Internet Engineering Task Force (IETF)
Request for Comments: 6157
Updates: 3264
Category: Standards Track
ISSN: 2070-1721

G. Camarillo
Ericsson
K. El Malki
Athonet
V. Gurbani
Bell Labs, Alcatel-Lucent

April 2011

IPv6 Transition in the Session Initiation Protocol (SIP)

Abstract

This document describes how the IPv4 Session Initiation Protocol (SIP) user agents can communicate with IPv6 SIP user agents (and vice versa) at the signaling layer as well as exchange media once the session has been successfully set up. Both single- and dual-stack (i.e., IPv4-only and IPv4/IPv6) user agents are considered.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc6157.

Table of Contents

1. Introduction

SIP [3] is a protocol to establish and manage multimedia sessions. After the exchange of signaling messages, SIP endpoints generally exchange session or media traffic, which is not transported using SIP but a different protocol. For example, audio streams are typically carried using the Real-Time Transport Protocol (RTP) [13].

Consequently, a complete solution for IPv6 transition needs to handle both the signaling layer and the media layer. While unextended SIP can handle heterogeneous IPv6/IPv4 networks at the signaling layer as long as proxy servers and their Domain Name System (DNS) entries are properly configured, user agents using different networks and address spaces must implement extensions in order to exchange media between them.

This document addresses the system-level issues in order to make SIP work successfully between IPv4 and IPv6. Sections 3 and 4 provide discussions on the topics that are pertinent to the signaling layer and media layer, respectively, to establish a successful session between heterogeneous IPv4/IPv6 networks.

2. Terminology

In this document, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in BCP 14, RFC 2119 [1] and indicate requirement levels for compliant implementations.

IPv4-only user agent: An IPv4-only user agent supports SIP signaling and media only on the IPv4 network. It does not understand IPv6 addresses.

IPv4-only node: A host that implements only IPv4. An IPv4-only node does not understand IPv6. The installed base of IPv4 hosts existing before the transition begins are IPv4-only nodes.

IPv6-only user agent: An IPv6-only user agent supports SIP signaling and media only on the IPv6 network. It does not understand IPv4 addresses.

IPv6-only node: A host that implements IPv6 and does not implement IPv4.

IPv4/IPv6 node: A host that implements both IPv4 and IPv6; such hosts are also known as "dual-stack" hosts [17].

IPv4/IPv6 user agent: A user agent that supports SIP signaling and media on both IPv4 and IPv6 networks.

IPv4/IPv6 proxy: A proxy that supports SIP signaling on both IPv4 and IPv6 networks.

3. The Signaling Layer

An autonomous domain sends and receives SIP traffic to and from its user agents as well as to and from other autonomous domains. This section describes the issues related to such traffic exchanges at the signaling layer, i.e., the flow of SIP messages between participants in order to establish the session. We assume that the network administrators appropriately configure their networks such that the

SIP servers within an autonomous domain can communicate between themselves. This section contains system-level issues; a companion document [15] addresses IPv6 parser torture tests in SIP.

3.1. Proxy Behavior

User agents typically send SIP traffic to an outbound proxy, which takes care of routing it forward. In order to support both IPv4-only and IPv6-only user agents, it is RECOMMENDED that domains deploy dual-stack outbound proxy servers or, alternatively, deploy both IPv4-only and IPv6-only outbound proxies. Furthermore, there SHOULD exist both IPv6 and IPv4 DNS entries for outbound proxy servers. This allows the user agent to query DNS and obtain an IP address most appropriate for its use (i.e., an IPv4-only user agent will query DNS for A resource records (RRs), an IPv6-only user agent will query DNS for AAAA RRs, and a dual-stack user agent will query DNS for all RRs and choose a specific network.)

Some domains provide automatic means for user agents to discover their proxy servers. It is RECOMMENDED that domains implement appropriate discovery mechanisms to provide user agents with the IPv4 and IPv6 addresses of their outbound proxy servers. For example, a domain may support both the DHCPv4 [11] and the DHCPv6 [10] options for SIP servers.

