

Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: April 25, 2013

M. Boucadair  
France Telecom  
H. Kaplan  
Acme Packet  
R. Gilman  
Independent  
S. Veikkolainen  
Nokia  
October 22, 2012

**Session Description Protocol (SDP) Alternate Connectivity (ALTC)  
Attribute  
draft-boucadair-mmusic-altc-06.txt**

**Abstract**

This document proposes a mechanism which allows to carry multiple IP addresses, of different address families (e.g., IPv4, IPv6), in the same SDP offer. The proposed attribute solves the backward compatibility problem which plagued ANAT, due to its syntax.

**Requirements Language**

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](#) [[RFC2119](#)].

**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 April 25, 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

<a href="#">1.</a>	Introduction . . . . .	<a href="#">3</a>
<a href="#">1.1.</a>	Overall Context . . . . .	<a href="#">3</a>
<a href="#">1.2.</a>	Purpose . . . . .	<a href="#">4</a>
<a href="#">1.3.</a>	Scope . . . . .	<a href="#">5</a>
<a href="#">2.</a>	Use Cases . . . . .	<a href="#">5</a>
<a href="#">3.</a>	Overview of the ALTC Mechanism . . . . .	<a href="#">6</a>
<a href="#">3.1.</a>	Overview . . . . .	<a href="#">6</a>
<a href="#">3.2.</a>	Rationale for the Chosen Syntax . . . . .	<a href="#">7</a>
<a href="#">4.</a>	Alternate Connectivity Attribute . . . . .	<a href="#">8</a>
<a href="#">4.1.</a>	ALTC Syntax . . . . .	<a href="#">8</a>
<a href="#">4.2.</a>	Usage and Interaction . . . . .	<a href="#">9</a>
<a href="#">4.2.1.</a>	Usage . . . . .	<a href="#">9</a>
<a href="#">4.2.2.</a>	Usage of ALTC in an SDP Answer . . . . .	<a href="#">10</a>
<a href="#">4.2.3.</a>	Interaction with ICE . . . . .	<a href="#">10</a>
<a href="#">4.2.4.</a>	Interaction with SDP-Cap-Neg . . . . .	<a href="#">11</a>
<a href="#">5.</a>	IANA Considerations . . . . .	<a href="#">11</a>
<a href="#">6.</a>	Security Considerations . . . . .	<a href="#">11</a>
<a href="#">7.</a>	Acknowledgements . . . . .	<a href="#">11</a>
<a href="#">8.</a>	References . . . . .	<a href="#">11</a>
<a href="#">8.1.</a>	Normative References . . . . .	<a href="#">11</a>
<a href="#">8.2.</a>	Informative References . . . . .	<a href="#">11</a>
<a href="#">Appendix A.</a>	ALTC Use Cases . . . . .	<a href="#">13</a>
<a href="#">A.1.</a>	Terminology . . . . .	<a href="#">13</a>
<a href="#">A.2.</a>	Multicast Use Case . . . . .	<a href="#">14</a>
<a href="#">A.3.</a>	Introducing IPv6 into SIP-based Architectures . . . . .	<a href="#">15</a>
<a href="#">A.3.1.</a>	Avoid Crossing CGN Devices . . . . .	<a href="#">15</a>
<a href="#">A.3.2.</a>	Basic Scenario for IPv6 SIP Service Delivery . . . . .	<a href="#">15</a>
<a href="#">A.3.3.</a>	Avoid IPv4/IPv6 Interworking . . . . .	<a href="#">17</a>
<a href="#">A.3.4.</a>	DBE Bypass Procedure . . . . .	<a href="#">18</a>
<a href="#">A.3.5.</a>	Direct Communications Between IPv6-enabled User Agents . . . . .	<a href="#">20</a>
	Authors' Addresses . . . . .	<a href="#">21</a>



## **1. Introduction**

### **1.1. Overall Context**

Due to the IPv4 address exhaustion problem, IPv6 deployment is becoming an urgent need, along with the need to properly handle IPv6 and IPv4 co-existence. The reality of IPv4-IPv6 co-existence introduces heterogeneous scenarios with combinations of IPv4 and IPv6 nodes, some of which are capable of supporting both IPv4 and IPv6 dual-stack (DS) and some of which are capable of supporting only IPv4 or only IPv6. In this context, Session Initiation Protocol (SIP [[RFC3261](#)]) User Agents (UAs) need to be able to indicate their available IP capabilities in order to increase the ability to establish successful SIP sessions, and also to avoid invocation of adaptation functions such as Application Layer Gateways (ALGs) and IPv4-IPv6 interconnection functions (e.g., NAT64 [[RFC6146](#)]), and to avoid using private IPv4 addresses through consumer NATs or Carrier Grade NATs (CGN [[I-D.ietf-behave-lsn-requirements](#)]).

In the meantime, service providers are investigating scenarios to upgrade their service offering to be IPv6-capable. The current strategies involve either offering IPv6 only, for example to mobile devices, or providing both IPv4 and IPv6 but with private IPv4 addresses which are NAT'ed by CGNs. In the latter case the end device may be using "normal" IPv4 and IPv6 stacks and interfaces, or it may tunnel the IPv4 packets through a DS-Lite stack integrated into the host; in either case the device has both address families available from a SIP and media perspective.

