A Media-based Traceroute Function for
the Session Initiation Protocol (SIP)
draft-kaplan-straw-sip-traceroute-01

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with
the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering
Task Force (IETF), its areas, and its working groups. Note that
other groups may also distribute working documents as Internet-
Drafts.

Internet-Drafts are draft documents valid for a maximum of six
months and may be updated, replaced, or obsoleted by other documents
at any time. It is inappropriate to use Internet-Drafts as
reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at
http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at

This Internet-Draft will expire on December 15, 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
1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119. The terminology in this document conforms to RFC 2828, "Internet Security Glossary".

B2BUA: a SIP Back-to-Back User Agent, which is the logical combination of a User Agent Server (UAS) and User Agent Client (UAC).

UAS: a SIP User Agent Server.

UAC: a SIP User Agent Client.
Traceroute: a mechanism to trace a path of hops from an originator to a destination. For IP, this is typically done using the TTL field of the IP header, starting at the value 1 and incrementing by 1 as each IP hop responds with an ICMP error. For SIP this can be done using Max-Forwards header field starting with the value 0, in a similar fashion to the TTL field.

It is assumed the reader is already familiar with media-loopback [RFC6849].

2. Introduction

In many deployments, the media for SIP-created sessions does not flow directly from the originating user’s UAC to the answering user’s UAS. Often, SIP B2BUAs in the SIP signaling path participate in the media plane, either for injecting media such as ringtones or music-on-hold, or for relaying media in order to provide functions such as transcoding, IPv4-IPv6 conversion, NAT traversal, SRTP termination, media steering, etc.

As more and more SIP domains get deployed and interconnect, the odds of a SIP session crossing such media-plane B2BUAs increases, as well as the number of such B2BUAs any given SIP session may go through. In other words, any given SIP session may cross any number of B2BUA’s both in the SIP signaling plane as well as media plane.

If failures or degradation occurs in the media plane, it is difficult to determine where in the media path they occur. In order to aid managing and troubleshooting SIP-based sessions and media crossing such B2BUAs, it would be useful to be able to test the media path to each B2BUA separately from the source. A mechanism to perform media-loopback test sessions has been defined in [RFC6849], but it would be difficult to use the mechanism directly to test B2BUAs because typically the B2BUAs do not have an Address of Record (AoR) to be targeted, nor is it known a priori which B2BUAs will be crossed for any given session.

For example, suppose calls from Alice to Bob have media problems. Alice would like to test the media path to each B2BUA in the path to Bob separately, to determine which segment has the issues. Alice cannot target the B2BUAs directly for each test call, because she doesn’t know what URIs to use to target them; nor would using such URIs guarantee the same media path be used as a call to Bob. A better solution would be to make a test call targeted to Bob, but with a SIP traceroute-type mechanism that makes the call terminate at the B2BUAs, such that she can perform test sessions to test the media path to each downstream B2BUA.
This document defines how such a mechanism can be employed, using the [RFC6849] mechanism along with the Max-Forwards SIP header field such that a SIP User Agent can make multiple test calls, each reaching a B2BUA further downstream. Each B2BUA in the path that supports this mechanism would answer the media-loopback call, and thus the originating SIP UA can test the media path up to that B2BUA.

3. The SIP Traceroute Mechanism

The Max-Forwards header field can already be used to perform a simple SIP-request traceroute mechanism by generating a SIP request initially using a Max-Forwards value of 0, receiving a 483 Too Many Hops response from the next-hop, and then incrementing the value for subsequent SIP requests, thereby reaching SIP devices further and further downstream and receiving 483 from each of them.

The mechanism described in this document uses such a Max-Forwards style traceroute to perform media-loopback testing. To perform a SIP media-plane traceroute, the originating UAC generates a SIP INVITE to a target AoR, with SDP based on [RFC6849]. When the request reaches the first B2BUA that supports this mechanism, if the B2BUA allows such testing from the requesting UAC, then the B2BUA will answer the INVITE to establish the dialog and create a media-loopback session.

The originating UAC can then generate another INVITE to the same target AoR with a B2bua-Hops header value of 1, which will reach the second B2BUA that supports this mechanism, and so on.

Using this mechanism a SIP UAC can test the path from itself to each successive B2BUA on the path to a target. Such a mechanism could also be useful for establishing a permanent test call between an Enterprise and a Service Provider across a SIP Trunk, for example, or for automated measurement systems to test the media path between domains, etc.

3.1. Processing a Received Max-Forwards Header Field

As currently defined in [RFC3261], the UAS half of a B2BUA does not technically need to inspect the Max-Forwards header field value for received requests - only Proxies do. This behavior was updated by [draft-loop-detection], such that a compliant B2BUA needs to both inspect the value in order to prevent loops, as well as copy and decrement the value as if it were a Proxy. This document also requires such behavior in order to succeed, therefore a B2BUA supporting the traceroute mechanism defined in this document MUST also comply with [draft-loop-detection].
3.2. Answering the INVITE

If a SIP B2BUA or UAS receives a dialog-creating INVITE request with a Max-Forwards header value of 0, with SDP for media-loopback based on [RFC6849], and the policies of the B2BUA/UAS allow it to answer such a request, then it is answered as if the original target of the request were the local SIP B2BUA/UAS. The normal procedures of SIP apply, as well as [RFC6849], as if the request had been targeted at the local B2BUA device all along.

[Open Issue: how does the UAC know the request reached a B2BUA vs. the final UAS? (e.g., how does it know when to stop testing?)]

4. Open Issues

- How does the UAC know when the request finally reached the ultimate UAS; or conversely that it has not yet reached the ultimate UAS? (e.g., insert a param somewhere?)

5. Security Considerations

There are security implications for the mechanism defined in this document. Answering media-loopback calls in a B2BUA consumes resources on the B2BUA, and network bandwidth in between; therefore, B2BUAs should have some means of controlling who can make such test calls, how many concurrent calls can be established and maintained, and for how long. Such policies are typically vendor-specific based on local policies, and do not need to be defined in this document.

6. IANA Considerations

This document makes no request of IANA yet - if a new parameter or field needs to be inserted when answering the INVITE, then it will be registered in IANA.

7. Acknowledgments

The general concept of performing media-loopback on a hop-by-hop basis using a decrementing header traceroute style approach came out of discussions several years ago, between the author, Kaynam Hedayat, Nagarjuna Venna, Patrick McLampy, and others. Other people that have contributed to the topic over the years since then: Zaid Ally, Dianna Stiller, Jon Boone, and several others whom I have lost the names of since.

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).
8. References

8.1. Normative References


Author’s Address

Hadriel Kaplan
Oracle
Email: hadriel.kaplan@oracle.com