On the receiving side, user agents inside an autonomous domain receive SIP traffic from sources external to their domain through an inbound proxy, which is sometimes co-located with the registrar of the domain. As was the case previously, it is RECOMMENDED that domains deploy dual-stack inbound proxies or, alternatively, deploy both IPv4-only and IPv6-only inbound proxy servers. This allows a user agent external to the autonomous domain to query DNS and receive an IP address of the inbound proxy most appropriate for its use (i.e., an IPv4-only user agent will query DNS for A RRs, an IPv6-only user agent will query DNS for AAAA RRs, and a dual-stack user agent will query DNS for all RRs and choose a specific network). This strategy, i.e., deploying dual-stack proxies, also allows for an IPv6-only user agent in the autonomous domain to communicate with an IPv4-only user agent in the same autonomous domain. Without such a proxy, user agents using different networks identifiers will not be able to successfully signal each other.

Proxies MUST follow the recommendations in Section 5 to determine the order in which to contact the downstream servers when routing a request.

3.1.1. Relaying Requests across Different Networks

A SIP proxy server that receives a request using IPv6 and relays it to a user agent (or another downstream proxy) using IPv4, and vice versa, needs to remain in the path traversed by subsequent requests. Therefore, such a SIP proxy server MUST be configured to Record-Route in that situation.

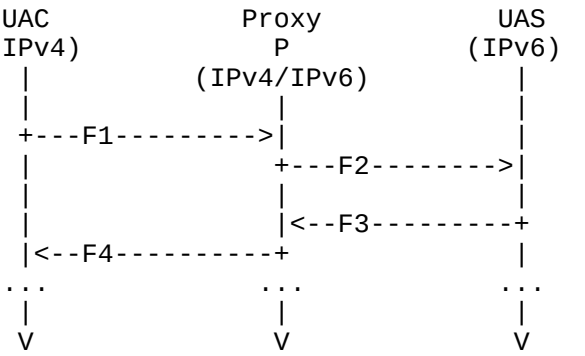
> Note that while this is the recommended practice, some problems may still arise if an RFC 2543 [14] endpoint is involved in signaling. Since the ABNF in RFC 2543 did not include production rules to parse IPv6 network identifiers, there is a good chance that an RFC 2543-only compliant endpoint is not able to parse or regenerate IPv6 network identifiers in headers. Thus, despite a dual-stack proxy inserting itself into the session establishment, the endpoint itself may not succeed in the signaling establishment phase.
>
> This is generally not a problem with RFC 3261 endpoints; even if such an endpoint runs on an IPv4-only node, it still is able to parse and regenerate IPv6 network identifiers.

Relaying a request across different networks in this manner has other ramifications. For one, the proxy doing the relaying must remain in the signaling path for the duration of the session; otherwise, the upstream client and the downstream server would not be able to communicate directly. Second, to remain in the signaling path, the proxy MUST insert one or two Record-Route headers: if the proxy is inserting a URI that contains a Fully Qualified Domain Name (FQDN) of the proxy, and that name has both IPv4 and IPv6 addresses in DNS, then inserting one Record-Route header suffices. But if the proxy is inserting an IP address in the Record-Route header, then it must insert two such headers; the first Record-Route header contains the proxy's IP address that is compatible with the network type of the downstream server, and the second Record-Route header contains the proxy's IP address that is compatible with the upstream client.

An example helps illustrate this behavior. In the example, we use only those headers pertinent to the discussion. Other headers have been omitted for brevity. In addition, only the INVITE request and final response (200 OK) are shown; it is not the intent of the example to provide a complete call flow that includes provisional responses and other requests.

In this example, proxy P, responsible for the domain example.com, receives a request from an IPv4-only upstream client. It proxies this request to an IPv6-only downstream server. Proxy P is running on a dual-stack host; on the IPv4 interface, it has an address of

192.0.2.1, and on the IPv6 interface, it is configured with an address of 2001:db8::1 (Appendix A contains a sample DNS zone file entry that has been populated with both IPv4 and IPv6 addresses.)




```
F1: INVITE sip:alice@example.com SIP/2.0
    ...

F2: INVITE sip:alice@2001:db8::10 SIP/2.0
    Record-Route: <sip:2001:db8::1;lr>
    Record-Route: <sip:192.0.2.1;lr>
    ...

F3: SIP/2.0 200 OK
    Record-Route: <sip:2001:db8::1;lr>
    Record-Route: <sip:192.0.2.1;lr>
    ...

F4: SIP/2.0 200 OK
    Record-Route: <sip:2001:db8::1;lr>
    Record-Route: <sip:192.0.2.1;lr>
    ...
```

Figure 1: Relaying requests across different networks

When the User Agent Server (UAS) gets an INVITE and it accepts the invitation, it sends a 200 OK (F3) and forms a route set. The first entry in its route set corresponds to the proxy's IPv6 interface. Similarly, when the 200 OK reaches the User Agent Client (UAC) (F4), it creates a route set by following the guidelines of RFC 3261 and reversing the Record-Route headers. The first entry in its route set corresponds to the proxy's IPv4 interface. In this manner, both the UAC and the UAS will have the correct address of the proxy to which they can target subsequent requests.