Regardless of the IPv6 transition strategy being used, it is obvious that there will be a need for dual-stack SIP devices to communicate with IPv4-only legacy UAs, and IPv6-only UAs, and other dual-stack UAs. It may not, for example, be possible for a dual-stack UA to communicate with an IPv6-only UA unless the dual-stack UA had a means of providing the IPv6-only UA with its IPv6 local address for media, while clearly it needs to provide a legacy IPv4-only device its local IPv4 address. The communication must be possible in a backwards-compatible fashion, such that IPv4-only SIP devices need not support the new mechanism to communicate with dual-stack UAs.

The current means by which multiple address families can be communicated are through ANAT [[RFC4091](#)] or ICE [[RFC5245](#)]. ANAT has serious backwards-compatibility problems as described in [[RFC4092](#)], which effectively make it unusable, and it is deprecated by the IETF [[RFC5245](#)]. ICE at least allows interoperability with legacy devices, by not doing ICE in such cases, but it is a complicated and processing intensive mechanism, and has seen limited deployment and implementation in SIP applications. In some deployment models (e.g.,



closed networks), ICE is not usable at all.

The use of the ALTC solution is compliant with [[RFC6157](#)] which states:

"The use of ICE can be avoided for signaling messages that stay within such managed networks."

ALTC has been implemented as reported in [[I-D.boucadair-pcp-nat64-experiments](#)]; no issue has been reported in that document.

## **1.2. Purpose**

This document proposes a new alternative: a backwards-compatible syntax for indicating multiple media connection addresses and ports in an SDP offer, which can immediately be selected from and used in an SDP answer.

The proposed mechanism is independent of the model described in [[RFC5939](#)] and does not require implementation of sdp-capabilities-negotiations (a.k.a., sdp-cap-neg) to function. When sdp-cap-neg is supported, the CCAP attribute defined in [[I-D.garcia-mmusic-sdp-misc-cap](#)] should be used.

It should be noted that "backwards-compatible" in this document generally refers to working with legacy IPv4-only devices. The choice has to be made, one way or the other, because to interoperate with legacy devices requires constructing SDP bodies which they would understand and support, such that they detect their local address family in the SDP connection line. It is not possible to support interworking with both legacy IPv4-only and legacy IPv6-only devices with the same SDP offer. Clearly, there are far more legacy IPv4-only devices in existence, and thus those are the ones assumed in this document. However, the syntax allows for a UA to choose which address family to be backwards-compatible with, in case it has some means of determining it.

Furthermore, even for cases where both sides support the same address family, there should be a means by which the "best" address family transport is used, based on what the UAs decide. The address family which is "best" for a particular session cannot always be known a priori. For example, in some cases the IPv4 transport may be better, even if both UAs support IPv6.

The proposed solution provides the following benefits:



- o Allows a UA to signal more than one IP address (type) in the same SDP offer/answer;
- o Is backwards compatible. No parsing or semantic errors will be experienced by a legacy UA or intermediary SIP nodes which do not understand this new mechanism;
- o Is as lightweight as possible to achieve the goal, while still allowing and interoperating with nodes which support other similar or related mechanisms;
- o Is easily deployable in managed networks;
- o Requires minimal increase of the length of the SDP offer (I.e., minimizes fragmentation risks).

ALTC may also be useful for the multicast context (e.g., Section 3.4 of [[I-D.venaas-behave-v4v6mc-framework](#)]).

More detailed information about ALTC use cases is provided in [Appendix A](#).

### **[1.3](#). Scope**

This document proposes an alternative scheme, as replacement to the ANAT procedure, to carry several IP address types in the same SDP offer/answer while preserving backward compatibility.

While clearly two UAs communicating directly at a SIP layer need to be able to support the same address family for SIP itself, current SIP deployments almost always have Proxy Servers or B2BUA's in the SIP signaling path, which can provide the necessary interworking of the IP address family at the SIP layer. SIP-layer address family interworking is out of scope of this document (see [[I-D.boucadair-sipping-ipv6-atypes](#)] for a solution candidate). Instead, this document focuses on the problem of communicating media address family capabilities in a backwards-compatible fashion. Since media can go directly between two UAs, without a priori knowledge by the UAC of which address family the far-end UAS supports, it has to offer both, in a backwards-compatible fashion.

## **[2](#). Use Cases**

Although the ALTC mechanism defined in this document is meant for general use, the following use cases were explicitly considered:





- o A dual-stack UAC initiating a SIP session without knowing the address family of the ultimate target UAS.
- o A UA receiving a SIP session request with SDP offer and wishes to avoid using IPv4, or to avoid IPv6.
- o An IPv6-only UA wishes to avoid using a NAT64 [[RFC6146](#)].
- o A SIP UA behind a Dual-Stack Lite CGN [[RFC6333](#)].
- o A SIP Service Provider or Enterprise domain of IPv4-only and/or IPv6-only UA, which provides interworking by invoking IPv4-IPv6 media relays, wishes to avoid invoking such functions and let media go end-to-end as much as possible.
- o A SIP Service Provider or Enterprise domain of a UA, which communicates with other domains and wishes to either avoid invoking IPv4-IPv6 interworking or let media go end-to-end as much as possible.
- o A SIP Service Provider providing transit peering services for SIP sessions, which may need to modify SDP in order to provide IPv4-IPv6 interworking, but would prefer to avoid such interworking or avoid relaying media in general, as much as possible.
- o SIP sessions using the new mechanism crossing legacy SDP-aware middleboxes which may not understand this new mechanism.

