SIPCORE
Internet-Draft
Intended status: Standards Track
Expires: June 15, 2017

O. Johansson
Edvina AB
G. Salgueiro
Cisco Systems
D. Worley
Ariadne

December 12, 2016

Happy EarBalls: Success with Dual-Stack, Connection-Oriented SIP
draft-worley-sip-he-connection-01

Abstract

The Session Initiation Protocol (SIP) supports multiple transports running both over IPv4 and IPv6 protocols. In more and more cases, a SIP user agent (UA) is connected to multiple network interfaces. In these cases setting up a connection from a dual stack client to a dual stack server may suffer from the issues described in RFC 6555 [RFC6555] ("Happy Eyeballs") - significant delays in the process of setting up a working flow to a server. This negatively affects user experience.

This document builds on RFC 6555 and explains how a [RFC3261] compliant SIP implementation can minimize delays when contacting a host name (obtained by using DNS NAPTR and SRV lookups) in a dual stack network using connection-oriented transport protocols.

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 June 15, 2017.

Table of Contents

1. Introduction

The Session Initiation Protocol (SIP) [RFC3261] and the documents that extended it provide support for both IPv4 and IPv6. However, this support has problems with environments that are characteristic of the transitional migratory phase from IPv4 to IPv6 networks. During this phase, many server and client implementations run on dual-stack hosts. In such environments, a dual-stack host will likely suffer greater connection delay, and by extension an inferior user experience, than an IPv4-only host. The need to remedy this diminished performance of dual-stack hosts led to the development of the "Happy Eyeballs" [RFC6555] algorithm, which has since been implemented in many protocols and applications.

This document revises the the [RFC3263] procedures to apply the "Happy Eyeballs" framework. A dual-stack client using connection-oriented transport should set up multiple connections in parallel, to targets based on the result of DNS queries. This document starts at the point where a SIP implementation has a host name that resolves using A and AAAA records. Such a host name can either be the host part of a SIP URI (possibly including a port number) or the result of a lookup using DNS NAPTR and SRV records as described in RFC 3263 (as updated by RFC 7984[RFC7984]).

Procedures for connectionless transport protocols for SIP are outside the scope of this document. Procedures allowing a client to change the order of contacting targets that were derived from different host names are outside the scope of this document.

The concepts in this document are elaborated from those developed in [RFC6555], and so some background information in RFC 6555 is not repeated here. The reader is encouraged to read the available documentation regarding implementations of RFC 6555, as well as study Open Source implementations, in order to learn from the experience accumulated since the publishing of RFC 6555 in 2012.

2. Terminology and Conventions Used in This Document

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].

RFC 3261 [RFC3261] defines additional terms used in this document that are specific to the SIP domain such as "proxy"; "registrar"; "redirect server"; "user agent server" or "UAS"; "user agent client" or "UAC"; "back-to-back user agent" or "B2BUA"; "dialog"; "transaction"; "server transaction".

This document uses the term "SIP Server" that is defined to include the following SIP entities: user agent server, registrar, redirect server, a SIP proxy in the role of user agent server, and a B2BUA in the role of a user agent server.

This document also uses the following terminology to make clear distinction between SIP entities supporting only IPv4, only IPv6 or supporting both IPv4 and IPv6.

IPv4-only UA/UAC/UAS: An IPv4-only UA/UAC/UAS supports SIP signaling and media only on the IPv4 network. It does not understand IPv6 addresses.

IPv6-only UA/UAC/UAS: An IPv6-only UA/UAC/UAS supports SIP signaling and media only on the IPv6 network. It does not understand IPv4 addresses.

IPv4/IPv6 UA/UAC/UAS: A UA/UAC/UAS that supports SIP signaling and media on both IPv4 and IPv6 networks; such a UA/UAC/UAS is known (and will be referred to in this document) as a "dual-stack" [RFC4213] UA/UAC/UAS.

Discussion: Do we need special handling of websocket transport?

While this document uses the term "dual-stack" based on RFC 6555 and earlier terminology, the authors acknowledge that the same solution can be applied to multi-interface environments as well as future versions of IP alongside with the current ones.

3. DNS Procedures in a Dual-Stack Network

A SIP client uses DNS to find a server based on a SIP URI. This process is described in [RFC3263] and updated in [RFC7984]. Using this process, a list of "targets" is constructed, where each target consists of an IP address, a port number, and a protocol (e.g., TCP, UDP, TLS) by which to contact that address/port. The process proceeds by constructing a sequence of host names, possibly by looking up NAPTR and/or SRV DNS records, and then for each host name looking up DNS address records (for all address families supported by the client) to generate the list of IP addresses for targets that are derived from that host name. The addresses for each host name are ordered using the client's destination selection rules[RFC6724]. The sorted targets for all the host names are then concatenated into the sequence of targets to which the client will attempt to send the SIP message.