Alternatively, the proxy could have inserted its FQDN in the Record-Route URI and the result would have been the same. This is because the proxy has both IPv4 and IPv6 addresses in the DNS; thus, the URI resolution would have yielded an IPv4 address to the UAC and an IPv6 address to the UAS.

3.2. User Agent Behavior

User agent clients MUST follow the normative text specified in Section 4.2 to gather IP addresses pertinent to the network. Having done that, clients MUST follow the recommendations in Section 5 to determine the order of the downstream servers to contact when routing a request.

Autonomous domains SHOULD deploy dual-stack user agent servers, or alternatively, deploy both IPv4-only and IPv6-only servers. In either case, the RR in DNS for reaching the server should be specified appropriately.

4. The Media Layer

SIP establishes media sessions using the offer/answer model [4]. One endpoint, the offerer, sends a session description (the offer) to the other endpoint, the answerer. The offer contains all the media parameters needed to exchange media with the offerer: codecs, transport addresses, protocols to transfer media, etc.

When the answerer receives an offer, it elaborates an answer and sends it back to the offerer. The answer contains the media parameters that the answerer is willing to use for that particular session. Offer and answer are written using a session description protocol. The most widespread protocol to describe sessions at present is called, aptly enough, the Session Description Protocol (SDP) [2].

A direct offer/answer exchange between an IPv4-only user agent and an IPv6-only user agent does not result in the establishment of a session. The IPv6-only user agent wishes to receive media on one or more IPv6 addresses, but the IPv4-only user agent cannot send media to these addresses, and generally does not even understand their format. Consequently, user agents need a means to obtain both IPv4 and IPv6 addresses to receive media and to place them in offers and answers.

> This IP version incompatibility problem would not exist if hosts implementing IPv6 also implemented IPv4, and were configured with both IPv4 and IPv6 addresses. In such a case, a UA would be able

to pick a compatible media transport address type to enable the hosts to communicate with each other.

Pragmatism dictates that IPv6 user agents undertake the greater burden in the transition period. Since IPv6 user agents are not widely deployed yet, it seems appropriate that IPv6 user agents obtain IPv4 addresses instead of mandating an upgrade on the installed IPv4 base. Furthermore, IPv6 user agents are expected to be dual-stacked and thus also support IPv4, unlike the larger IPv4-only user agent base that does not or cannot support IPv6.

An IPv6 node SHOULD also be able to send and receive media using IPv4 addresses, but if it cannot, it SHOULD support Session Traversal Utilities for NAT (STUN) relay usage [8]. Such a relay allows the IPv6 node to indirectly send and receive media using IPv4.

The advantage of this strategy is that the installed base of IPv4 user agents continues to function unchanged, but it requires an operator that introduces IPv6 to provide additional servers for allowing IPv6 user agents to obtain IPv4 addresses. This strategy may come at an additional cost to SIP operators deploying IPv6. However, since IPv4-only SIP operators are also likely to deploy STUN relays for NAT (Network Address Translator) traversal, the additional effort to deploy IPv6 in an IPv4 SIP network should be limited in this aspect.

However, there will be deployments where an IPv4/IPv6 node is unable to use both interfaces natively at the same time, and instead, runs as an IPv6-only node. Examples of such deployments include:

1. Networks where public IPv4 addresses are scarce and it is preferable to make large deployments only on IPv6.

2. Networks utilizing Layer-2's that do not support concurrent IPv4 and IPv6 usage on the same link.

4.1. Updates to RFC 3264

This section provides a normative update to RFC 3264 [4] in the following manner:

1. In some cases, especially those dealing with third party call control (see Section 4.2 of [12]), there arises a need to specify the IPv6 equivalent of the IPv4 unspecified address (0.0.0.0) in the SDP offer. For this, IPv6 implementations MUST use a domain name within the .invalid DNS top-level domain instead of using the IPv6 unspecified address (i.e., ::).

