A mechanism for conveying alternate addresses using ICE syntax
draft-hutton-mmusic-icemicrolite-00

Abstract

This document proposes a mechanism for conveying multiple IP
addresses, of different address families (e.g., IPv4, IPv6) for a
given medium, in the same Session Initiation Protocol (SDP) offer.
This proposed mechanism solves the backward compatibility which
exists with ANAT, due to its syntax, and provides a migration path
towards support for ICE. The proposed mechanism is significantly
less complex then ICE or ICE-Lite but uses ICE syntax. The mechanism
described in this document has been named ICE-microLite.

Status of this Memo

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

This Internet-Draft will expire on September 2, 2010.

Copyright Notice

Copyright (c) 2010 IETF Trust and the persons identified as the
document authors. All rights reserved.
1. 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] and indicate requirement levels for compliant mechanisms.

2. Introduction

This document proposes a mechanism which allows the carriage of multiple IP addresses, of different address families (e.g., IPv4, IPv6) for a given medium, in the same Session Description Protocol (SDP) [RFC4566] offer [RFC3264]. The proposed attribute solves the backward compatibility problem which exists with the earlier ANAT [RFC4091] mechanism, due to its syntax, and provides a migration path towards support for ICE [I-D.ietf-mmusic-ice]. The proposed mechanism is significantly less complex than ICE or ICE-Lite but uses ICE syntax. The mechanism described in this document has been named ICE-microLite. This mechanism should be considered as an alternative mechanism to that described in [I-D.boucadair-mmusic-altc].

2.1. Purpose

The purpose of this proposal is to provide a mechanism for conveying multiple media connection addresses for a given medium in an SDP offer in a way that is backwards compatible with existing SDP implementations. The proposed mechanism allows the receiver of the offer, if it understands the syntax, to select the media connection address which it prefers. If the receiver of the offer does not understand the proposed mechanism then it will use the SDP connection data ("c=") and therefore fall back to the default address family as preferred by the offerer, which for reasons of backwards compatibility will normally be an IPv4 address.

Currently the IETF intended mechanism for providing an IPv4 to IPv6 transition mechanism for conveying multiple media addresses in SDP is ICE acting as replacement for ANAT. However support for ICE or ICE-Lite, which is a subset of ICE, is over complex for scenarios in which a transition mechanism is needed without the need to use ICE for NAT traversal and connectivity checking.

The mechanism proposed in this document does however make use of ICE syntax, so providing the possibility for an implementation to be enhanced to support ICE or ICE-Lite without having to use a completely different syntax.

The proposed solution provides the following benefits:
allows a UA to signal more than one IP address (type) for the same medium in an SDP offer/answer;

is backwards compatible with non-ICE implementations because it uses the same backwards compatibility mechanism as ICE;

it is a lightweight mechanism that uses ICE syntax but does not require any of the complexities of ICE (E.g. Connectivity checking etc.);

by using ICE Syntax it provides a migration path towards support for ICE-Lite or full ICE.

2.2. Scope

This document proposes a simplified ICE-like syntax to carry several IP address types for the same medium in an SDP offer/answer while preserving backward compatibility.

The scope of this proposal is the same as [I-D.boucadair-mmusic-altc] and likewise does not include addressing at the SIP layer.

2.3. Use Cases

The use cases for this proposal are the same as the use cases detailed in [I-D.boucadair-mmusic-altc].

3. Overview of ICE-microLite

ICE-microLite is a migration step towards introduction of ICE-Lite. Compared to ICE-Lite, the functionality of an ICE-microLite implementation is further reduced such that:

an ICE-microLite implementation does not require a STUN server as no connectivity checking is performed.

an ICE-microLite implementation only provides candidates for a single foundation (I.e. RTP Component and RTCP Component) in an SDP answer.

it is a lightweight mechanism which uses ICE syntax but does not require any of the complexities of ICE (E.g. Connectivity checking etc.)

an ICE-microLite implementation uses dummy values for the STUN-related ICE attributes since STUN is not used. The attributes are still included to maintain compatibility with ICE.
An extension to the ICE a=candidate attribute provides the required functionality. For example:

```
a=candidate:1 1 UDP 2130706431 192.0.2.1 0 typ host microliteport 3478
```

This extension does not use the port field of the regular a=candidate attribute, but rather provides the port in an extension called microliteport. An ICE-microLite implementation will set the value of the regular port field to "0". Consequently, implementations which support ICE according to [I-D.ietf-mmusic-ice], but are not ICE-microLite aware will not find a default candidate matching port and IP address in the SDP’s m/c-line, will detect an ICE mismatch and will fall-back to default SDP processing.

An example of an SDP offer using this mechanism is as follows when IPv6 is preferred but IPv4 is the fall back option.

```
v=0
o=test 2890844342 2890842164 IN IP4 192.0.2.2
c=IN IP4 192.0.2.2
t=0 0
a=ice-lite
a=ice-pwd:microlitemicrolitemicrolite
a=ice-ufrag:microlite
m=audio 3480 RTP/AVP 0
b=RS:0
b=RR:0
a=candidate:1 1 UDP 2130706431 2001:::1 0 typ host microliteport 3478
a=candidate:2 1 UDP 2130706430 192.0.2.2 0 typ host microliteport 3480
```

4. Procedures

4.1. Sending the initial Offer

The procedures for sending the initial offer are the same as specified for an ICE-lite implementation in [I-D.ietf-mmusic-ice] except for the following:

- An ICE-microLite implementation is not required to use random values for the a=ice-ufrag and the a=ice-pwd attributes. An ICE-microLite implementation MUST use the following default values of "a=ice-pwd:microlitemicrolitemicrolite" and "a=ice-pwd:microlitemicrolitemicrolite".
ufrag:microlite". These attributes are not actually required by ICE-microLite but are maintained for compatibility with ICE implementations.

