Session Initiation Protocol (SIP) INFO Method Use

draft-burger-sip-info-02

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of 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 http://www.ietf.org/shadow.html.

This Internet-Draft will expire on May 21, 2008.

Copyright Notice

Copyright (C) The IETF Trust (2007).

Abstract

The purpose of the INFO request for the Session Initiation Protocol (SIP), as described by RFC 2976, is to provide mid-session SIP User Agent (UA)-to-SIP UA application data transport. In the years since the introduction of the INFO request, experience with the use of the INFO request indicates a number of problems. This document explains why there are INFO-based, proprietary protocols in the wild; the
flaws of using INFO; and explains why it is not possible to create a framework to rescue INFO for general purpose use. Since SIP has evolved considerably since the introduction of INFO, this document highlights some of the new, robust mechanisms for achieving the work that previously led people to use INFO. As these mechanisms are now available, this document formally deprecates the use of INFO for new usages beyond the existing standardized ones, namely RFC 3372 (SIP-T) and RFC 4497 (QSIG).

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 [1].

Table of Contents

1. Introduction .................................................. 3
2. Flaws With INFO .............................................. 4
   2.1. Message Context ........................................... 4
   2.2. Dialog Reuse ............................................. 6
   2.3. Other Issues ............................................. 7
3. Models for User Agent-User Agent Interaction .................. 7
   3.1. SUBSCRIBE / NOTIFY ...................................... 7
   3.2. Communication Channel ................................... 7
   3.3. INVITE Dialog Signaling ................................. 8
4. Alternatives for Common INFO Use ............................... 9
   4.1. State Updates ............................................ 9
   4.2. User Stimulus: Touch Tones and Others .................. 9
   4.3. Direct Signaling Channel ............................... 9
   4.4. Proxy-Aware Signaling ................................. 9
   4.5. Dialog Probe ........................................... 10
   4.6. Malicious Indicator .................................... 10
5. INFO Use Clarification ........................................ 11
6. Security Considerations ...................................... 11
7. IANA Considerations .......................................... 11
8. References .................................................... 12
   8.1. Normative References .................................... 12
   8.2. Informative References ................................ 12
Appendix A. Acknowledgements .................................... 13
Author’s Address .................................................. 14
Intellectual Property and Copyright Statements .................. 15
1. Introduction

There is a need for mid-session, Session Initiation Protocol (SIP) User Agent (UA)-to-SIP User Agent session layer signaling. Examples of such signaling include the following.

- Transporting foreign, non-SIP protocol messages for ISUP call setup
- Transport of supplemental dialed digits for ISUP or other call setup
- Transport of user stimulus to proxies and UAs
- Transport of generic DTMF digit entry
- SIP media server control
- SIP video encoding control
- SIP floor control
- Transport of application-specific data

The INFO [2] request transports mid-session signaling between two User Agents. These messages follow the signaling path established by the SIP INVITE, including visiting proxies that inserted themselves in the Record-Route path.

All of the examples above have implementations using the INFO request. There have been numerous Internet Drafts proposing the transport of DTMF using INFO. Likewise, there have been Internet Drafts describing the use of INFO for video encoding control (such as fast frame refresh requests) and conference floor control. RFC 3372 [3] describes the use of INFO for ISUP, also known as SIP-T. RFC 4497 [4] describes a similar use of INFO for QSIG. RFC 5022 [6] describes a use of INFO for media server control.

Clearly, there must be some advantages to using INFO, or people would not be using it. First and foremost, for many of these uses, INFO was the only option at the time. For example, MSCML’s inception predated SUBSCRIBE/NOTIFY by 18 months. Moreover, one of the driving concepts in MSCML is the concept of doing an operation on "this" leg. It is much easier to send a message on the SIP dialog, following an already established routing path, than to establish another end-to-end communication channel, such as a new SIP dialog, and referencing the target dialog.

One advantage of using any method in the signaling plane is reliable delivery. A common service provider customer service issue is end user devices not being able to transmit DTMF tones reliably across the service network. This is because the media plane does not have reliable delivery characteristics. This is by design, as the goal is to trade-off network latency and jitter for reliable packet delivery. Another advantage is that if the endpoint is only interested in the user signaling, they only need a signaling stack and access to the
much lower packet rate signaling channel, as opposed to having a media stack and receiving all of the media.