Previously, the client contacts the targets in order until one is contacted successfully. In order to contact a target, the client

establishes a transport connection (if necessary), sends the message using the transport (possibly resending the message several times), and then (for requests) waits for a response (either provisional or final). The process ends successfully if the client receives a response. The process ends unsuccessfully if the client receives a permanent error from the transport layer or if a SIP timer (Timer B or Timer F in [RFC3261]) expires. Timeouts generally default to 32 seconds.

If the user has to wait for even one timeout, this will seriously degrade the user experience. Thus, it is desirable to minimize the number of times the client has timeouts when sending requests.

If the target list contains both IPv6 addresses and IPv4 addresses, this procedure can degrade the user's experience in common situations. Typically, this problem arises when the client has an IPv6 interface, the server's preferred address is an IPv6 address, but the transit networks between the client and server do not carry IPv6. This can cause the client to attempt to send a SIP request for 32 seconds before it times out that target and continues with an IPv4 target. This problem parallels a problem that was widely seen in web browsers that was cured by specifying that web browsers should use a "Happy Eyeballs" algorithm[RFC6555] to determine the order in which to contact target addresses.

This document specifies an amendment to these procedures, by which the subsequences of targets derived from individual host names may be contacted in a different order than is specified by the destination selection rules. As in [RFC6555], the algorithm that the client uses is not specified by this document, but this document places requirements on the algorithm that improve the user's experience without unduly burdening the Internet infrastructure. By analogy with the name "Happy Eyeballs" for similar algorithms in web browsers, we label these algorithms "Happy EarBalls"[UD].

This document modifies the transport procedures only in the case when all targets for a host name have connection-oriented protocols (currently, TCP, TLS, and SCTP). Other cases are outside the scope of this document. The case of SIP using WebSocket transport is outside the scope of this document because there is only one transport target, the WebSocket transport provided by the context.

4. Establishing a Connection

This section discusses the situation that most closely resembles RFC 6555, which is when the SIP client has no active connection to any of the targets in a subsequence of targets derived from one host name. This specification allows the client to attempt to send a request to

targets in the subsequence in a different order than is prescribed by RFC 3262 and RFC 6724. In addition, this specification allows the client to attempt to initiate a connection to a target without subsequently sending a request to the target. However, the algorithm which the client uses use meet the constraints in this section.

Typically, the SIP client will set up two connections, with some head start for one address family (which is possibly be configurable) and then select the first completed connection for use and close the other one. The SIP message is sent on the selected connection only.

The reason for this approach is to avoid the timeout associated with sending an unsuccessful SIP request, requiring the client to wait for a timeout before the request can be sent on a connection to another target - which in the case of SIP with default timers is 32 seconds. Waiting for timeout before trying with a secondary address will lead to a very poor user experience.

4.1. Requirements

The following requirements apply to any implementation that takes advantage of the relaxed requirements on message transmission specified by this document.

4.1.1. Address Preferences

An implementation MUST prefer the first IP address family returned by the host's address preference policy, unless it implements a stateful algorithm as described in Section 4.1.2. This usually means giving preference to IPv6 over IPv4, although that preference can be overridden by user configuration or by network configuration. If the host's policy is unknown or not attainable, the implementation MUST prefer IPv6 over IPv4.

4.1.2. Stateful Behavior

The algorithm may be stateful -- that is, the algorithm will remember that IPv6 always fails, or that IPv6 to certain prefixes always fails, and so on. This section constrains such algorithms. Stateless algorithms, which do not remember the success/failure of previous connections, are not discussed in this section.

After making a connection attempt using the preferred address family (e.g., IPv6) and failing to establish a connection within a certain time period (see Section 6.6), a Happy EarBalls implementation will decide to initiate a second connection attempt using the same address family or the other address family.

Such an implementation MAY make prioritize making subsequent connection attempts (to the same host or to other hosts) using the successful address family (e.g., IPv4). So long as new connections are being attempted by the host, such an implementation MUST occasionally make connection attempts using the host's preferred address family, as that family may have become functional again, and the client SHOULD do so every 10 minutes. The 10-minute delay before retrying a failed address family avoids the simple doubling of connection attempts on both IPv6 and IPv4. This can be achieved by flushing Happy EarBalls state every 10 minutes, which does not significantly harm the application's subsequent connection setup time. If connections using the preferred address family are again successful, the preferred address family MUST be used for subsequent connections. A stateful implementation MAY track connection success and failure based on IPv6 or IPv4 prefix. E.g., connections to addresses with the same prefix as the interface's address may be successful whereas connections to addresses with different prefixes fail.