- An ICE-microLite implementation MUST set the port field in ICE a=candidate attribute to "0" and MUST include the local port of the candidate in the microliteport field of the candidate.

### 4.2. Receiving an SDP Offer

An ICE-microLite SIP UA MUST ignore any candidate-types that are not of type host.

If the a=candidate attributes for the host candidates in the received SDP offer do not contain the microliteport field in the ICE a=candidate attribute, the ICE-microLite SIP UA MUST ignore all other ICE attributes and process the SDP based on normal [RFC3264] procedures. Consequently, the SDP answer of the ICE-microLite SIP UA MUST NOT contain any ICE attribute in this case. This means that when the offer was generated by a UA which is not ICE-microLite aware the result will be a fall-back to the default candidates and normal SDP procedures.

If the a=candidate attributes for the host candidates in the received SDP offer contains the microliteport field, the ICE-microLite SIP UA MUST use that value as port number instead of using the value in the port field. An ICE-microLite SIP UA MUST ignore any port value received in the regular port field of the a=candidate attribute.

### 4.3. Sending an SDP Answer

An ICE-microLite implementation is not required to send the SDP answer in a 18x provisional response or send the response reliably as recommended by [I-D.ietf-mmusic-ice]. This is because no connectivity checking takes place.

An ICE-microLite implementation is not required to use random values for the a=ice-ufrag and the a=ice-pwd attributes as specified in [I-D.ietf-mmusic-ice]. An ICE-microLite implementation MUST use the following default values of "a=ice-pwd:microlitemicrolitemicrolite" and "a=ice-ufrag:microlite". These attributes are not actually required by ICE-microLite but are maintained for compatibility with ICE implementations.

An ICE-microLite SIP UA generating an SDP Answer MUST set the port field in the a=candidate attribute to "0" and MUST include the local port of the candidate in the microliteport field.
An ICE-microLite SIP UA MUST select its preferred candidate (E.g. based on IP address family) by listing only the selected candidate foundation in its SDP answer.

4.4. Receiving an SDP Answer

An ICE-microLite SIP UA will receive an SDP answer with only a single candidate pair per component from an ICE-microLite aware SIP UA compliant to this specification.

An SDP answer received from a SIP UA that is not ICE, ICE-lite or ICE-microLite will not contain ICE attributes other than "a=icemismatch".

5. Compatibility with ICE and ICE-Lite

An ICE or ICE-Lite implementation that is not ICE-microLite aware will detect an ice mismatch when receiving an ICE-microLite SDP offer and fallback to normal SDP processing.

An ICE or ICE-Lite implementation that is ICE-microLite aware will be aware that a received SDP offer is a ICE-microLite offer and respond according to the procedures in this document.

An ICE-microLite implementation that receives an SDP offer from a full ICE implementation will fallback to normal SDP processing.

An ICE-microLite implementatin that receives an SDP offer from an ICE-Lite implementation may respond with an ICE-Lite compatible answer. This is safe because the ICE-lite client will never initiate connectivity checking.

An ICE or ICE-Lite implementation that also supports ICE-microLite can include in the offer the microliteport extension which would allow an ICE-microLite implementation receiving the offer to generate an SDP answer also containing the microliteport extension.
6. Extension to ICE candidate attribute

The ICE [I-D.ietf-mmusic-ice] a=candidate attribute is extended as follows:

```
candidate-attribute = "candidate" ":" foundation SP component-id SP transport SP 
priority SP 
connection-address SP ;from RFC 4566 
port ;port from RFC 4566 
SP cand-type 
[SP rel-addr] 
[SP rel-port] 
[SP microliteport] 
*(SP extension-att-name SP extension-att-value) 

microliteport = "microliteport" SP port
```

7. IANA considerations

If this document moves forward, it requests a new extension attribute "microliteport", to be defined for the ICE candidate-attribute and a new ice-options attribute "microlite" to be reserved.

8. Security considerations

As with any SDP offer or answer, an attacker modifying the SDP can mount a variety of attacks. To counter this, SDP should be integrity protected by some means. Also an attacker eavesdropping on the SDP can discover addresses and ports on which to mount attacks (e.g., denial of service). To counter this, SDP should be confidentiality protected by some means. Such means will depend on the protocol used to convey SDP. For example, the Session Initiation Protocol (SIP) [RFC3261] can be transported over TLS or can protect SDP bodies using S/MIME.

9. Acknowledgements

The authors would like to acknowledge the assistance of Heinrich Haager, Thomas Stach and Ernst Horvath.
10. References

10.1. Normative References


10.2. Informative References


Authors’ Addresses

Andrew Hutton
Siemens Enterprise Communications
Phone: +44 1908 855395
Email: andrew.hutton@siemens-enterprise.com
URI: http://www.siemens-enterprise.com
John Elwell  
Siemens Enterprise Communications  

Phone:  +44 1908 855608  
Email:  john.elwell@siemens-enterprise.com  
URI:  http://www.siemens-enterprise.com