It is clear there are existence proofs for the use of INFO. However, there is a serious flaw with the INFO request. The INFO request itself has neither a context for interpreting any given message nor a negotiation method for accepting different INFO request types. One of the main reasons why INFO appears to work is most of the uses to date have been in limited or controlled deployments, where one entity controls the endpoints. For example, application servers, in a session with a media server, will not expect to receive user stimulus. Likewise, a routing proxy, such as the 3GPP IMS S-CSCF, will not be scaled to receive media server control messages, as that can be independent of subscriber size (call volume). Furthermore, a Voice-over-IP service provider may supply or strictly mandate the manufacturer and firmware for customer-premises equipment such as terminal adapters. However, with the further adoption of SIP, such collisions and misinterpretation of context becomes highly likely.

This document first describes the flaws with INFO. Then it offers alternatives for INFO that covers most of the use cases for which the work group has seen Internet Drafts in the past. This document describes how one can unambiguously create application session signaling that does require proxy traversal by using new SIP methods. Lastly, this document formally restricts the use of INFO to that described by RFC 3372 [3].

2. Flaws With INFO

There are two classes of issues with the INFO method. The first class of problem is INFO requests are totally without context. There is no indication of what the content means in the message. There are various mechanisms that could fix this problem. However, none are backwards compatible. The second class of problem is INFO requests are inextricably bound to the INVITE dialog. For some uses of INFO, this is not a problem. For others, this is a serious problem.

2.1. Message Context

There is no programmatic way of determining what the content of an INFO request means. From the User Agent’s point of view, a INFO request appears. Is this INFO request conveying a DTMF digit, a SIP-T encapsulated message, or a video update request? There is an argument saying the User Agent can figure it out. The content of the INFO request will have a MIME type. For example, SIP-T [7] messages will have a MIME type of application/ISUP, while MSCML [6] messages will have a MIME type of application/mediaservercontrol+xml.
Likewise, the endpoints can negotiate what MIME types they support, thus advertising their capabilities.

However, as we learned in the messaging community [8], relying on the MIME type alone is not sufficient to determine the context of the message. Clearly, as shown in the previous paragraph, the message content type relates to the message context. However, it is quite easy to imagine situations where the same content type has multiple meanings for a User Agent. For example, a DTMF digit type could be for user stimulus, post-dial digit collection, or the simple transport of a digit with no signaling context.

In addition, there are times when an endpoint will be hosting a number of different applications, each looking for different DTMF patterns. For example, a call management application may be looking for a long "#", while a messaging application may be looking for digits. Using INFO, or named tones, for that matter, each application has to examine each digit. Using subscription-based protocols such as KPML, one can limit the traffic and processing to only the tones the application has interest.

For that matter, there are application scenarios where an application separate from the endpoint needs to monitor for user stimulus. For example, a calling card application might want to monitor for a re-origination signal. Likewise, a lawful intercept trap and trace application wants to monitor only the user’s entered digits. With the INFO method, that application must insert itself in the signaling path. This requires the application to become a Back-to-Back User Agent (B2BUA), which means that it must handle all of the state transactions in the dialogs, as well as intrusively be in the call path.

An interesting issue is every INFO request traverses the same proxy path as any other dialog-related SIP request. Proxies in the path that have no interest in INFO requests still must process the request. This may put undue load on those proxies. What makes this issue interesting, is one may wish the request to traverse the proxy. The problem is there is no way for proxies to know whether or not they have an interest in INFO requests. Getting the requests is an all-or-nothing proposition, driven by Record-Route.

Let us consider these issues with respect to DTMF transport.

First of all, if the end device is using a compressed codec, the device will most likely use named tones [9]. If the device also sends DTMF in INFO messages, the device will be sending the digits multiple times. This would not be a problem if the endpoints could negotiate the use of INFO for DTMF transport. If they could, then
each device would know to ignore the named tone packets, which do not have reliable delivery characteristics. However, since there is no negotiation, the endpoints have to assume, when they receive a named tone packet, the packet represents DTMF user stimulus. When an INFO arrives with DTMF in it, the endpoint will double detect the digit.