### **[3.](#) Overview of the ALTC Mechanism**

#### **[3.1.](#) Overview**

The ALTC mechanism relies solely on the SDP offer/answer mechanism, with specific syntax to indicate alternative connection addresses. The basic concept is to use a new SDP attribute "altc", to indicate the IP addresses for potential alternative connection addresses. The address which is most likely to get chosen for the session is in the normal 'c=' line. Typically in current operational networks this would be an IPv4 address. The "a=altc" lines contain, in preference order, the alternative addresses offered for this session. This way, a dual-stack UA might encode its IPv4 address in the "c=" line, while possibly preferring to use an IPv6 address by indicating this by the "a=altc" attribute line ordering. One of the "a=altc" lines duplicates the address contained in the "c=" line, for reasons explained in [Section 3.2](#)). The SDP answerer would indicate its chosen address, by simply using that address family in the "c=" line of its response.



An example of an SDP offer using this mechanism is as follows when IPv4 is considered most likely to be used for the session, but IPv6 is preferred:

```
v=0
o=- 25678 753849 IN IP4 192.0.2.1
s=
c=IN IP4 192.0.2.1
t=0 0
m=audio 12340 RTP/AVP 0 8
a=altc IP6 2001:db8::1 45678
a=altc IP4 192.0.2.1 12340
```

If IPv6 was considered most likely to be used for the session, the SDP offer would be as follows:

```
v=0
o=- 25678 753849 IN IP6 2001:db8::1
s=
c=IN IP6 2001:db8::1
t=0 0
m=audio 12340 RTP/AVP 0 8
a=altc IP6 2001:db8::1 45678
a=altc IP4 192.0.2.1 12340
```

Since an alternative address is likely to require an alternative TCP/UDP port number as well, the new "altc" attribute includes both an IP address and a receive transport port number (or multiple port numbers). The ALTC mechanism does not itself support offering a different transport type (i.e., UDP vs. TCP), codec, nor any other attribute. It is only intended for offering an alternative IP address and port number.

### **3.2. Rationale for the Chosen Syntax**

The use of an 'a=' attribute line is, according to [\[RFC4566\]](#), the primary means for extending SDP and tailoring it to particular applications or media. A compliant SDP parser will ignore any session description that contains attribute lines it does not support.

The rationale for encoding the same address and port in the "a=altc" line as in the "m=" and "c=" lines is to provide detection of legacy SDP-changing middleboxes. Such systems may change the connection address and media transport port numbers, but not support this new mechanism, and thus two UAs supporting this mechanism would try to connect to the wrong addresses. Therefore, the rules detailed in this document require the SDP processor to check for matching altc



and connection line addresses and media ports, before choosing one of the alternatives.

## 4. Alternate Connectivity Attribute

### 4.1. ALTC Syntax

The altc attribute adheres to the [RFC4566] "attribute" production. The ABNF syntax [RFC5234] of altc is provided below:

```
altc-attr = "altc" att-value
att-value = addrtype SP connection-address SP port ["/" integer]
```

Figure 1: Connectivity Capability Attribute ABNF

The meaning of the fields are listed hereafter:

- o addrtype: the addrtype field as defined in [RFC4566] for connection data.
- o connection-address: a network address as defined in [RFC4566] corresponding to the address type specified by addrtype.
- o port: the port number to be used, as defined in [RFC4566]. Distinct port numbers may be used per IP address type. If the specified address type does not require a port number, a value defined for that address type should be used.

The "altc" attribute is only applicable in an SDP offer. The "altc" attribute is a media-level-only attribute, and MUST NOT appear at the SDP session level (since it defines a port number, it is inherently tied to the media level). There MUST NOT be more than one "altc" attribute per addrtype within each media description. This restriction is necessary in order that the addrtype of the reply may be used by the offerer to determine which alternative was accepted.

The <addrtype>'s of the altc MUST correspond to the <nettype> of the current connection (c=) line.

A media description MUST contain at least two "altc" attributes: the alternative address and port as well as an address and port which "duplicates" the address/port information from the current 'c=' and 'm=' lines. Each media level MUST contain at least one such duplicate altc attribute, of the same IP address family, address, and transport port number as those in the SDP connection and media lines of its level. In particular, if a 'c=' line appears within a media description, the addr-type and connection-address from that 'c=' line



MUST be used in the duplicate "altc" attribute for that media description. If a 'c=' line appears only at the session level and a given media description does not have its own connection line, then the duplicate "altc" attribute for that media description MUST be the same as the session-level address information.

The "altc" attributes appearing within a media description MUST be prioritized in order of appearance, with the first altc given highest priority and the following altc attributes prioritized in descending order. Given this rule, and the requirement that the address information provided in the "m=" line and "o=" line must be provided in an "altc" attribute as well, it is possible that the address in the "m=" line and "o=" line are not the preferred choice.

If the addrtype of an "altc" attribute is not compatible with the transport protocol or media format specified in the media description, that altc attribute MUST be ignored.

Note that "a=altc" lines describe alternative connection addresses, NOT addresses for parallel connections. When several altc lines are present, multiple sessions establishment MUST be avoided. Only one session is to be maintained with the remote party for the associated media description.

## **4.2. Usage and Interaction**

### **4.2.1. Usage**

In an SDP offer/answer model, the SDP offer includes "altc" attributes to indicate alternative connection information (i.e., address type, address and port number(s)), including the "duplicate" connection information already identified in the 'c=' and 'm=' lines.

Additional, subsequent offers MAY include "altc" attributes again, and may change the IP address, port numbers, and order of preference; but they MUST include a duplicate "altc" attribute for the connection and media lines in that specific subsequent offer. In other words, every offered SDP media description with an alternative address offer with an "altc" attribute has at least two of them:

- one duplicating the 'c=' and 'm=' line information for that media description, and
- one for each of the alternatives,

even though these need not be the same as the original SDP offer.

