Mobility with ICE (MICE)
draft-wing-mmusic-ice-mobility-01

Abstract

This specification describes how endpoint mobility can be achieved using ICE. Two mechanisms are shown, one where both endpoints support ICE and another where only one endpoint supports ICE.

Status of this Memo

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

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

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 17, 2013.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

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

1. Introduction ............................................. 3
2. Notational Conventions ................................. 3
3. Mobility using ICE ...................................... 4
   3.1. Gaining an Interface .............................. 4
   3.2. Receiving ICE Mobility event .................. 5
   3.3. Losing an Interface ............................... 6
   3.4. New STUN Attributes MOBILITY-EVENT and MOBILITY-SUPPORT . 6
4. Mobility using TURN ..................................... 6
   4.1. Creating an Allocation ............................ 7
      4.1.1. Sending an Allocate Request ................ 7
      4.1.2. Receiving an Allocate Request ............. 8
      4.1.3. Receiving an Allocate Success Response ...... 8
      4.1.4. Receiving an Allocate Error Response ...... 8
      4.2. Refreshing an Allocation ....................... 9
      4.2.1. Sending a Refresh Request .................. 9
      4.2.2. Receiving a Refresh Request ............... 9
      4.2.3. Receiving a Refresh Response ............... 9
      4.3. New STUN Attribute MOBILITY-TICKET ........... 10
   4.4. New STUN Error Response Code ..................... 10
5. IANA Considerations .................................... 10
6. Security Considerations ............................... 10
   6.1. Considerations for ICE mechanism ................. 10
   6.2. Considerations for TURN mechanism ............... 11
7. Acknowledgements ...................................... 11
8. References ............................................ 11
   8.1. Normative References ........................... 11
   8.2. Informative References .......................... 11
Authors’ Addresses ........................................ 11
1. Introduction

When moving between networks, an endpoint has to change its IP address. This change breaks upper layer protocols such as TCP and RTP. Various techniques exist to prevent this breakage, all tied to making the endpoint’s IP address static (e.g., Mobile IP, Proxy Mobile IP, LISP). Other techniques exist, which make the upper layer protocol ambivalent to IP address changes (e.g., SCTP). The mechanisms described in this document are in that last category.

ICE [RFC5245] ensures two endpoints have a working media path between them, and is typically used by Internet-connected interactive media systems (e.g., SIP endpoints). ICE does not expect either the local host or the remote host to change their IP addresses. Although ICE does allow an "ICE restart", this is done by sending a re-INVITE which goes over the SIP signaling path. The SIP signaling path is often slower than the media path (which needs to be recovered as quickly as possible), consumes an extra half round trip, and incurs an additional delay if the mobility event forces the endpoint to reconnect with its SIP proxy. Thus, this document attempts to perform mobility entirely on the media path.

A TURN [RFC5766] server relays media packets and is used for a variety of purposes, including overcoming NAT and firewall traversal issues and IP address privacy. The existing TURN specification does not allow the client address to change, especially if multiple clients share the same TURN username (e.g., the same credentials are used on multiple devices).

This document proposes two mechanisms to achieve RTP mobility: a mechanism where both endpoints support ICE, and a mechanism where only one endpoint supports ICE. When both endpoints support ICE, ICE itself can be used to provide mobility. When only one endpoint supports ICE, a TURN server provides mobility. Both mobility techniques work across and between network types (e.g., between 3G and wired Internet access), so long as the client can still access the remote ICE peer or TURN server.

Readers are assumed to be familiar with ICE [RFC5245].

2. Notational Conventions

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 note uses terminology defined in [RFC5245].
3. Mobility using ICE

When both endpoints support ICE, ICE itself can provide mobility functions. One of the primary aspects of ICE is its address gathering, wherein ICE has each endpoint determine all of the IP addresses and ports that might be usable for that endpoint and communicate that list of addresses and ports to its peer, usually over SDP. That enables the next primary aspect of ICE, which is its connectivity checks: each ICE endpoint sends a connectivity check to that list of addresses and ports. A connectivity check may unknowingly traverse a NAT, which means the ICE endpoint receiving the connectivity check cannot validate the source IP address or port of the connectivity against the list of IP addresses and ports provided by the ICE peer. In fact, if the source IP address and port is not known to the ICE endpoint, it is added to the list of candidates (Section 7.2.1.3 of [RFC5245]).