One might argue that upon receipt of an INFO message with DTMF in it, the recipient could ignore named tone packets in the media stream. However, in almost all scenarios, the media stream will reach the end device well before an INFO will. A negotiation mechanism would solve this problem. If the endpoints explicitly agree to transport user signaling in the signaling channel, then they can safely ignore named tones in the media stream.

Unless the signaling path is secure, using S/MIME or sips, user digit entry is in the clear. This has clear security and privacy implications with respect to credit card numbers, account numbers, personal identification numbers, and so on.

One argument often heard for using INFO for DTMF is that it is easy and does not use very much bandwidth. However, studies of, for example, KPML versus INFO for DTMF [10], show significantly better bandwidth utilization for KPML [11], even with the supposed overhead of the SUBSCRIBE / NOTIFY [12] mechanism. This is because sending tones digit-by-digit in an INFO message is very inefficient.

### 2.2. Dialog Reuse

INFO, by design, is a method within an INVITE dialog usage. RFC 5057 [13] enumerates the problems with using dialogs for multiple usages, and we strongly urge the reader to review RFC 5057. The most relevant issue is a failure of transmission or processing of an INFO may render the dialog terminated. 

Prior to RFC 5057 it was underspecified if the INFO usage was a separate usage or not. However, RFC 5057 clarifies the INFO method is always part of the INVITE usage.

Some uses of INFO can tolerate this fate sharing of the INFO message over the entire dialog. For example, in the SIP-T usage, it may be acceptable for a call to fail, or to tear down the call, if one cannot deliver the associated SS7 information. This is not a value judgement on high availability service versus high availability billing, it is just an observation of service provider choice. However, it may not be acceptable for a call to fail if, for example, a DTMF buffer overflows. Then again, for some services, that may be the exact desired behavior. Again, this is not a value judgement on those who would build less than ideal user interfaces.
The issue is dialog reuse presents all of these problems. Alternatives which we will explore in Section 4 do not have these problems.

2.3. Other Issues

There is no throttling mechanism for INFO. Consider that most call signaling occurs on the order of 3 messages per minute. DTMF tones occur in bursts at a rate of 240 messages per minute. This is a considerably higher rate than for call signaling. As an Internet protocol, any mechanism we use must have some throttling mechanism in place.

3. Models for User Agent-User Agent Interaction

Models for User Agent-to-User Agent Interaction (UUI) include SUBSCRIBE / NOTIFY, establishing a communication channel associated with the SIP dialog, and issuing signaling over the INVITE-initiated SIP dialog.

3.1. SUBSCRIBE / NOTIFY

The first model for UUI is SUBSCRIBE / NOTIFY, RFC 3265 [12]. In this model, one user agent requests state information, such as key pad presses from a device to an application server or key map images from an application server to a device. The SUBSCRIBE creates a new dialog that does not share the fate of the related INVITE-initiated dialog. Moreover, using the SUBSCRIBE model enables multiple applications to receive state updates. These application can be outside the media path and potentially outside the INVITE-initiated dialog’s proxy path.

Implementors do need to be aware the prize of having a protocol that works in all cases, can scale, can easily load balance, and will not mysteriously fail a session in the event of state synchronization failure does come at a cost. Session establishment is a minimum of two messages in addition to the INVITE dialog establishment. If the SUBSCRIBE application is co-resident with the INVITE application, the application will have to manage two SIP dialogs instead of one. Tracking the UUI state dominates memory and processing for some applications, and as such the doubling of SIP dialogs is not an issue. However, for other applications, this may be an issue.

3.2. Communication Channel

The second model for UUI is to establish a communication channel. One model for this is MRCPv2 [14]. Here, the INVITE-initiated dialog
establishes a separate reliable, connection-oriented channel, such as
a TCP [15] or SCTP [16] stream. One uses SIP to locate the remote
endpoint, but uses a direct connection for the UUI. One then can
create whatever protocol one wishes, whether from whole-cloth (as in
MRCPv2) or using a substrate such as BEEP [17].

Another model is to use a totally externally signaled channel, such
as HTTP [18]. In this model, the user agent knows about a rendezvous
point to direct HTTP requests to for the transfer of information.
Examples include encoding of a prompt to retrieve in the SIP Request
URI in RFC 4240 [19] or the encoding of a SUBMIT target in a VoiceXML

3.3. INVITE Dialog Signaling

For situations where delivery or protocol failure of a UUI message
should terminate the INVITE-initiated dialog, one can envision
creating a framework for using a UUI method within the INVITE-
initiated dialog.