The purpose of encoding a duplicate "altc" attribute is to allow





receivers of the SDP offer to detect if a legacy SDP-changing middle box has modified the 'c=' and/or 'm=' line address/port information. If the SDP answerer does not find a duplicate "altc" attribute value for which the address and port match exactly those in the 'c=' line and 'm=' line, the SDP answerer MUST ignore the "altc" attributes and use the 'c=' and 'm=' offered address/ports for the entire SDP instead, as if no "altc" attributes were present. The rationale for this is that many SDP-changing middleboxes will end the media sessions if they do not detect media flowing through them; if a middlebox modified the SDP addresses, media MUST be sent using the modified information.

Note that for RTCP, if applicable for the given media types, each side would act as if the chosen "altc" attribute's port number was in the 'm=' media line. Typically, this would mean RTCP is sent to the odd +1 of the port number, unless some other attribute determines otherwise.

#### **4.2.2. Usage of ALTC in an SDP Answer**

The SDP answer SHOULD NOT contain "altc" attributes, as the answer's 'c=' line implicitly and definitively chooses the address family from the offer and includes it in "c=" and "m=" lines of the answer. Furthermore, this avoids establishing several sessions simultaneously between the participating peers.

Any solution requiring the use of ALTC in SDP answer SHOULD document its usage, in particular how sessions are established between the participating peers.

#### **4.2.3. Interaction with ICE**

Since ICE also includes address and port number information in its candidate attributes, a potential problem arises: which one wins. Since ICE also includes specific ICE attributes in the SDP answer, the problem is easily avoided: if the SDP offerer supports both ALTC and ICE, it may include both sets of attributes in the same SDP offer. A legacy ICE-only answerer will simply ignore the ALTC attributes, and use ICE. An ALTC-only answerer will ignore the ICE attributes and reply without them. An answerer which supports both MUST choose one and only one of the mechanisms to use: either ICE or ALTC (unless the 'm=' or 'c=' lines were changed by a middlebox, in which case the rules for both ALTC and ICE would make the answerer revert to basic SDP semantics).



#### **4.2.4. Interaction with SDP-Cap-Neg**

The ALTC mechanism is orthogonal to sdp-cap-neg. If the offerer supports both ALTC and sdp-cap-neg, it may offer both.

A method based on sdp-cap-neg is described in [[I-D.garcia-mmusic-sdp-misc-cap](#)] and may be used to specify different connectivity for alternative configurations.

### **5. IANA Considerations**

This document requests a new SDP attribute name "altc".

### **6. Security Considerations**

The security implications for ALTC are effectively the same as they are for SDP in general [[RFC4566](#)].

### **7. Acknowledgements**

Many thanks to T. Taylor for his review and comments.

### **8. References**

#### **8.1. Normative References**

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", [RFC 3261](#), June 2002.
- [RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", [RFC 4566](#), July 2006.
- [RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, [RFC 5234](#), January 2008.

#### **8.2. Informative References**

- [I-D.boucadair-pcp-nat64-experiments]  
Abdesselam, M., Boucadair, M., Hasnaoui, A., and J.



Queiroz, "PCP NAT64 Experiments",  
[draft-boucadair-pcp-nat64-experiments-00](#) (work in progress), September 2012.

[I-D.boucadair-sipping-ipv6-atypes]

Boucadair, M., Noisette, Y., and A. Allen, "The atypes media feature tag for Session Initiation Protocol (SIP)",  
[draft-boucadair-sipping-ipv6-atypes-02](#) (work in progress), July 2009.

[I-D.garcia-mmusic-sdp-misc-cap]

Garcia, M., Veikkolainen, S., and R. Gilman,  
"Miscellaneous Capabilities Negotiation in the Session Description Protocol (SDP)",  
[draft-garcia-mmusic-sdp-misc-cap-01](#) (work in progress), July 2009.

[I-D.ietf-behave-lsn-requirements]

Perreault, S., Yamagata, I., Miyakawa, S., Nakagawa, A., and H. Ashida, "Common requirements for Carrier Grade NATs (CGNs)", [draft-ietf-behave-lsn-requirements-09](#) (work in progress), August 2012.

[I-D.ietf-mboned-64-multicast-address-format]

Boucadair, M., Qin, J., Lee, Y., Venaas, S., Li, X., and M. Xu, "IPv6 Multicast Address With Embedded IPv4 Multicast Address",  
[draft-ietf-mboned-64-multicast-address-format-04](#) (work in progress), August 2012.

[I-D.jaclee-behave-v4v6-mcast-ps]

Jacquet, C., Boucadair, M., Lee, Y., Qin, J., and T. Tsou, "IPv4-IPv6 Multicast: Problem Statement and Use Cases", [draft-jaclee-behave-v4v6-mcast-ps-03](#) (work in progress), October 2011.

[I-D.venaas-behave-v4v6mc-framework]

Venaas, S., Li, X., and C. Bao, "Framework for IPv4/IPv6 Multicast Translation",  
[draft-venaas-behave-v4v6mc-framework-03](#) (work in progress), June 2011.

[RFC2871] Rosenberg, J. and H. Schulzrinne, "A Framework for Telephony Routing over IP", [RFC 2871](#), June 2000.

[RFC4091] Camarillo, G. and J. Rosenberg, "The Alternative Network Address Types (ANAT) Semantics for the Session Description Protocol (SDP) Grouping Framework", [RFC 4091](#), June 2005.