4.1.3. Reset on Network (Re-)Initialization

Because every network has different characteristics (e.g., working or broken IPv6 or IPv4 connectivity), a Happy EarBalls algorithm SHOULD re-initialize when the interface is connected to a new network. Interfaces can determine network (re-)initialization by a variety of mechanisms (e.g., Detecting Network Attachment in IPv4 (DNAv4) [RFC4436], DNAv6 [RFC6059]).

4.1.4. Abandon Non-Winning Connections

Non-winning connections that are not assigned as flows for the purposes of [RFC5626] SHOULD be abandoned, even though they could -- in some cases -- be put to reasonable use.

Justification: This reduces the load on the server (file descriptors, TCP control blocks) and stateful middleboxes (NAT and firewalls). Also, if the abandoned connection is IPv4, this reduces IPv4 address sharing contention.

(There are some unlikely situations where a non-winning connection could be useful in the future: If at a later time, the client must send a request to a different host name, but one which has as a target the peer of the non-winning connection and does not have as a target the peer of the winning connection.)

5. Using an Existing Connection

When a client desires to send a message to a target that is within a subsequence of targets derived from one host name, the client may already have a connection established to one of the targets through either SIP Outbound[RFC5626] or the procedures of Section 4. The client SHOULD attempt to send the message using the existing connection in preference to using a new connection to one of the targets.

If, in the client's operational environment, there is a significant risk that the connection has become unusable without the client becoming aware of it, the client SHOULD consider testing whether the connection is usable before sending the message using the connection. Some possible ways to probe a connection to determine if it is still usable are:

- Send a keep-alive, as specified by the protocol of the connection.
- Send a CR-LF-CR-LF keep-alive on a SIP Outbound connection[RFC5626].
- Send an OPTIONS request with "Max-Forwards: 0".

(Note that a probe using an OPTIONS request can be used with any protocol. If the OPTIONS reaches the target, the target is required to respond with either a 200 or 483 response[RFC3261] without forwarding it to another entity. Conveniently, a server can respond to such a request statelessly, so such requests are low-overhead. (Although the [RFC5626] keep-alive methods have even lower overhead.))

6. Additional Considerations

This section discusses additional considerations related to Happy EarBalls.

6.1. Preemtive Actions

A client may be in a situation where it has advance notice that it is likely to need to send a message to a particular host name, for instance, if the user of a UA begins dialing an outgoing call which will be routed through a particular outgoing proxy. In such a situation, the client SHOULD consider preemptively establishing a connection (Section 4) or probing an existing connection (Section 5).

6.2. Determining the Type of an Address

For some transitional technologies, such as a dual-stack host, it is easy for the application to recognize a native IPv6 address (learned via a AAAA query) and a native IPv4 address (learned via an A query). The use of IPv6/IPv4 translation in the local network makes it difficult or impossible to determine the address family by which the connection will traverse the global network. However, IPv6/IPv4 translators do not need to be deployed on networks with dual-stack clients because dual-stack clients can use their native IP address family. Environments where IPv6/IPv4 translation is active will degrade the ability of Happy EarBalls algorithms to establish working connections.

6.3. Debugging and Troubleshooting

Happy EarBalls is aimed at ensuring a reliable user experience regardless of connectivity problems affecting any single transport. However, this naturally means that applications employing these techniques are by default less useful for diagnosing issues with a particular address family. To assist in that regard, an implementation MAY provide a mechanism to disable their Happy EarBalls behavior via a user setting, and to provide data useful for debugging (e.g., a log or way to review current preferences).

6.4. Three or More Interfaces

A dual-stack host normally has one physical interface, and all network access is done via IPv4 and IPv6 addresses assigned to that interface. However, a dual-stack host might have additional physical interfaces or additional logical interfaces (e.g., because of a VPN). Additional Happy EarBalls considerations for optimal operation with additional physical or logical interfaces is for further study and is outside the scope of this document.

6.5. Multiple A and AAAA Resource Records

It is possible that a DNS query for an A or AAAA resource record will return more than one A or AAAA address. When this occurs, it is RECOMMENDED that a Happy EarBalls implementation order the responses following the host's address preference policy and then try the first target. If that fails after a certain time (see Section 6.6), the next target SHOULD be chosen from the other address family.

If the second attempt fails to connect, a Happy EarBalls implementation SHOULD try the other targets; the order of these connection attempts is not important.

Servers sometimes have multiple A records to provide load-balancing across their servers (although load-balancing is better obtained using SRV records). This same technique can be used for AAAA records, as well. However, if multiple AAAA records are returned to a client that is not using Happy EarBalls and that has broken IPv6 connectivity, the multiple AAAA records will further increase the delay to fall back to IPv4, as the client will attempt to connect to all of their addresses first. Thus, SIP server operators with native IPv6 connectivity SHOULD NOT offer multiple AAAA records. If Happy EarBalls is widely deployed in the future, this recommendation might be revisited.