2. Each media description in the SDP answer MUST use the same network type as the corresponding media description in the offer. Thus, if the applicable "c=" line for a media description in the offer contained a network type with the value "IP4", the applicable "c=" line for the corresponding media description in the answer MUST contain "IP4" as the network type. Similarly, if the applicable "c=" line for a media description in the offer contained a network type with the value "IP6", the applicable "c=" line for the corresponding media description in the answer MUST contain "IP6" as the network type.

4.2. Initial Offer

We now describe how user agents can gather addresses by following the Interactive Connectivity Establishment (ICE) [7] procedures. ICE is protocol that allows two communicating user agents to arrive at a pair of mutually reachable transport addresses for media communications in the presence of NATs. It uses the STUN [18] protocol, applying its binding discovery and relay usages.

When following the ICE procedures, in addition to local addresses, user agents may need to obtain addresses from relays; for example, an IPv6 user agent would obtain an IPv4 address from a relay. The relay would forward the traffic received on this IPv4 address to the user agent using IPv6. Such user agents MAY use any mechanism to obtain addresses in relays, but, following the recommendations in ICE, it is RECOMMENDED that user agents support STUN relay usage [6] [8] for this purpose.

IPv4/IPv6 user agents SHOULD gather both IPv4 and IPv6 addresses using the ICE procedures to generate all their offers. This way, both IPv4-only and IPv6-only answerers will be able to generate a mutually acceptable answer that establishes a session (having used ICE to gather both IPv4 and IPv6 addresses in the offer reduces the session establishment time because all answerers will find the offer valid.)

> Implementations are encouraged to use ICE; however, the normative strength of the text above is left at a SHOULD since in some managed networks (such as a closed enterprise network) it is possible for the administrator to have control over the IP version utilized in all nodes and thus deploy an IPv6-only network, for example. The use of ICE can be avoided for signaling messages that stay within such managed networks.

4.3. Connectivity Checks

Once the answerer has generated an answer following the ICE procedures, both user agents perform the connectivity checks as specified by ICE. These checks help prevent some types of flooding attacks and allow user agents to discover new addresses that can be useful in the presence of NATs.

5. Contacting Servers: Interaction of RFC 3263 and RFC 3484

RFC 3263 maps a SIP or SIPS URI to a set of DNS SRV records for the various servers that can handle the URI. The Expected Output, given an Application Unique String (the URI) is one or more SRV records, sorted by the "priority" field, and further ordered by the "weight" field in each priority class.

> The terms "Expected Output" and "Application Unique String", as they are to be interpreted in the context of SIP, are defined in Section 8 of RFC 3263 [5].

To find a particular IP address to send the request to, the client will eventually perform an A or AAAA DNS lookup on a target. As specified in RFC 3263, this target will have been obtained through NAPTR and SRV lookups, or if NAPTR and SRV lookup did not return any records, the target will simply be the domain name of the Application Unique String. In order to translate the target to the corresponding set of IP addresses, IPv6-only or dual-stack clients MUST use the newer getaddrinfo() name lookup function, instead of gethostbyname() [16]. The new function implements the Source and Destination Address Selection algorithms specified in RFC 3484 [9], which is expected to be supported by all IPv6 hosts.

The advantage of the additional complexity is that this technique will output an ordered list of IPv6/IPv4 destination addresses based on the relative merits of the corresponding source/destination pairs. This will guarantee optimal routing. However, the Source and Destination Selection algorithms of RFC3484 are dependent on broad operating system support and uniform implementation of the application programming interfaces that implement this behavior.

> Developers should carefully consider the issues described by Roy et al. [19] with respect to address resolution delays and address selection rules. For example, implementations of getaddrinfo() may return address lists containing IPv6 global addresses at the top of the list and IPv4 addresses at the bottom, even when the host is only configured with an IPv6 local scope (e.g., link-local) and an IPv4 address. This will, of course, introduce a delay in completing the connection.

6. Security Considerations

This document describes how IPv4 SIP user agents can communicate with IPv6 user agents (and vice versa). To do this, it uses additional protocols (STUN relay usage [6], ICE [7], SDP [2]); the threat model of each such protocol is included in its respective document. The procedures introduced in this document do not introduce the possibility of any new security threats; however, they may make hosts more amenable to existing threats. Consider, for instance, a UAC that allocates an IPv4 and an IPv6 address locally and inserts these into the SDP. Malicious user agents that may intercept the request can mount a denial-of-service attack targeted to the different network interfaces of the UAC. In such a case, the UAC should use mechanisms that protect confidentiality and integrity of the messages, such as using the SIPS URI scheme as described in Section 26.2.2 of RFC3261 [3], or secure MIME as described in Section 23 of RFC3261 [3]. If HTTP Digest is used as an authentication mechanism in SIP, then the UAC should ensure that the quality of protection also includes the SDP payload.