- [RFC4092] Camarillo, G. and J. Rosenberg, "Usage of the Session Description Protocol (SDP) Alternative Network Address Types (ANAT) Semantics in the Session Initiation Protocol (SIP)", [RFC 4092](#), June 2005.
- [RFC5245] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", [RFC 5245](#), April 2010.
- [RFC5853] Hautakorpi, J., Camarillo, G., Penfield, R., Hawrylyshen, A., and M. Bhatia, "Requirements from Session Initiation Protocol (SIP) Session Border Control (SBC) Deployments", [RFC 5853](#), April 2010.
- [RFC5939] Andreasen, F., "Session Description Protocol (SDP) Capability Negotiation", [RFC 5939](#), September 2010.
- [RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", [RFC 6146](#), April 2011.
- [RFC6157] Camarillo, G., El Malki, K., and V. Gurbani, "IPv6 Transition in the Session Initiation Protocol (SIP)", [RFC 6157](#), April 2011.
- [RFC6333] Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", [RFC 6333](#), August 2011.
- [RFC6406] Malas, D. and J. Livingood, "Session PEERing for Multimedia INTERconnect (SPEERMINT) Architecture", [RFC 6406](#), November 2011.

## [Appendix A](#). ALTC Use Cases

### [A.1](#). Terminology

The following terms are used:

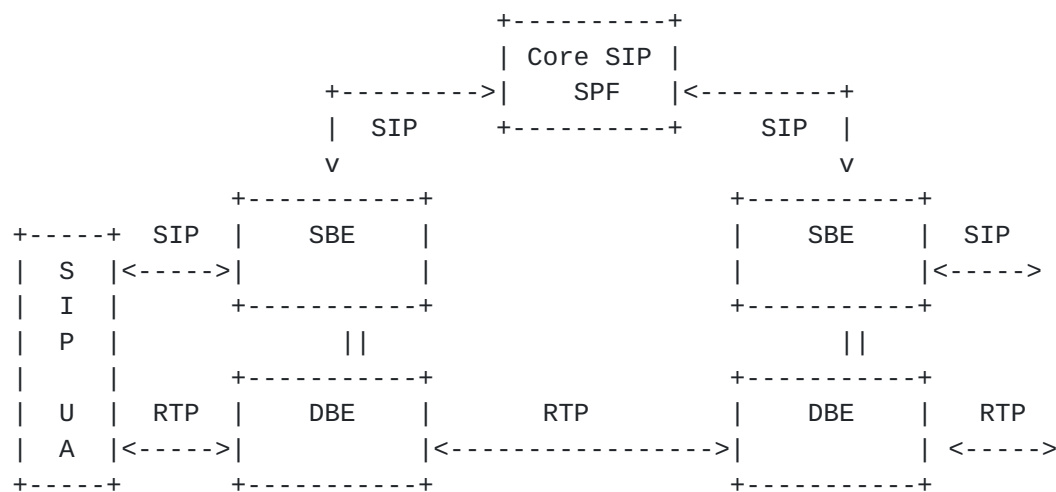
- o SBE (Signaling Path Border Element) denotes a functional element, located at the boundaries of an ITAD (IP Telephony Administrative Domain, [RFC2871](#)), which is responsible for intercepting signaling flows received from User Agents and relay them to the core service platform. A SBE may be located at the access segment (i.e., be the service contact point for User Agents) or be located at the interconnection with adjacent domains ([RFC6406](#)). A SBE





controls one or more DBEs. SBE and DBE may be located in the same device (e.g., SBC [[RFC5853](#)]) or be separated.

- o DBE (Data Path Border Element) denotes a functional element, located at the boundaries of an ITAD, which is responsible for intercepting media/data flows received from User Agents and relay them to another DBE (or media servers, e.g., announcement server or IVR). An example of DBE is a media gateway intercepting RTP flows. SBE may be located at the access segment (i.e., be the service contact point for User Agents) or be located at the interconnection with adjacent domains ([[RFC6406](#)]).
- o Core service platform is a macro functional block including session routing, interfaces to advanced services and access control. Figure 2 provides an overview of the overall architecture including SBE, DBE and Core service platform.



SIP UA can be embedded in the CPE or in a host behind the CPE

Figure 2: Service Architecture: Overview

## A.2. Multicast Use Case

Recently, a significant effort has been undertaken within IETF to specify new mechanisms to interconnect IPv6-only hosts to IPv4-only servers (e.g., [[RFC6146](#)]). This effort covered exclusively unicast transfer mode. An ongoing initiative, called multrans, has been launched to cover multicast issues to be encountered during IPv6 transition. The overall problem statement is documented in [[I-D.jaclee-behave-v4v6-mcast-ps](#)].

A particular issue encountered in the context of IPv4/IPv6 co-



existence and IPv6 transition of multicast services is the discovery of multicast group and source (refer to Section 3.4 of [[I-D.jaclee-behave-v4v6-mcast-ps](#)]):

1. An IPv6-only receiver requesting multicast content generated by an IPv4-only source:
  - (1.1) An ALG is required to help an IPv6 receiver to select the appropriate IP address when only the IPv4 address is advertised (e.g., using SDP); otherwise the access to the IPv4 multicast content can not be offered to the IPv6 receiver. The ALG may be located downstream the receiver. As such, the ALG does not know in advance whether the receiver is dual-stack or IPv6-only. The ALG may be tuned to insert both the original IPv4 address and corresponding IPv6 multicast address using for instance the ALTC SDP attribute.
  - (1.2) In order to avoid involving an ALG in the path, an IPv4-only source can advertise both its IPv4 address and IPv4-embedded IPv6 multicast address [[I-D.ietf-mboned-64-multicast-address-format](#)] using for instance the ALTC SDP attribute.
2. A dual-stack source sending its multicast content over IPv4 and IPv6: both IPv4 and IPv6 addresses need to be inserted in the SDP part. A means (e.g, ALTC) is needed for this purpose.