ICE Mobility takes advantage of that existent ICE functionality. Media can be switched to the new interface before or after the previous interface is lost.

When an interface is lost, media traffic might or might not be utilizing that interface. If media traffic is currently traversing the interface, this is considered a "break before make", because the host has not already moved its media traffic to a different interface.

Endpoints that support ICE Mobility perform ICE normally, and MUST also include the MOBILITY-SUPPORT attribute in all of their STUN requests and their STUN responses. The inclusion of this attribute allows the ICE peer to determine if it can achieve mobility using ICE or needs to use TURN (or needs to use some other mechanism, such as Mobile IP). To force the use of TURN to achieve ICE mobility, the ICE endpoint SHOULD NOT respond to ICE connectivity checks that have an IP address and port different from the TURN server, unless those connectivity checks contain the MOBILITY-SUPPORT attribute. In this way, the remote peer will think those other candidates are invalid (because its connectivity checks did not succeed).

After concluding ICE and moving to the ICE completed state (see Section 8 of [RFC5245]) either endpoint or both endpoints can initiate ICE Mobility, no matter if it was the Controlling Agent or the Controlled Agent during normal ICE processing.

3.1. Gaining an Interface

When gaining an interface which is suitable to send media by the host’s policy (if any), the ICE endpoint performs ICE Mobility. ICE
Mobility is performed by:

1. The ICE endpoint clears its ICE check list.

2. The ICE endpoint initiating an ICE connectivity check on the new interface, with the MOBILITY-EVENT attribute.

3. If this interface is the only suitable interface for media (that is, other suitable interfaces have been lost), then the connectivity check from the previous step SHOULD also include the USE-CANDIDATE attribute to signals an aggressive nomination (see Section 2.6 of [RFC5245]), and media MAY immediately begin flowing over that interface.

4. The ICE endpoint performs Sections 7.2.1.3, 7.2.1.4, and 7.2.1.5 of [RFC5245].

5. If the ICE connectivity check succeeds then ICE agents creates a new pair, adds the pair to the valid list and marks it as selected. The ICE agent can now send media using the newly selected candidate pair, even if it is running in Regular Nomination mode.

6. Once ICE connectivity checks for all of the media streams are completed, the controlling ICE endpoint follows the procedures in Section 11.1 of [RFC5245], specifically to send updated offer if the candidates in the m and c lines for the media stream (called the DEFAULT CANDIDATES) do not match ICE’s SELECTED CANDIDATES (also see Appendix B.9 of [RFC5245]).

3.2. Receiving ICE Mobility event

A STUN Binding Request containing the MOBILITY-EVENT attribute MAY be received by an ICE endpoint. If this is received before the endpoint is in the ICE Concluded state, it should be silently discarded.

The agent remembers the highest-priority nominated pairs in the Valid list for each component of the media stream, called the previous selected pairs. It continues sending media to that address until it finishes with the steps described below. Because those packets might not be received due to the mobility event, it MAY cache a copy of those packets.

The ICE endpoint clears its ICE check list.

The ICE endpoint performs Sections 7.2.1.3, 7.2.1.4, and 7.2.1.5 of [RFC5245].
3.3. Losing an Interface

When an interface is lost, the SDP MAY be updated, so that the remote ICE host does not waste its efforts with connectivity checks to that address, as those checks will fail. Because it can be argued that this is merely an optimization, and that the interface loss might be temporary (and soon regained), and that ICE has reasonable accommodation for candidates where connectivity checks timeout, this specification does not strongly encourage updating the SDP to remove a lost interface. Likewise, this specification recommends that ICE candidate addresses be maintained actively, subject to the host’s policy. For example, battery operated hosts have a strong incentive to not maintain mappings to TURN servers, as that maintenance requires periodic keepalive messages. As another example, a host that is receiving media over IPv6 may not want to persist with keeping a NATted IPv4 mapping alive (because that consumes a NAT mapping that could be more useful to a host actively utilizing the mapping for real traffic).