7. Acknowledgments

The authors would like to thank Mohamed Boucadair, Christine Fischer, Cullen Jennings, Aki Niemi, Jonathan Rosenberg, and Robert Sparks for discussions on the working group list that improved the quality of this document.

Additionally, Francois Audet, Mary Barnes, Keith Drage, and Dale Worley provided invaluable comments as part of the working group Last Call review process. Jari Arkko, Lars Eggert, Tobias Gondrom, Suresh Krishnan, and Tim Polk conducted an in-depth review of the work as part of the IESG and Gen-ART reviews.

8. References

8.1. Normative References

[1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[2] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.

[3] 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.

[4] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with the Session Description Protocol (SDP)", RFC 3264, June 2002.

[5] Rosenberg, J. and H. Schulzrinne, "Session Initiation Protocol (SIP): Locating SIP Servers", RFC 3263, June 2002.

[6] Mahy, R., Matthews, P., and J. Rosenberg, "Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN)", RFC 5766, April 2010.

[7] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", RFC 5245, April 2010.

[8] Camarillo, G., Novo, O., and S. Perreault, "Traversal Using Relays around NAT (TURN) Extension for IPv6", RFC 6156, April 2011.

[9] Draves, R., "Default Address Selection for Internet Protocol version 6 (IPv6)", RFC 3484, February 2003.

8.2. Informative References

[10] Schulzrinne, H. and B. Volz, "Dynamic Host Configuration Protocol (DHCPv6) Options for Session Initiation Protocol (SIP) Servers", RFC 3319, July 2003.

[11] Schulzrinne, H., "Dynamic Host Configuration Protocol (DHCP-for-IPv4) Option for Session Initiation Protocol (SIP) Servers", RFC 3361, August 2002.

[12] Rosenberg, J., Peterson, J., Schulzrinne, H., and G. Camarillo, "Best Current Practices for Third Party Call Control (3pcc) in the Session Initiation Protocol (SIP)", BCP 85, RFC 3725, April 2004.

[13] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.

[14] Handley, M., Schulzrinne, H., Schooler, E., and J. Rosenberg, "SIP: Session Initiation Protocol", RFC 2543, March 1999.

[15] Gurbani, V., Boulton, C., and R. Sparks, "Session Initiation Protocol (SIP) Torture Test Messages for Internet Protocol Version 6 (IPv6)", RFC 5118, February 2008.

[16] Shin, M-K., Hong, Y-G., Hagino, J., Savola, P., and E. Castro, "Application Aspects of IPv6 Transition", RFC 4038, March 2005.

[17] Nordmark, E. and R. Gilligan, "Basic Transition Mechanisms for IPv6 Hosts and Routers", RFC 4213, October 2005.

[18] Rosenberg, J., Mahy, R., Matthews, P., and D. Wing, "Session Traversal Utilities for NAT (STUN)", RFC 5389, October 2008.

[19] Roy, S., Durand, A., and J. Paugh, "IPv6 Neighbor Discovery On-Link Assumption Considered Harmful", RFC 4943, September 2007.

Appendix A. Sample IPv4/IPv6 DNS File

A portion of a sample DNS zone file entry is reproduced below that has both IPv4 and IPv6 addresses. This entry corresponds to a proxy server for the domain "example.com". The proxy server supports the Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) transport for both IPv4 and IPv6 networks.

```
...
_sip._tcp  SRV  20 0 5060 sip1.example.com
           SRV   0 0 5060 sip2.example.com
_sip._udp  SRV  20 0 5060 sip1.example.com
           SRV   0 0 5060 sip2.example.com

sip1 IN A     192.0.2.1
sip1 IN AAAA  2001:db8::1
sip2 IN A     192.0.2.2
sip2 IN AAAA  2001:db8::2
...
```

Authors' Addresses

Gonzalo Camarillo
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

EMail: Gonzalo.Camarillo@ericsson.com

Karim El Malki
Athonet
AREA Science Park
Padriciano 99
Trieste (TS) 34149
Italy

EMail: karim@athonet.com

Vijay K. Gurbani
Bell Laboratories, Alcatel-Lucent
1960 Lucent Lane
Rm 9C-533
Naperville, IL 60563
USA

Phone: +1 630 224 0216
EMail: vkg@bell-labs.com