Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 7, 2010

M. Boucadair
France Telecom
H. Kaplan
Acme Packet
July 06, 2009

Session Description Protocol (SDP) Connectivity Capability (CCAP) Attribute
draft-boucadair-mmusic-ccap-00.txt

Status of this Memo

Copyright Notice

Abstract

This memo proposes a mechanism which allows to carry multiple IP

addresses, of different address families (e.g., IPv4, IPv6), in the same SDP offer/answer. 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].

Table of Contents

1. Introduction

[Editorial Note: this section is lengthy/verbose, and is here simply to provide some initial background. This section, as well as Appendix-A, will be removed or at least reduced after initial draft versions.]

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, SIP 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 ALGs and NAT64, and to avoid using private IPv4 addresses through consumer NATs or Carrier-Grade NATs (CG-NAT).

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 CG-NATs. In the latter case the end device may be using "normal" IPv4 and IPv6 stacks and interfaces, or it may tunnel the IPv4 packets though 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 [I-D.ietf-mmusic-ice]. ANAT has serious backwards-compatibility problems as described in [RFC4092], which effectively make it unusable, and it is planned to be deprecated by the IETF. 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, ICE is not usable at all. Further details of why neither model is appropriate are described in Appendix A.

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 (called ccap) follows the model described in [I-D.ietf-mmusic-sdp-capability-negotiation] in syntax, but does not propose a full implementation of sdp-capabilities-negotiations (a.k.a., sdp-cap-neg) to function. If the full model is implemented, the mechanism proposed in this memo works with it as well, but orthogonally. The mechanism is an alternative to ICE, such that both mechanisms may be offered, but only one is chosen and 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. [Note: though this may be considered odd, it is technically and practically possible for certain devices to know such a thing in real-world deployments]

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. Which address family is "best" for a particular session is not defineable 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:

- Allows a UA to signal more than one IP address (type) in the same SDP offer/answer;

- Is backwards compatible. No parsing or semantic errors will be experienced by a legacy UA or intermediary nodes (e.g., Proxy Servers, Registrar Servers, etc.) which do not understand this new mechanism;

- Is as lightweight as possible to achieve the goal, while still allowing and interoperating with nodes which support other similar or related mechanisms.

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

- A dual-stack UAC initiating a SIP session without knowing the address family of the ultimate target UAS.

- A UA receiving a SIP session request with SDP offer and wishes to avoid using IPv4, or to avoid IPv6.

- An IPv6-only UA wishes to avoid using a NAT64.

- 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.

- 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.

- 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.

- SIP sessions using the new mechanism crossing legacy SDP-aware middleboxes which may not understand this new mechanism.

3. Overview of the CCAP Mechanism

3.1. Overview

The CCAP mechanism relies solely on the SDP offer/answer mechanism, with specific syntax to indicate capabilities. Following the sdp-cap-neg model, the basic concept is to use a new SDP attribute "ccap", to indicate the IP addresses for potential alternative connection addresses, while using the most likely-to-succeed address in the normal 'c=' connection line. Typically this would be an IPv4 address, however the new attribute also indicate if another address is more preferred. For example, a dual-stack UA might encode its IPv4 in the connection line, while possibly preferring to use an IPv6 address by indicating such in the attributes (though, it actually encodes both addresses in the attributes, for reasons explained later). The SDP answerer would indicate its chosen address, by simply using that address family in the SDP connection line of its response.

An example of SDP offer using this mechanism is as follows:

```
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=ccap:1 IP6 2001:db8::1 45678
a=ccap:2 IP4 192.0.2.1 12340
```

Since an alternative address is likely to require an alternative TCP/UDP port number as well, the new attribute includes both an IP address and a receive transport port number. The CCAP 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. The syntax of the attributes follows sdp-cap-neg and ICE in some regards, but does not require support for either of them. Other mechanisms, such as sdp-cap-neg, may be used at the same time to offer other alternative semantics, but they are orthogonal to the address and port alternatives in this memo.

3.2. Rationale

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. An SDP parser will ignore any session description that contains attribute lines it does not support. [Note: of course some devices in the wild may not ignore unknown attributes, but then it is not compliant with SDP rules, and nothing will help it]

The rationale for encoding the same address/port as in the media and connection 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 ccap and connection line addresses and media ports, before choosing one of the alternatives.

4. Connectivity Capability Attribute

4.1. CCAP Syntax

The ccap attribute adheres to the RFC 4566 "attribute" production. The ABNF syntax of ccap is provided below:

```
   ccap-attr     = "ccap" ":" att-value
   att-value     = addr-cap-num SP addrtype SP connection-address SP
port ;defined in [RFC4566]
   addr-cap-num  = 1*DIGIT ;defined in [RFC5234]
```

Figure 1: Connectivity Capability Attribute ABNF

Note that white space is not permitted before the addr-cap-num.

The meaning of the fields are listed hereafter:

- addr-cap-num: digit to uniquely refer to an address alternative. It must be in preference order (1=most-preferred).