Note: this differs from Section 8.3 of [RFC5245], which encourages abandoning un-used candidates.

Note: A future version of this document will have more normative language in this section.

3.4. New STUN Attributes MOBILITY-EVENT and MOBILITY-SUPPORT

Two new attributes are defined by this section: MOBILITY-EVENT and MOBILITY-SUPPORT.

The MOBILITY-EVENT attribute indicate the sender experienced a mobility event. This attribute has no value, thus the attribute length field MUST always be 0. Rules for sending and interpretation of receiving are described above.

The MOBILITY-SUPPORT attribute indicates the sender supports ICE Mobility, as defined in this document. This attribute has no value, thus the attribute length field MUST always be 0. Rules for sending and interpretation of receiving are described above.

4. Mobility using TURN

To achieve mobility, a TURN client should be able to retain an allocation on the TURN server across changes in the client IP address as a consequence of movement to other networks.

When the client sends the initial Allocate request to the TURN
server, it will also include the new STUN attribute MOBILITY-TICKET (with zero length value), which indicates that the client is capable of mobility and desires a ticket. The TURN server provisions a ticket that is sent inside the new STUN attribute MOBILITY-TICKET in the Allocate Success response to the client. The ticket will be used by the client when it wants to refresh the allocation but with a new client IP address and port. It also ensures that the allocation can only be refreshed this way by the same client. When a client’s IP address changes due to mobility, it presents the previously obtained ticket in a Refresh Request to the TURN server. If the ticket is found to be valid, the TURN server will retain the same relayed address/port for the new IP address/port allowing the client to continue using previous channel bindings — thus, the TURN client does not need to obtain new channel bindings. Any data from external peer will be delivered by the TURN server to this new IP address/port of the client. The TURN client will continue to send application data to its peers using the previously allocated channelBind Requests.

4.1. Creating an Allocation

4.1.1. Sending an Allocate Request

In addition to the process described in Section 6.1 of [RFC5766], the client includes the MOBILITY-TICKET attribute with length 0. This
indicates the client is a mobile node and wants a ticket.

4.1.2. Receiving an Allocate Request

In addition to the process described in Section 6.2 of [RFC5766], the server does the following:

If the MOBILITY-TICKET attribute is included, and has length zero, and the TURN session mobility is forbidden by local policy, the server MUST reject the request with the new Mobility Forbidden error code. Following the rules specified in [RFC5389], if the server does not understand the MOBILITY-TICKET attribute, it ignores the attribute.

If the server can successfully process the request create an allocation, the server replies with a success response that includes a STUN MOBILITY-TICKET attribute. TURN server stores it’s session state, such as 5-tuple and NONCE, into a ticket that is encrypted by a key known only to the TURN server and sends the ticket in the STUN MOBILITY-TICKET attribute as part of Allocate success response.

The ticket is opaque to the client, so the structure is not subject to interoperability concerns, and implementations may diverge from this format. TURN Allocation state information is encrypted using 128-bit key for Advance Encryption Standard (AES) and 256-bit key for HMAC-SHA-256 for integrity protection.

4.1.3. Receiving an Allocate Success Response

In addition to the process described in Section 6.3 of [RFC5766], the client will store the MOBILITY-TICKET attribute, if present, from the response. This attribute will be presented by the client to the server during a subsequent Refresh request to aid mobility.

4.1.4. Receiving an Allocate Error Response

If the client receives an Allocate error response with error code TBD (Mobility Forbidden), the error is processed as follows:

- **TBD (Mobility Forbidden):** The request is valid, but the server is refusing to perform it, likely due to administrative restrictions. The client considers the current transaction as having failed. The client MAY notify the user or operator and SHOULD NOT retry the same request with this server until it believes the problem has been fixed.

All other error responses must be handled as described in [RFC5766].
4.2. Refreshing an Allocation

4.2.1. Sending a Refresh Request

If a client wants to refresh an existing allocation and update its time-to-expiry or delete an existing allocation, it will send a Refresh Request as described in Section 7.1 of [RFC5766]. If the client wants to retain the existing allocation in case of IP change, it will include the MOBILITY-TICKET attribute received in the Allocate Success response. If a Refresh transaction was previously made, the MOBILITY-TICKET attribute received in the Refresh Success response of the transaction must be used.

