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/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html

This Internet-Draft will expire on May 4, 2009.

Copyright and License Notice

Copyright (c) 2009 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.

Abstract

This document identifies several commonly found interoperability issues with SIP, and provides guidance to implementers for how to
1. Introduction

SIP has grown both in terms of vendor/customer adoption and protocol complexity, with numerous implementations, and differing assumptions, leading to numerous interoperability issues. Unlike some other protocols, it suffers from a lack of either a single dominant vendor, or of a single autocratic standards body. The large number of vendors involved, from different regions of the World, and the differences in needs and wants of the customers of those vendors, has led to a complicated interoperability problem space.

This document lists some of the more common interoperability issues encountered in deployments, and provides BCP recommendations for how to avoid or resolve them.

2. Applicability

This draft is focused on SIP interoperability issues only.
3. General Interoperability Issues

3.1. Response code issues

There are numerous reasons a given response code may be sent, and in some cases more than one response code may be appropriate, which has led to differing expectations and behaviors. The need to resolve such conflicts between domains of proxies has led to middle-boxes changing the response codes, which may well exacerbate the problem in the future.

In general the interoperability problems that arise are where the upstream proxies or UAC perform automatic re-attempts to alternate paths for certain response codes but not others, and such action cannot be known in advance to the downstream device. For example a 404 Not Found or 480 Temporarily Unavailable are commonly returned by a proxy when it cannot find a route to the target for any number of reasons, and this response causes some upstream nodes to try alternate paths and some not to. Because a 404/480 can be returned for a variety of reasons, some of which should cause a re-route and some not, some vendors send different response codes than 404/480 for those conditions: response codes which are more explicit about whether a re-route is the appropriate action.

Another example is 503, which seems to cover everything from temporary overload conditions, administrative-down state, permanent failure, and as a catch-all for anything not easily identified by other codes. Some devices treat this response code as a semi-permanent condition for the next-hop, and avoid sending any subsequent requests to the next-hop for a sustained period of time, which may or may not be the correct action to take. Unfortunately the upstream nodes have no idea which downstream proxy actually generated the 503.

[See "REGISTER response behavior" section for related problems]

3.2. SIP field and message lengths

While the RFCs do not define any maximum lengths for SIP header fields (values, parameters, etc.) or SIP messages, the reality of computing technology is such that vendors often do impose maximum lengths for received fields and messages. Whether it’s due to security concerns, product architecture, logging constraints, etc., the fact is there are many systems which cannot or will not handle fields as large as other systems can generate. Although [RFC3261] does define some specific response codes (413/414/513) for this case, it does not fix the underlying interoperability issue.
Devices cannot simply stop sending larger fields based on a SIP response code.

This issue has been appearing more frequently lately, with the use of embedded cookies in URIs and parameter growth.

3.2.1 Recommended Behavior

It is RECOMMENDED that a UA, B2BUA or Proxy not generate any SIP field longer than 256 bytes. This includes any header parameters, URI, and header value. It is RECOMMENDED that a Proxy, B2BUA, and UA be able to process SIP fields of at least 1024 bytes in length.

It is RECOMMENDED that a UA or B2BUA not generate any SIP message size larger than 64,000 bytes. Note this is not 65,535, in order to accommodate message growth due to header insertion in Proxies. It is RECOMMENDED that a Proxy, B2BUA, or UA be able to process SIP messages of at least 65,535 bytes.

It is RECOMMENDED that a UA or B2BUA not generate any SIP message bodies larger than 32,768 bytes. It is RECOMMENDED that a B2BUA or UA be able to process SIP message bodies of at least 64,000 bytes in length.

3.3. SIP and TEL URI formats

Despite all RFC wording to the contrary, the SIP URI format has seen widespread use as essentially the semantic equivalent of the TEL URI, albeit with different syntax. Many provider systems treat sip:+16035551212@example.net as logically equivalent to tel:+16035551212, even though the former has local scope to example.net only, and the latter has global scope. Part of the reason for this, I believe, is that originating UA’s have no real way of knowing when a URI should be one or the other – the user pressed digit buttons and hit "send", and all the UAC can do is send the request to sip:[digits]@[local-domain]. It doesn’t know the numbers pressed were global in scope, or even E.164 numbers. Only the routing proxies know this, and even then they only know the numbers they’re each responsible for. Thus we see the domain portion of SIP URIs getting replaced by middle-boxes at provider boundaries, if the username portion looks like an E.164 number.