- addrtype: the addrtype field as defined in [RFC4566] for connection data.

- connection-address: an IPv4 or IPv6 address as defined in [RFC4566].

- port: the port number to be used, as defined in [RFC4566]. Distinct port numbers may be used per IP address type.

The ccap attribute is only applicable in an SDP offer. The ccap attribute 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 ccap attribute per IP Address family, per media level. Each and every media level MUST contain exactly two ccap attributes: one for one address family, and a second for the other.

This document's mechanism requires a "duplicate" ccap attribute to be included, with the same address/port information as in the RFC 4566 base SDP 'c=' connection and 'm=' media lines. Each media level MUST contain at least one such duplicate ccap attribute, of the same IP address family, address, and transport port number as those in the SDP connection and media lines of its level.

If a 'c=' connection line appears at the media level, the same address as that 'c=' line MUST be used in the duplicate ccap attribute for that media level.

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

When several ccap lines are present, multiple sessions establishment MUST be avoided. Only one session is to be maintained with remote party.

4.2. Usage and Interaction

4.2.1. Usage

In an SDP offer/answer model, the SDP offer would include ccap attributes to indicate alternative connection information (i.e., address family, address and port number), as well as the "duplicate" connection information already identified in the 'c=' connection and 'm=' media lines. The SDP answer MUST NOT contain ccap attributes,

as the answer's 'c=' line implicitly and definitively "chooses" the address family from the offer.

Additional, subsequent offers MAY include ccap attributes again, and change the IP address, ports, and order of preference; but they MUST include a duplicate ccap attribute of the connection and media lines in that specific subsequent offer. In other words, every SDP offer with a ccap attribute has two of them:

- one duplicating the 'c=' and 'm=' line information in that SDP offer, and

- one for the alternative, even though both of those need not be the same as the original SDP offer.

The purpose of encoding a "duplicate" ccap 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 ccap attribute value for which the address and port match exactly those in the 'c=' line and 'm=' line, the SDP answerer MUST ignore the ccap attributes and use the 'c=' and 'm=' offered address/ports for the entire SDP instead, as if no ccap attributes were present. The rationale for this is that many SDP-changing middleboxes will end the SIP session if they do not detect media flowing through them; if a middlebox modified the SDP, 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 ccap 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. 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 CCAP and ICE, it may include both sets of attributes in the same SDP offer. A legacy ICE-only answerer will simply ignore the CCAP attributes, and use ICE. A CCAP-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 CCAP (unless the 'm=' or 'c=' lines were changed by a middlebox, in which case the rules for both CCAP and ICE would make the answerer revert to basic SDP semantics).

4.2.3. Interaction with SDP-Cap-Neg

The CCAP mechanism is orthogonal to sdp-cap-neg. If the offerer supports both ccap and sdp-cap-neg, it may offer both. At this time, sdp-cap-neg does not provide a means of offering alternative addresses/ports, other than through ICE, for which the behavior was described previously. Therefore, there is no conflicting interaction. CCAP capabilities are not negotiated as part of the potential and actual configuration attribute syntax and semantics defined in [I-D.ietf-mmusic-sdp-capability-negotiation].

[Note: it was tempting to in fact make CCAP be yet another set of alternative capabilities in an sdp-cap-neg, but the complexities of sdp-cap-neg, and the subtleties of potentially tying address/port options with media capabilities do not seem to be worth the effort for this case]

5. The CCAP Option Tag

This document defines a new SIP option-tag for use in the "Supported" SIP header field called "ccap". This option-tag is for the purpose of indicating that a UA supports the CCAP mechanism defined in this document AND actually has multiple address family addresses available, in order to improve troubleshooting, and in some cases provide a hint to other nodes that the UA is capable of both IPv4 and IPv6 and CCAP.

A UA MUST NOT include this option tag unless it both (1) supports the CCAP mechanism AND (2) has *both* an IPv4 and IPv6 address available for media use. The reason it only includes the ccap option-tag if it actually has both addresses, is that having only a single address family available implies the UA cannot truly perform CCAP in an offer; it may have the necessary logic to, but it does not have the addresses to do so. (remember one does not include the ccap attribute in SDP unless one has both address families available)

A UA SHOULD include the CCAP option-tag in a "Supported" SIP header field in SIP REGISTER, OPTIONS, and INVITE requests and related responses, if it has both address-family addresses available and supports the CCAP mechanism. A UA MUST NOT include the CCAP option-tag in the "Require" or "Proxy-Require" SIP header fields under any conditions.

6. IANA Considerations

If this document moves forward, it requests a new SDP attribute name

"ccap", as defined earlier; and a new SIP option-tag be reserved, named "ccap", for the purposes described earlier.

7. Security Considerations

The security implications for CCAP are effectively the same as they are for SDP in general.

8. Acknowledgements

TBC

9. References

9.1. 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.

[RFC3388] Camarillo, G., Eriksson, G., Holler, J., and H. Schulzrinne, "Grouping of Media Lines in the Session Description Protocol (SDP)", RFC 3388, December 2002.