Such a framework would need to address the following issues.

- Fully-specified context: When such a UUI method arrives at a UAC,
  the UAC knows exactly the semantics of the message. Leveraging
  the terminology of RFC 3265, the method includes an indicator of
  the package the message belongs to. One can have further
  refinement of the UUI message type by using MIME types.
- Fully negotiated: When the UAC makes an offer, it can offer which
  UUI packages the UAC can send to the UAS as well as which UUI
  packages the UAC would like the UAS to send to the UAC. In the
  answer, the UAS can indicate which UUI packages it will use for
  the session. Following the model of RFC 3264 [21], the offer and
  answer can be late. Note we mention the model of RFC 3264 and not
  the protocol of RFC 3264, as such indication may well be in the
  SIP headers and not the SDP body.

Experience with IMAP [22] does offer a potential drawback of such a
scheme. In particular, the offer can be quite long.

It is important to note that INFO is in protocol jail unless one can
create a backward-compatible mechanism to negotiate packages. To
wit, if an upgraded UAC offers a package, such as DTMF, but the
server is not upgraded, the server will ignore the negotiation
request. At that point, the UAC has to assume the server will not
send or receive DTMF in INFO. Likewise, if an upgraded UAS receives
an INVITE without any package support indications, the UAS has to
assume the client will not send or receive DTMF in INFO.
4. Alternatives for Common INFO Use

What alternatives to INFO are there for UA-to-UA application session signaling? As noted above, there are three broad classes of session signaling available. The choice depends on the circumstances. Following is a list of situations that have used INFO in the past.

- State updates
- User stimulus
- Direct signaling channel
- Proxy-aware signaling
- Dialog probe

4.1. State Updates

This is the broad class of one User Agent updating another with changes in state. The design goal of the SUBSCRIBE/NOTIFY [12] event framework is to meet just this need.

4.2. User Stimulus: Touch Tones and Others

This is the class of the user entering stimulus at one User Agent, and the User Agent transporting that stimulus to the other. A key thing to realize is key presses on the telephone keypad is user stimulus. Thus, the appropriate mechanism to use here is KPML [11].

4.3. Direct Signaling Channel

State updates and user stimulus tend to have relatively few messages per session. Sometimes, User Agents need to exchange a relatively high number of messages. In addition, User Agents may have a need for a relatively low-latency exchange of messages. In this latter case, the User Agent may not be able to tolerate the latency introduced by intermediate proxies. Likewise, the intermediate proxies may have no interest in processing all of that data.

In this case, establishing a separate, direct control channel, as in MSRP [23] or MRCPv2 [14] is appropriate.

In addition, not every situation requires a SIP solution. Some signaling is really just one-shot to third-party endpoints. That situation may better be handled using an appropriate protocol, such as HTTP [18].

4.4. Proxy-Aware Signaling

Sometimes, one does want proxies to be in the signaling path for UA-to-UA application signaling. In this case, the use of a SIP request is appropriate. To date, there are no mechanisms for completely
disambiguating INFO requests. For example, one could create a registry of INFO packages. The definition of the package would define the contexts for the various MIME Content-Types, as well as the context of the request itself. However, a package can have multiple content types. Moreover, having the context, or package identifier, at the SIP level precludes bundling multiple contexts responding in the same INFO request. For example, a User Agent might want to bundle two different responses in a multipart/mixed MIME body type.

Because there is no difference in either the protocol machinery or registration process due to these factors, we will not create an INFO framework. If one needs a SIP User Agent-to-SIP User Agent application session signaling transport protocol that touches all Record-Route proxies in a path, one MUST create a new SIP method as described in Section 27.4 of RFC 3261 [5].

4.5. Dialog Probe

Some implementations in the wild use INFO to probe if an INVITE-initiated dialog is alive. While this works, it is NOT RECOMMENDED. In particular, RFC 4028 [24] describes how to ensure an INVITE-initiated dialog is alive.