6.6. Connection Timeout

The primary purpose of Happy EarBalls is to reduce the wait time for a dual-stack connection to complete, especially when the IPv6 path is broken and IPv6 is preferred. Using a short timeout between initiating an IPv6 connection and initiating an IPv4 connection (on the order of tens of milliseconds) achieves this goal, but at the cost of network traffic. This network traffic may be billable on certain networks, will create state on some middleboxes (e.g., firewalls, intrusion detection systems, NATs), and will consume ports if IPv4 addresses are shared. For these reasons, it is RECOMMENDED that connection attempts be paced to give connections a chance to complete. It is RECOMMENDED that connection attempts be paced 150-250 ms apart to balance human factors against network load. A stateful algorithm MAY be more aggressive (that is, make connection attempts closer together), if it maintains estimates of the expected connection completion times.

7. Security Considerations

This document places additional restrictions on the existing procedures in the SIP protocol. The specific security vulnerabilities, attacks and threat models of the various protocols discussed in this document (SIP, DNS, SRV records, etc.) are well-documented in their respective specifications.

8. IANA Considerations

This document does not require any actions by IANA.

9. History

Note to RFC Editor: Upon publication, remove this section.

9.1. Changes from draft-worley-sip-he-connection-00 to draft-worley-sip-he-connection-01

9.2. Changes from draft-johansson-sip-he-connection-01 to draft-worley-sip-he-connection-00

This version has a different name for technical reasons. It is, in reality, the successor to draft-johansson-sip-he-connection-01.

Move Acknowledgments after References, as that is the style the Editor prefers.

Updated Security Considerations: This increment of the H.E. work does not make normative changes in existing SIP.

Copy a lot of text from RFC 6555, as this I-D is parallel to RFC 6555.

Changed "hostname" to "host name", as the latter form is more common in RFCs by a moderate margin.

Revised some of the introduction text to parallel the introduction of RFC 7984.

Changed name of algorithm to "Happy EarBalls", added reference to Urban Dictionary.

Many expansions of the discussion and revisions of the wording.

10. References

10.1. Normative References

[RFC6555] Wing, D. and A. Yourtchenko, "Happy Eyeballs: Success with Dual-Stack Hosts", RFC 6555, DOI 10.17487/RFC6555, April 2012, <http://www.rfc-editor.org/info/rfc6555>.

10.2. Informative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, DOI 10.17487/RFC3261, June 2002, <http://www.rfc-editor.org/info/rfc3261>.

[RFC3263] Rosenberg, J. and H. Schulzrinne, "Session Initiation Protocol (SIP): Locating SIP Servers", RFC 3263, DOI 10.17487/RFC3263, June 2002, <http://www.rfc-editor.org/info/rfc3263>.

[RFC4213] Nordmark, E. and R. Gilligan, "Basic Transition Mechanisms for IPv6 Hosts and Routers", RFC 4213, DOI 10.17487/RFC4213, October 2005, <http://www.rfc-editor.org/info/rfc4213>.

[RFC5626] Jennings, C., Ed., Mahy, R., Ed., and F. Audet, Ed., "Managing Client-Initiated Connections in the Session Initiation Protocol (SIP)", RFC 5626, DOI 10.17487/RFC5626, October 2009, <http://www.rfc-editor.org/info/rfc5626>.

[RFC6724] Thaler, D., Ed., Draves, R., Matsumoto, A., and T. Chown, "Default Address Selection for Internet Protocol Version 6 (IPv6)", RFC 6724, DOI 10.17487/RFC6724, September 2012, <http://www.rfc-editor.org/info/rfc6724>.

[RFC7984] Johansson, O., Salgueiro, G., Gurbani, V., and D. Worley, Ed., "Locating Session Initiation Protocol (SIP) Servers in a Dual-Stack IP Network", RFC 7984, DOI 10.17487/RFC7984, September 2016, <http://www.rfc-editor.org/info/rfc7984>.

[UD] The Jews Who Stole Christmas, , "Urban Dictionary, entry 'Earballs'", December 2011, <http://www.urbandictionary.com/define.php?term=Earballs>.

Acknowledgements

The authors would like to acknowledge the support and contribution of the SIP Forum IPv6 Working Group. This document is based on a lot of tests and discussions at SIPit events, organized by the SIP Forum.

Most of the material in Section 4 and Section 6 is taken from [RFC6555], whose authors are Dan Wing and Andrew Yourtchenko.

Authors' Addresses

Olle E. Johansson
Edvina AB
Runbovaegen 10
Sollentuna SE-192 48
SE

Email: oej@edvina.net

Gonzalo Salgueiro
Cisco Systems
7200-12 Kit Creek Road
Research Triangle Park, NC 27709
US

Email: gsalguei@cisco.com

Dale R. Worley
Ariadne Internet Services
738 Main St.
Waltham, MA 02451
US

Email: worley@ariadne.com