### **[A.3.](#) Introducing IPv6 into SIP-based Architectures**

#### **[A.3.1.](#) Avoid Crossing CGN Devices**

Some service providers are in the process of enabling DS-Lite [[RFC6333](#)] as a means to continue delivering IPv4 services to their customers. To avoiding crossing four levels of NAT when placing a media session (2 NAT in DS-Lite AFTR + 2 NAT in the DBE), it is recommended to enable IPv6 functions in some SBEs/DBEs. Therefore DS-Lite AFTRs won't be crossed for DS-Lite serviced customers if their UA is IPv6-enabled:

- o For SIP UA embedded in the CPE, this is easy to implement since the SIP UA [[RFC3261](#)] can be tuned to behave as IPv6-only UA when DS-Lite is enabled. No ALTC is required for that use case.
- o But for SIP User Agents located behind the CPE, a solution to indicate both IPv4 and IPv6 (e.g., ALTC) is required in order to avoid crossing the DS-Lite CGN.

#### **[A.3.2.](#) Basic Scenario for IPv6 SIP Service Delivery**

A basic solution to deliver SIP-based services using IPv4-only core service platform to IPv6-enabled UA is to enabled IPv4/IPv6



interworking function in SBE/DBE. Signaling and media between two SBEs and DBEs is maintained over IPv4. IPv6 is used between an IPv6-enabled UA and a SBE/DBE.

Figure 3 shows the results of session establishment between UAs. In this scenario, IPv4/IPv6 interworking function is invoked even when both involved UAs are IPv6-enabled.

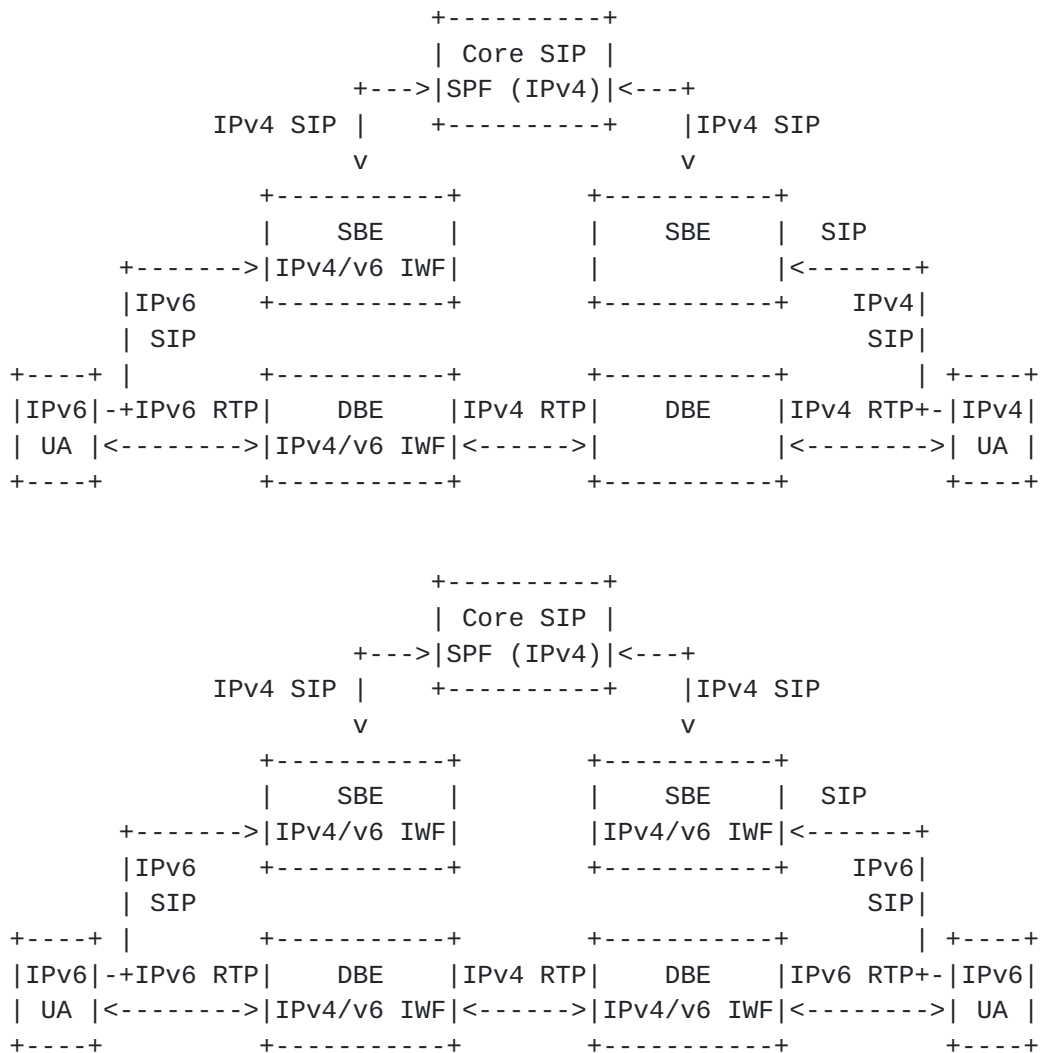


Figure 3: Basic scenario

Solutions to avoid redundant IPv4/IPv6 NAT and involving several DBEs may be valuable to consider by service providers.