4.6. Malicious Indicator

Take the case of Malicious Indicator. This is where a subscriber receives a call, realizes it is a malicious call (threatening, SPIT, etc.). They then press the SPIT button (or press *xx), which tells their service provider to mark the UAC as a bad actor. One might be tempted to think that INFO would be a great option for this service. It follows the return path of the INVITE, and so the INFO will hit the caller’s inbound proxy, which it can learn the caller is (statistically) a bad actor. That way the inbound proxy can do stuff like notify law enforcement, add a vote to "this is a SPIT source," or other useful action.

However, consider a few issues. First, since INFO lives exclusively within an established dialog, there is no way to assert this message after the call completes. Second, this mechanism *relies* on an active service provider topology. If there is no proxy in the chain that will eat the INFO, the caller will see the "this is a bad guy" message, which may have consequences in the real world. Third, there is no a’priori way for the UAS to know whether or not it can issue the INFO. The caller CERTAINLY will not advertise, "please tell me if I am bad, particularly I know in advance that I *am* a bad actor."

One approach is for the service provider’s proxy to SUBSCRIBE for the
SPIT event at the UAS. At this point, life is good, interoperable, and works across networks. This enables events after the dialog is torn down, as presumably the SPIT event will refer not to, "this dialog," which does not exist, but to "that dialog identifier," which exists (and is theoretically unique) forever.

Another approach that saves considerably on the overhead of subscriptions would be for the service provider to insert a HTTP URI in the initial INVITE, noting it is for reporting malicious behavior. When the subscriber presses the SPIT button, an HTTP POST gets executed, delivering the call information to the service provider. The service provider can encode basic call information in the HTTP URI and can instruct the device to send whatever arbitrary data is necessary in the POST. This method has the added benefit of being entirely outside the real-time SIP proxy network.

5. INFO Use Clarification

There is no way to unambiguously use the INFO request in a general framework. The IETF has already standardized use of INFO for SIP-T [3] and QSIG [4]. Thus we will not deprecate the use of INFO for those purposes. However, this document explicitly updates INFO [2], in that one MUST NOT use the INFO request for anything other than the use described by RFC 3372 for ISUP and RFC 4497 for QSIG.

In recognition of existing, proprietary use of INFO, proxies MUST NOT take any action other than that described by RFC 3261 and RFC 2976 with respect to handling INFO requests.

6. Security Considerations

By eliminating the multiple uses of INFO messages without adequate community review, and by eliminating the possibility for rogue SIP User Agents from confusing another User Agent by purposely sending unrelated INFO messages, we expect the INFO use clarification to improve the security of the Internet.

7. IANA Considerations

None.

8. References
8.1. Normative References


8.2. Informative References


[13] Sparks, R., "Multiple Dialog Usages in the Session Initiation


Appendix A. Acknowledgements

We are standing on the shoulders of giants. Jonathan Rosenberg did the original "INFO Considered Harmful" Internet Draft on 26 December 2002, which influenced the work group and this document. Likewise, Dean Willis influenced the text from his Internet Draft, "Packaging
and Negotiation of INFO Methods for the Session Initiation Protocol" of 15 January 2003. My, we have been working on this for a long time!

This draft has elicited well over 450 messages on the SIP list. People who have argued with its thesis, supported its thesis, added to the examples, or argued with the examples, include the following individuals:

Adam Roach, Bram Verburg, Brian Stucker, Chris Boulton, Cullen Jennings, Dale Worley, Dean Willis, Frank Miller, Gonzalo Camarillo, Gordon Beith, Henry Sinnreich, James Jackson, James Rafferty, Jeroen van Bemmelen, Joel Halpern, John Elwell, Johnathan Rosenberg, Juha Heinanen, Keith Drage, Kevin Attard Compagno, Manpreet Singh, Martin Dolly, Mary Barnes, Michael Procter, Paul Kyzivat, Peili Xu, Peter Blatherwick, Raj Jain, Rayees Khan, Robert Sparks, Roland Jesske, Salvatore Loreto, Sam Ganesan, Sanjay Sinha, Spencer Dawkins, Steve Langstaff, Sumit Garg, and Xavier Marjou.

Christer Holmberg and Hadriel Kaplan provided numerous counter examples that helped hone the message of this document.

John Elwell and Francois Audet helped with QSIG references. In addition, Francois Audet provided actual text for the revised abstract.

Author’s Address

Eric W. Burger
BEA Systems, Inc.
USA

Email: eburger@standardstrack.com
URI: http://www.standardstrack.com