registers a new MIME type, "message/nice", according to the procedures of RFC 2048 (Freed, N., Klensin, J., and J. Postel, “Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures,” November 1996.) [RFC2048]. This allows NICE to readily be used with protocols that provide MIME transport, though MIME transport is not required to use NICE.

> **MIME media type name:** message
>
> **MIME subtype name:** nice
>
> **Mandatory parameters:** None
>
> **Optional parameters:** None.
>
> **Encoding considerations:** None
>
> **Security considerations:** See Section 16 (Security Considerations) of RFC XXXX [[RFC EDITOR: Replace with RFC number of this specification]].
>
> **Interoperability considerations:** none.

Published specification:

RFC XXXX [[NOTE TO RFC EDITOR: Please replace XXXX with the published RFC number of this specification.]].

Applications which use this media type: This media type is used in the NICE protocol defined in in RFC XXXX [[NOTE TO RFC EDITOR: Please replace XXXX with the published RFC number of this specification.]].

Additional Information:

Magic Number: None

File Extension: .nic

Macintosh file type code: "TEXT"

Personal and email address for further information: Jonathan Rosenberg, jdrosen@jdrosen.net

Intended usage: COMMON

Author/Change controller: The IETF.

---

18. Tongue Twister

TOC

Say this five times fast: "ICE is nice, but NICE is nicer ICE".

---

19. References

TOC

19.1. Normative References

TOC

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

[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).
[RFC5234]	Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF," STD 68, RFC 5234, January 2008 (TXT).
[RFC2048]	Freed, N., Klensin, J., and J. Postel, "Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures," BCP 13, RFC 2048, November 1996 (TXT, HTML, XML).

19.2. Informative References

TOC

[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).

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

Copyright © The IETF Trust (2008).
This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.
This document and the information contained herein are provided on an “AS IS” basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.
Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.
The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.