4.2.2. Receiving a Refresh Request

In addition to the process described in Section 7.2 of [RFC5766], the client does the following:

If the STUN MOBILITY-TICKET attribute is included in the Refresh Request then the server will not retrieve the 5-tuple from the packet to identify an associated allocation. Instead TURN server will decrypt the received ticket, verify the ticket’s validity and retrieve the 5-tuple allocation from the contents of the ticket. If this 5-tuple obtained from the ticket does not identify an existing allocation then the server MUST reject the request with an error.

If the source IP address and port of the Refresh Request is different from the stored 5-tuple allocation, the TURN server proceeds with checks to see if NONCE in the Refresh request is the same as the one provided in the ticket. The TURN server also uses MESSAGE-INTEGRITY validation to identify the that it is the same user which had previously created the TURN allocation. If the above checks are not successful then server MUST reject the request with a 441 (Wrong Credentials) error.

If all of the above checks pass, the TURN server understands that the client has moved to a new network and acquired a new IP address. The source IP address of the request could either be the host transport address or server-reflexive transport address. The server then updates it’s 5-tuple with the new client IP address and port. TURN server calculates the ticket with the new 5-tuple and sends the new ticket in the STUN MOBILITY-TICKET attribute as part of Refresh Success response.

4.2.3. Receiving a Refresh Response

In addition to the process described in Section 7.3 of [RFC5766], the client will store the MOBILITY-TICKET attribute, if present, from the
response. This attribute will be presented by the client to the server during a subsequent Refresh Request to aid mobility.

4.3. New STUN Attribute MOBILITY-TICKET

This attribute is used to retain an Allocation on the TURN server. It is exchanged between the client and server to aid mobility. The value is encrypted and identifies session state such as 5-tuple and NONCE. The value of MOBILITY-TICKET is a variable-length value.

4.4. New STUN Error Response Code

This document defines the following new error response code:

Mobility Forbidden: Mobility request was valid but cannot be performed due to administrative or similar restrictions.

5. IANA Considerations

IANA is requested to add the following attributes to the STUN attribute registry [iana-stun],

- MOBILITY-TICKET (0x802E, in the comprehension-optional range)
- MOBILITY-EVENT (0x802, in the comprehension-required range)
- MOBILITY-SUPPORT (0x8000, in the comprehension-optional range)

and to add a new STUN error code "Mobility Forbidden" with the value 501 to the STUN Error Codes registry [iana-stun].

6. Security Considerations

6.1. Considerations for ICE mechanism

A mobility event only occurs after both ICE endpoints have exchanged their ICE information. Thus, both username fragments are already known to both endpoints. Each endpoint contributes at least 24 bits of randomness to the ice-ufrag (Section 15.4 of [RFC5245]), which provides 48 bits of randomness. An off-path attacker would have to guess those 48 bits to cause the endpoints to perform HMAC-SHA1 validation of the MESSAGE-INTEGRITY attribute.

An attacker on the path between the ICE endpoints will see both ice-ufrags, and can cause the endpoints to perform HMAC-SHA1 validation
by sending messages from any IP address.

6.2. Considerations for TURN mechanism

TURN server MUST use strong encryption and integrity protection for the ticket to prevent an attacker from using a brute force mechanism to obtain the ticket’s contents or refreshing allocations.

Security considerations described in [RFC5766] are also applicable to this mechanism.

7. Acknowledgements

Thanks to Alfred Heggestad for his review and comments.

8. References

8.1. Normative References


8.2. Informative References

Authors’ Addresses

Dan Wing
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, California  95134
USA
Email: dwing@cisco.com

Prashanth Patil
Cisco Systems, Inc.
Cessna Business Park, Varthur Hobli
Sarjapur Marathalli Outer Ring Road
Bangalore, Karnataka  560103
India
Email: praspati@cisco.com

Tirumaleswar Reddy
Cisco Systems, Inc.
Cessna Business Park, Varthur Hobli
Sarjapur Marathalli Outer Ring Road
Bangalore, Karnataka  560103
India
Email: tireddy@cisco.com