### **A.3.3. Avoid IPv4/IPv6 Interworking**

For services providers wanting:

1. Means to promote the invocation of IPv6 transfer capabilities can be enabled while no parsing error is to be experienced by core service nodes legacy nodes
2. Optimize cost related to IPv4-IPv6 translation licenses
3. Reduce the dual-stack lifetime
4. Maintain an IPv4-only core
5. Only a set of SBE/DBE are IPv6-enabled

A solution to indicate both IPv4 and IPv6 addresses is required. This section provides an overview of this procedure:

When a SBE receives an INVITE, it instantiates in its DBE an IPv6-IPv6 context and an IPv6-IPv4 context. Both an IPv6 address and an IPv4 address are returned together with other information such as port numbers. SBE builds an SDP offer including both IPv4 and IPv6-related information using ALTC attribute. IPv6 is indicated as preferred connectivity type.

```
o=- 25678 753849 IN IP4 192.0.2.2
c=IN IP4 192.0.2.2
m=audio 12340 RTP/AVP 0 8
a=altc IP6 2001:db8::2 6000
a=altc IP4 192.0.2.2 12340
```

Figure 4: SDP offer updated by the SBE

The request is then forwarded to the core SPF which in its turn forwards the requests to the terminating SBE.

- o If this SBE is a legacy one, then it will ignore ALTC attributes and use "c" line.
- o If the terminating SBE is IPv6-enabled:
  - \* If the called UA is IPv4-only, then an IPv6-IPv4 context is created in the corresponding DBE.
  - \* If the called UA is IPv6-enabled, then an IPv6-IPv6 context is created in the corresponding DBE.

Figure 5 shows the result of the procedure when placing a session between an IPv4 and IPv6 UAs while Figure 6 shows the results of establishing a session between two IPv6-enabled UAs. The result is still not optima since redundant NAT66 is required (Appendix A.3.4).





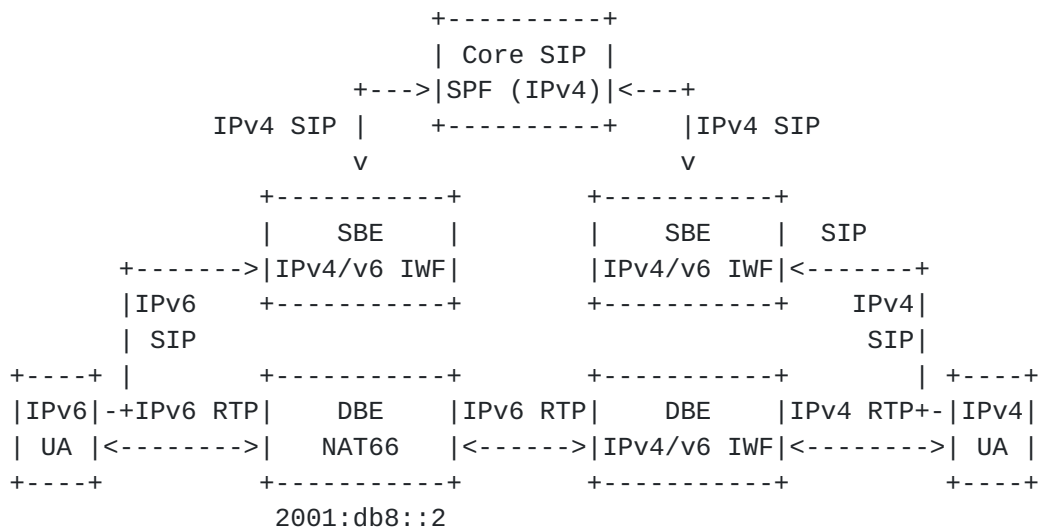


Figure 5: Session establishment between IPv4 and IPv6 UAs

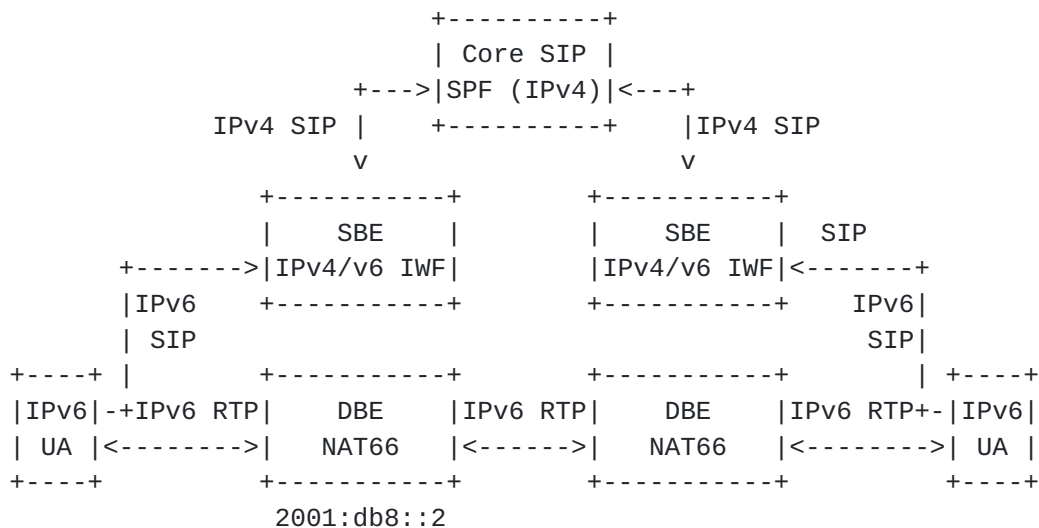


Figure 6: Session establishment between IPv6 UAs

#### A.3.4. DBE Bypass Procedure

For service providers wanting to involve only one DBE in the media path, when not all SBE/DBE and UAs are IPv6-enabled, a means to indicate both IPv4 and IPv6 addresses without inducing session failures is required. Below is proposed an example of a proposed procedure using ALTC attribute.