Furthermore, many systems have either been designed or provisioned to handle only one scheme type (i.e., SIP URIs). This has led to cases where requests are rejected unless the appropriate URI scheme is used, and frequently that single common scheme needs to be used in more than just the request-URI (e.g., To and From URI’s as well). This wholesale replacement of schemes and domain names in URIs leads
to interop issues when the same URIs are expected to be used for end-to-end purposes, in headers or XML bodies the middle-boxes do not or cannot change. The most recent example is [RFC4474] sip-identity.

3.3.1 Recommended Behavior

It is RECOMMENDED that a UA or B2BUA only use SIP URIs for any SIP header field values which need a URI; for example the request URI, To, From, Contact, P-Asserted-Identity, P-Called-Party-ID, P-Associated-URI, History-Info, and so on. It is RECOMMENDED that a UA, B2BUA, or Proxy be able to process/support Tel and SIPS URI’s per their associated RFC’s.

4. Specific Interoperability Issues

4.1. Offer-less Invites and Re-Invites

Although this is clearly a device implementation issue (i.e., a "bug"), we have seen numerous devices from different vendors have trouble handling Invites or re-Invites that do not contain SDP. For initial Invites without SDP, often the root cause for failure is that specific request routing or admission decision logic of intermediate devices depends on the SDP; for example devices which route calls based on codec, or bandwidth allocation devices, or 3PCC transcoding devices which themselves send out offer-less Invites but didn’t expect to receive such. (Apparently they never considered that a call could cross two such systems!)

For re-Invites, the delayed SDP offer model is performed for very specific use cases which are common, but were simply not envisioned by the developers of the UA’s.

4.1.1 Recommended Behavior

It is RECOMMENDED that a UAC provide the SDP offer with its INVITE requests. A UAS or B2BUA MUST support receiving an INVITE without an SDP offer.

4.2. REGISTER response behavior

Another form of the interop problems that arise from responses is the behavior of UA’s with regard to Registration and Subscribe response handling. For example, only a minority of UA’s properly support 3xx redirects for REGISTER, even though it would be a useful mechanism for load-balancing. For REGISTER requests specifically, it would be beneficial if there was explicit documentation of what actions should be performed by the UAC.
To reinforce this point, consider that UA’s perform Registrations and Subscriptions in a fairly automatic fashion with little user interaction, and so the way in which they treat specific response codes can have dramatic consequences. For example, it is not well-defined what a UA should do when its REGISTER is rejected with a 404, or even 503, and hardly any UA’s honor the Retry-After header. A very few UA’s will give up altogether and wait for user input; some UA’s will wait a few minutes and try again, indefinitely; some will re-attempt their Registration almost immediately, even faster, and never give up. This creates numerous problems in large network deployments, and has led SBC vendors to implement various protection schemes — from dynamic hardware ACLs, to even sending a 200 ok just to shut the UA up.

4.3. Call-hold signaling

The legacy mechanism defined in [RFC2543] for call-hold by setting the SDP connection address to 0.0.0.0 is unfortunately far from obsolete in usage, despite the superior direction attribute concept of [RFC3264]. To increase interoperability, some devices send both types in the re-Invite, which defeats the purpose of using a direction attribute (e.g., keeping RTCP flowing). Other vendors send the direction attribute first, and if the SDP answer does not mirror it they use the legacy approach, which leads to extraneous signaling overhead. An IETF recommendation/BCP for this is probably warranted. In hindsight [RFC3264] should have been backwards compatible (e.g., still using the 0.0.0.0 syntax with some new attribute for on-hold connection address, which would be ignored by legacy devices but used by newer ones). [note: I recognize this is SDP not SIP, but it’s a big deal and was caused by rfc2543]

5. References


Author’s Address

Hadriel Kaplan
Acme Packet
71 Third Ave.
Burlington, MA 01803, USA
Email: hkaplan@acmepacket.com

Acknowledgment

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