[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.

[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.

9.2. Informative References

[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-01 (work in progress), March 2009.

[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.

[I-D.ietf-mmusic-sdp-capability-negotiation]
Andreasen, F., "SDP Capability Negotiation", draft-ietf-mmusic-sdp-capability-negotiation-10 (work in progress), May 2009.

Appendix A. ANAT and ICE

A.1. ANAT

[RFC4091] describes a mechanism allowing multiple alternative network addresses to be enclosed in a single SDP offer/answer. This proposal consists at introducing a new attribute called ANAT (Alternative Network Address Types). ANAT is based on media grouping [RFC3388]. ANAT specification lists IP address as an example of Network Address (without providing other examples).

This attribute allows inserting multiple media/connection lines in the same SDP offer (or SDP answer). [RFC4092] defines how SIP can exploit the ANAT semantic by introducing a new option tag called "sdp-anat". This tag can be useful for SIP UAs to be aware of the capabilities of each other and then select from the supported media/network description lines the ones that are suitable for setting up the SIP communication (and also according to local preferences). A use case for illustrating the usage of this tag is a Dual Stack SIP UA which can communicate either using its IPv6 or its IPv4 connectivity. This type of SIP UAs can also set a preference associated with each type of enclosed connectivity type.

[RFC4092] states that answerers without support for ANAT will react in different ways upon receipt of an offer using ANAT and different implementations will behave in different ways. This issue is a real problem in current operational SIP-based service offerings. Indeed, in order to support IPv6 is SIP-based architectures, several

scenarios may be envisaged. The most pragmatic one is to update the access segment to support IPv6. The support of ANAT is such situations would encounter backward compatibility issues since core service nodes are not ANAT-compliant. This limitation may be a hurdle for the use of IPv6 and particularly to activate policies to encourage the usage of IPv6 and to guarantee successful communications involving heterogeneous (i.e. IPv4 and IPv6) parties.

Unfortunately, even with the sdp-anat option tag addition, ANAT is not truly usable in modern SIP usage. In most SIP usage today, the SIP UAC generates the SDP offer in its initial INVITE request. Since it does not know the capabilities of the ultimate far-end UAS, it cannot include ANAT syntax in its SDP due to the backwards-compatibility problem. Inserting an sdp-anat option tag in the Require header would lead to numerous failed INVITE attempts. Inserting it in the Supported header would only allow it to re-negotiate SDP using ANAT afterwards, which would lead to failed initial INVITE requests if it chose to offer an address family initially that the far-end could not support . Neither case is attractive, and in particular failed INVITE attempts are highly undesirable if not outright unacceptable, leaving the UAC with no choice but to either send an offer-less INVITE, or simply assume IPv4. Assuming IPv4 does not solve the address transition problem, as it would require all devices to continue to use IPv4 indefinitely, and offer-less INVITEs have well-known interoperability problems in practice.

Note that ANAT specification is to be deprecated by ICE [I-D.ietf-mmusic-ice].

A.2. ICE

ICE solves the IPv4-v6 SDP offer problem by having the UAC offer both addresses as alternative candidates in the SDP offer. If the far-end UAS supports ICE, it can choose among them; else it will simply use the one offered in the normal SDP connection line. If ICE is supported, STUN connectivity checks are performed, in a controlled fashion, along with an additional round of SDP negotiation for the final chosen connection path.

This solves the problem of backwards-compatibility, but at a heavy price: both sides must implement ICE. While ICE provides other benefits, specifically basic NAT traversal without the aid of middleboxes other than TURN servers, it is complicated and difficult to troubleshoot. It is a very high bar to place on SIP UAs, just to achieve IP address family negotiation. What's needed is as simple a mechanism as possible to achieve the goals, in order to provide reasonable chance of widespread adoption and deployment.

Furthermore, ICE does not work in some scenarios. In particular, it does not work when the address(es) are determined based on where the SIP session "goes". ICE assumes there may be many layers of NAT, but that they all cascade from a private to public side, towards the public Internet, and assume that both SIP UAs (ICE clients) can obtain addresses in the public Internet (e.g., through TURN servers), which can be used as a last resort point of media packet rendezvous. Such is not always the case. For example, in common SIP Provider peering arrangements, a SIP UAC is in a private network of one Provider and the UAS in a private network of the other Provider, and media communication does not and cannot cross the public Internet. Multi-hop transit peering cases exacerbate this issue even further. The only devices with knowledge of the correct addresses to use in such scenarios are middleboxes, and they do not know the addresses until the SIP session is initially signaled (and in fact the addresses may change during the session's lifetime).

For the sole purpose of negotiating IP address families, therefore, ICE is neither necessary nor sufficient.

Authors' Addresses

Mohamed Boucadair
France Telecom
3, Av Francois Chateau
Rennes 35000
France

Email: mohamed.boucadair@orange-ftgroup.com

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

Email: hkaplan@acmepacket.com