When the originating SBE receives an INVITE from an IPv6-enabled UA, it instantiates in its DBE an IPv6-IPv6 context and an IPv6-IPv4 context. Both an IPv6 address and an IPv4 address are returned together with other information such as port numbers. SBE builds an







- o No redundant NAT is maintained in the path when IPv6-enabled UAs are involved
- o End-to-end delay is optimized
- o The robustness of the service is optimized since the delivery of the service relies on fewer nodes
- o The signaling path is also optimized since no communication between the SBE (through SPDF in TISpan/IMS context) and DBE at the terminating side is required for some sessions.

#### **A.3.5. Direct Communications Between IPv6-enabled User Agents**

For service providers wanting to allow direct IPv6 communications between IPv6-enabled UAs, when not all SBE/DBE and UA are IPv6-enabled, a means to indicate both IPv4 and IPv6 addresses without inducing session failures is required. Below is proposed an example of a proposed procedure using ALTC attribute.

At the SBE originating side, when the SBE receives an INVITE from the calling IPv6 UA (Figure 9), it updates uses the ALTC to indicate two IP addresses:

1. An IPv4 address belonging to its controlled DBE
2. The same IPv6 address and port as received in the initial offer made by the calling IPv6

Figure 10 shows an excerpt example of the SDP offer generated by the originating SBE.

```
o=- 25678 753849 IN IP6 2001:db8::1
c=IN IP6 2001:db8::1
m=audio 12340 RTP/AVP 0 8
```

Figure 9: SDP offer of the calling UA

```
o=- 25678 753849 IN IP4 192.0.2.2
c=IN IP4 192.0.2.2
m=audio 12340 RTP/AVP 0 8
a=altc IP6 2001:db8::1 6000
a=altc IP4 192.0.2.2 12340
```

Figure 10: SDP offer updated by the SBE

The INVITE message will be routed appropriately to the destination SBE:

1. If the SBE is a legacy device (i.e., IPv4-only); it will ignore IPv6 addresses and contacts its DBE to instantiate an IPv4-IPv4 context.



2. If the SBE is IPv6-enabled, it will only forwards the INVITE to the address of contact of the called party:
  - A. If the called party is IPv6-enabled, the communication will be placed using IPv6. As such no DBE is involved in the data path as illustrated in Figure 11.
  - B. If not, IPv4 will be used between the originating DBE and called UA.

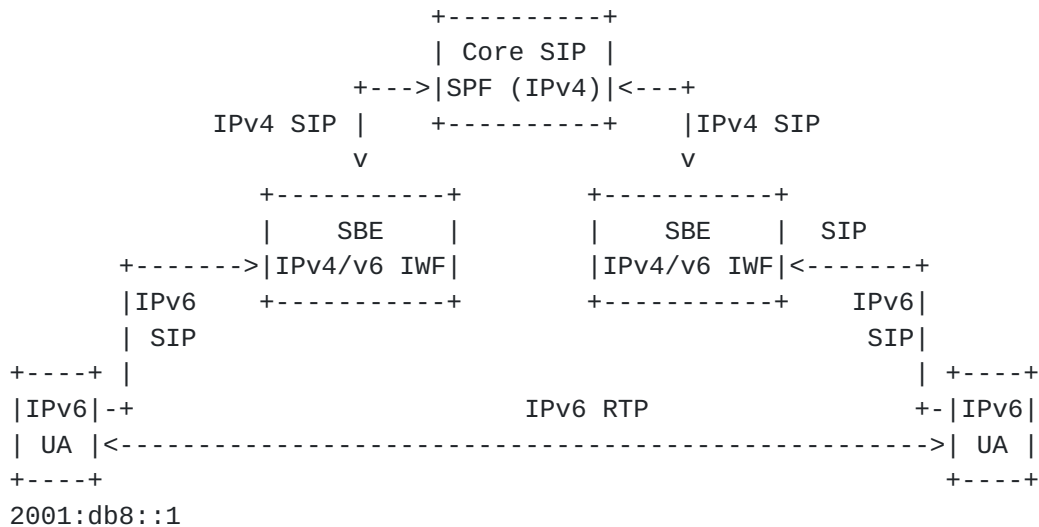


Figure 11: Direct IPv6 communication

## Authors' Addresses

Mohamed Boucadair  
 France Telecom  
 Rennes 35000  
 France

Email: mohamed.boucadair@orange.com

Hadriel Kaplan  
 Acme Packet  
 71 Third Ave.  
 Burlington, MA 01803  
 USA

Email: hkaplan@acmepacket.com





Robert R Gilman  
Independent

Email: [bob\\_gilman@comcast.net](mailto:bob_gilman@comcast.net)

URI:

Simo Veikkolainen  
Nokia

Email: [Simo.Veikkolainen@nokia.com](mailto:Simo.Veikkolainen@nokia.com)

URI: