

SIPCORE Working Group	C. Holmberg	
Internet-Draft	Ericsson	
Updates: 3262 (if approved)	December 9, 2010	
Intended status: Standards Track		
Expires: June 12, 2011		

[TOC](#)

Session Initiation Protocol (SIP) Response Code for Indication of Terminated Dialog

draft-ietf-sipcore-199-03.txt

Abstract

This specification defines a new Session Initiation Protocol (SIP) response code, 199 Early Dialog Terminated, that a SIP forking proxy and a User Agent Server (UAS) can use to indicate towards upstream SIP entities (including the User Agent Client (UAC)) that an early dialog has been terminated, before a final response is sent towards the SIP entities. In addition, this specification updates section 4 of RFC 3262.

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 12, 2011.

Copyright Notice

Copyright (c) 2010 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 Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	
2.	Terminology	
3.	Applicability and Limitation	
4.	User Agent Client behavior	
5.	User Agent Server behavior	
6.	Proxy behavior	
7.	Backward compatibility	
8.	Usage with SDP offer/answer	
9.	Usage with 100rel	
10.	Normative update of RFC 3262	
10.1.	General	
10.2.	RFC3262: 4. UAC Behavior	
11.	Message Flow Examples	
11.1.	Example with a forking proxy which generates 199	
11.2.	Example with a forking proxy which receives 200 OK	
11.3.	Example with two forking proxies, of which one generates 199	
12.	Security Considerations	
13.	IANA Considerations	
13.1.	IANA Registration of the 199 response code	
13.2.	IANA Registration of the 199 option-tag	
13.3.	RFC 3262 Update	
14.	Acknowledgements	
15.	Change Log	
16.	References	
16.1.	Normative References	
16.2.	Informational References	
§	Author's Address	

1. Introduction

[TOC](#)

As defined in RFC 3261 [\[RFC3261\]](#) (Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol," June 2002.), a Session Initiation Protocol (SIP) early dialog is created when a non-100 provisional response is sent to the initial dialog initiation request (e.g. INVITE, outside an existing dialog). The dialog is considered to be in early state until a final response is sent.

When a proxy receives an initial dialog initiation request, it can forward the request towards multiple remote destinations. When the proxy does that, it performs forking [\[RFC3261\] \(Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol," June 2002.\)](#).

When a forking proxy receives a non-100 provisional response, or a 2xx final response, it forwards the response upstream towards the sender of the associated request. After a forking proxy has forwarded a 2xx final response, it normally generates and sends CANCEL requests downstream towards all remote destinations where it previously forked the request associated with the 2xx final response and from which it has yet not received a final response. The CANCEL requests are sent in order to terminate any outstanding early dialogs associated with the request. Upstream SIP entities might receive multiple 2xx final responses. When a SIP entity receives the first 2xx final response, and it does not intend to accept any subsequent 2xx final response, it will automatically terminate any other outstanding early dialog associated with the request. If the SIP entity receives a subsequent 2xx final response, it will normally generate and send an ACK request, followed with a BYE request, using the dialog identifier retrieved from the 2xx final response.

NOTE: A User Agent Client (UAC) can use the Request-Disposition header field [\[RFC3841\] \(Rosenberg, J., Schulzrinne, H., and P. Kyzivat, "Caller Preferences for the Session Initiation Protocol \(SIP\)," August 2004.\)](#) to request that proxies do not generate and send CANCEL requests downstream once they have received the first 2xx final response.

When a forking proxy receives a non-2xx final response, it does not always immediately forward the response upstream towards the sender of the associated request. Instead, the proxy "stores" the response and waits for subsequent final responses from other remote destinations where the associated request was forked. At some point the proxy uses a specified mechanism to determine the "best" final response code, and forwards a final response using that response code upstream towards the sender of the associated request. When an upstream SIP entity receives the non-2xx final response it will release resources associated with the session. The UAC will terminate, or retry, the session setup. Since the forking proxy does not always immediately forward non-2xx final responses, upstream SIP entities (including the UAC that initiated the request) are not immediately informed that an early dialog has been terminated, and will therefor maintain resources associated with the early dialog reserved until a final response is sent by the proxy, even if the early dialog has already been terminated. A SIP entity could use the resources for other things, e.g. to accept subsequent early dialogs that it otherwise would reject. This specification defines a new SIP response code, 199 Early Dialog Terminated. A forking proxy can send a 199 provisional response to inform upstream SIP entities that an early dialog has been terminated.

A UAS can send a 199 response code, prior to sending a non-2xx final response, for the same purpose. SIP entities that receive the 199 response can use it to release resources associated with the terminated early dialog. In addition, SIP entities might also use the 199 provisional response to make policy related decisions related to early dialogs.

This specification updates RFC 3262 [\[RFC3841\]](#) (Rosenberg, J., Schulzrinne, H., and P. Kyzivat, "Caller Preferences for the Session Initiation Protocol (SIP)," August 2004.), by mandating a UAC to be prepared to receive unreliably sent provisional responses even if it has required provisional responses to be sent reliably.

2. Terminology

[TOC](#)

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 [\[RFC2119\]](#) (Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels," March 1997.).

3. Applicability and Limitation

[TOC](#)

The 199 response code is an optimization, and it only optimizes how quickly recipients might be informed about terminated early dialogs. The achieved optimization is limited. Since the response is normally not sent reliably by an UAS, and can not be sent reliably when generated and sent by a proxy, it is possible that some or all of the 199 responses get lost before they reach the recipients. In such cases, recipients will behave the same as if the 199 response code were not used at all.

One example for which a UA could use the 199 response, is that when it receives a 199 response it releases resources associated with the terminated early dialog. It could also use the 199 response to make policy related decisions related to early dialogs. For example, if a UAC is playing media associated with an early dialog, and the it receives a 199 response indicating the early dialog has been terminated, it could start playing media associated with a different early dialog.

Applications designers utilizing the 199 response code MUST ensure that the application's user experience is acceptable if all 199 responses are lost, and not delivered to the recipients.

4. User Agent Client behavior

[TOC](#)

When a UAC sends an initial request, and if it is willing to receive 199 responses, it MUST insert the "199" option-tag in the Supported header field. The option-tag indicates that the UAC supports 199 responses. The UAC SHOULD NOT insert the "199" option-tag in the Require or the Proxy-Require header fields, since in many cases it would result in unnecessary session establishment failures.

When a UAC receives a 199 response it might release resources associated with the terminated early dialog. It might also use the 199 response to make policy related decisions related to early dialogs.

NOTE: The 199 response indicates that the early dialog has been terminated, so there is no need for the UAC to send a BYE request in order to terminate the early dialog when it receives the 199 response.

NOTE: The 199 response does not affect other early dialogs associated with the session establishment. For those the normal SIP rules, regarding transaction timeout etc, still apply.

Once the UAC has received and accepted the 199 provisional response, it MUST NOT send or process any media associated with the early dialog that was terminated.

If multiple usages [\[RFC5057\] \(Sparks, R., "Multiple Dialog Usages in the Session Initiation Protocol," November 2007.\)](#) are used within an early dialog, and it is not clear which dialog usage the 199 response terminates, SIP entities that keep dialog state SHALL NOT release resources associated with the early dialog when they receive the 199 response.

If a SIP entity receives an unreliable 199 response on a dialog which has not previously been established (this can happen if a 199 response reaches the client before the 18x response that would establish the early dialog) it SHALL discard the 199 responses. If a SIP entity receives a reliable 199 response on a dialog which has not previously been created the UAC MUST acknowledge the 199 response, as described in RFC 3262.

If the UAC has received a 199 response for all early dialogs, and no early dialog associated session establishment remains, the UAC maintains the "Proceeding" state [\[RFC3261\] \(Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol," June 2002.\)](#) and waits for possible subsequent early dialogs to be established, and eventually for a final response to be received.

5. User Agent Server behavior

[TOC](#)

If a UAS receives an initial request that contains an "199" option-tag, it SHOULD NOT send a 199 response on an early dialog on which it

intends to send a final response, unless it e.g. has been configured to do so due to lack of 199 support by forking proxies or other intermediate SIP entities.

NOTE: If the UAS has created multiple early dialogs (the UAS is acting similar to a forking proxy), it does not always intend to send a final response for all of those dialogs.

When a UAS generates a 199 response, the response MUST contain a To header field tag parameter, in order to identify the early dialog that has been terminated. The UAS MUST also insert a Reason header field [[RFC3326](#)] ([Schulzrinne, H., Oran, D., and G. Camarillo, "The Reason Header Field for the Session Initiation Protocol \(SIP\)," December 2002.](#)) that contains a response code which describes the reason why the early dialog was terminated.

If the UAS intends to send 199 responses, and if it supports the procedures defined in RFC 3840 [[RFC3840](#)] ([Rosenberg, J., Schulzrinne, H., and P. Kyzivat, "Indicating User Agent Capabilities in the Session Initiation Protocol \(SIP\)," August 2004.](#)), it MAY during the registration procedure use the sip.extensions feature tag [[RFC3840](#)] ([Rosenberg, J., Schulzrinne, H., and P. Kyzivat, "Indicating User Agent Capabilities in the Session Initiation Protocol \(SIP\)," August 2004.](#)) to indicate support of the 199 response code.

A 199 response SHOULD NOT contain an SDP offer/answer message body, unless required by the rules in RFC 3264 [[RFC3264](#)] ([Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol \(SDP\)," June 2002.](#)).

According to RFC 3264, if the INVITE request does not contain an SDP offer, and the 199 response is the first reliably sent response, the 199 response is required to contain an SDP offer. In this case the UAS SHOULD send the 199 response unreliably, or include an SDP offer with no m- lines in the reliable 199 response.

Since the provisional response is only used for information purpose, the UAS SHOULD send it unreliably, unless the "100rel" option-tag [[RFC3262](#)] ([Rosenberg, J. and H. Schulzrinne, "Reliability of Provisional Responses in Session Initiation Protocol \(SIP\)," June 2002.](#)) is present in the Require header field of the associated request.

Once the UAS has sent a 199 response, it MUST NOT send or process any media associated with the terminated early dialog.

6. Proxy behavior

[TOC](#)

When a proxy receives a 199 response, it MUST process the response as any other non-100 provisional responses. The proxy will forward the response upstream towards the sender of the associated request. The proxy MAY release resources it has reserved associated with the early dialog that is terminated. If a proxy receives a 199 response out of

dialog, it processes it as other non-100 provisional responses received out of dialog.

When a forking proxy receives a non-2xx final response that it recognizes as terminating one or more early dialogs, it MUST generate and send a 199 response upstream for each of the terminated early dialogs that satisfy each of the following conditions:

- the forking proxy does not intend to forward the final response immediately (in accordance with rules for a forking proxy)
- the UAC has indicated support (using the "199" option-tag) for the 199 response code
- the forking proxy has not already received and forwarded a 199 response for the early dialog
- the forking proxy has not already sent a final response for any of the early dialogs

As a consequence, once a final response to the INVITE has been issued by the proxy, no further 199 responses associated with the INVITE request will be generated or forwarded by the proxy.

When the forking proxy forks the initial request, it generates a unique Via header branch parameter value for each forked leg. The proxy can determine whether additional forking has occurred downstream of the proxy by storing the top Via branch value from each response which creates an early dialog. If the same top Via branch value is received for multiple early dialogs, the proxy knows that additional forking has occurred downstream of the proxy. A non-2xx final response received for a specific early dialog also terminates all other early dialog for which the same top Via branch value was received in the responses which created those early dialogs.

Based on implementation policy, the forking proxy MAY wait before sending the 199 response, e.g. if it expects to receive a 2xx final response on another dialog shortly after it received the non-2xx final response which triggered the 199 response.

When a forking proxy generates a 199 response, the response MUST contain a To header field tag parameter, that identifies the terminated early dialog. The proxy MUST also insert a Reason header field that contains the SIP response code of the response that triggered the 199 response. The SIP response code in the Reason header field informs the receiver of the 199 response about the SIP response code that was used by the UAS to terminate the early dialog, and the receiver might use that information for triggering different types of actions and procedures.

A forking proxy that supports generating of 199 responses MUST keep track of early dialogs, in order to determine whether to generate a 199 response when the proxy receives a non-2xx final response. In addition, the proxy MUST keep track on which early dialogs it has received and forwarded 199 responses, in order to not generate additional 199 responses for those early dialogs.

If a forking proxy receives a reliably sent 199 response for a dialog, for which it has previously generated and sent a 199 response, it MUST forward the 199 response. If the proxy receives an unreliably sent 199

response, for which it has previously generated and sent a 199 response, it MAY forward the response, or it MAY discard it. When a forking proxy generates and sends a 199 response, it MUST NOT send the response reliably. When a forking proxy generates and sends a 199 response, the response SHOULD NOT contain a Contact header field or a Record-Route header field [\[RFC3261\]](#) (Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol," June 2002.).

7. Backward compatibility

[TOC](#)

Since all SIP entities involved in a session setup do not necessarily support the specific meaning of the 199 Early Dialog Terminated provisional response, the sender of the response MUST be prepared to receive SIP requests and responses associated with the dialog for which the 199 response was sent (a proxy can receive SIP messages from either direction). If such request is received by a UA, it MUST act in the same way as if it had received the request after sending the final non-2xx response to the INVITE request, as specified in RFC 3261. A UAC that receives a 199 response for an early dialog MUST NOT send any further requests on that dialog, except for requests which acknowledge reliable responses. A proxy MUST forward requests according to RFC 3261, even if the proxy has knowledge that the early dialog has been terminated.

A 199 response does not "replace" a final response. RFC 3261 specifies when a final response is sent.

8. Usage with SDP offer/answer

[TOC](#)

A 199 response SHOULD NOT contain an SDP offer/answer [\[RFC3264\]](#) (Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)," June 2002.) message body, unless required by the rules in RFC 3264.

If an INVITE request does not contain an SDP offer, and the 199 response is the first reliably sent response, the 199 response is required to contain an SDP offer. In this case the UAS SHOULD send the 199 response unreliable, or include an SDP offer with no m- lines in a reliable 199 response.

[TOC](#)

9. Usage with 100rel

Since the provisional response is only used for information purpose, the UAS SHOULD send it unreliably, unless the "100rel" option-tag [\[RFC3262\] \(Rosenberg, J. and H. Schulzrinne, "Reliability of Provisional Responses in Session Initiation Protocol \(SIP\)," June 2002.\)](#) is present in the Require header field of the associated request.

NOTE: Implementors need to ensure that a 199 response that is sent unreliably, even if the associated INVITE request contained a Require header field with an "100rel" option-tag, does not trigger errors or rejection of the 199 response.

When a forking proxy generates and sends a 199 response, it MUST NOT send the response reliably.

NOTE: If the forking proxy would generate a reliable 199 response, it would have to terminate the associated PRACK [\[RFC3262\] \(Rosenberg, J. and H. Schulzrinne, "Reliability of Provisional Responses in Session Initiation Protocol \(SIP\)," June 2002.\)](#) request.

10. Normative update of RFC 3262

[TOC](#)

10.1. General

[TOC](#)

The paragraph in this section is added to section 4 of RFC 3262.

10.2. RFC3262: 4. UAC Behavior

[TOC](#)

The UAC MUST support reception of all provisional responses, sent reliably or unreliably; use of the option tag value "100rel" in a Require header field does not change this requirement.

11. Message Flow Examples

[TOC](#)

[TOC](#)

11.1. Example with a forking proxy which generates 199

The figure shows an example, where a proxy (P1) forks an INVITE received from UAC. The forked INVITE reaches UAS_2, UAS_3 and UAS_4, which send 18x provisional responses in order to establish early dialogs between themselves and the UAC. UAS_2 and UAS_3 reject the INVITE by sending a 4xx error response each. When P1 receives the 4xx responses it immediately sends 199 responses towards the UAC, to indicate that the early dialogs for which it received the 4xx responses have been terminated. The early dialog leg is shown in parenthesis.

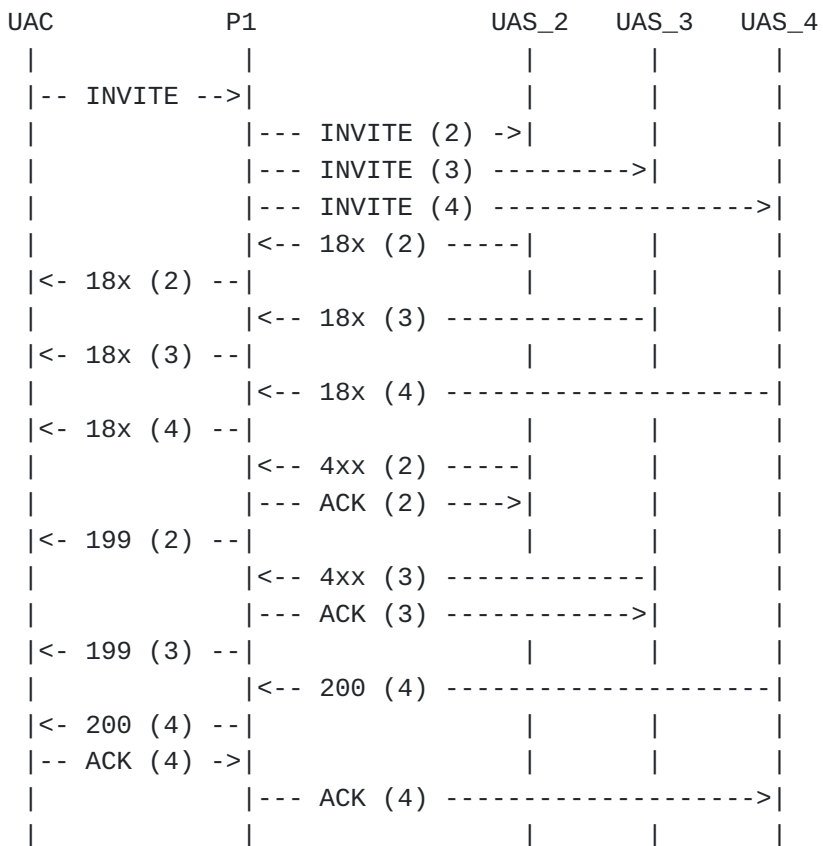


Figure 1: Example call flow

11.2. Example with a forking proxy which receives 200 OK

The figure shows an example, where a proxy (P1) forks an INVITE request received from UAC. The forked request reaches UAS_2, UAS_3 and UAS_4, that all send 18x provisional responses in order to establish early dialogs between themselves and the UAC. Later UAS_4 accepts the session and sends a 200 OK final response. When P1 receives the 200 OK responses it immediately forwards it towards the UAC. P1 does not send 199 responses for the early dialogs from UAS_2 and UAS_3, since P1 has yet not received any final responses on those early dialogs (even if P1 sends CANCEL requests to UAS_2 and UAS_3 P1 may still receive 200 OK final response from UAS_2 or UAS_3, that P1 would have to forward towards the UAC. The early dialog leg is shown in parenthesis.

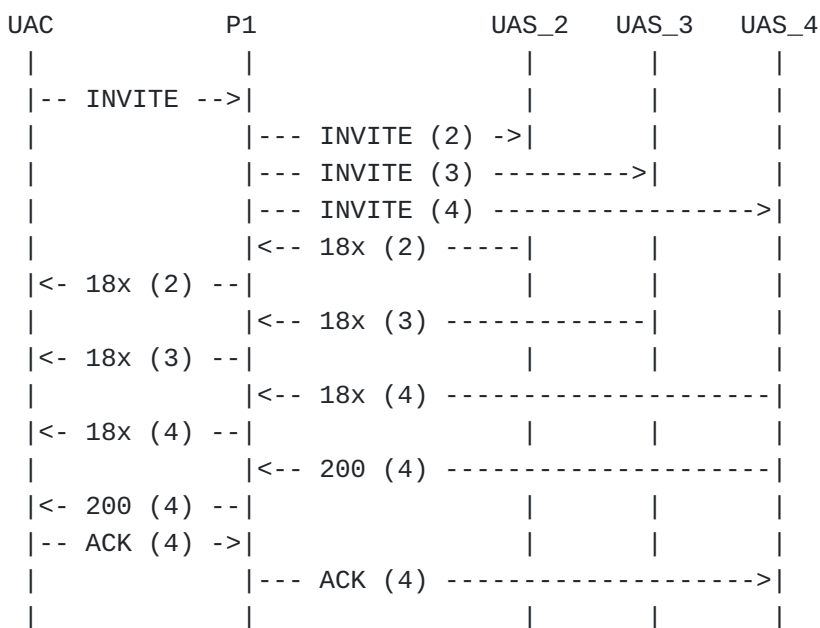


Figure 2: Example call flow

11.3. Example with two forking proxies, of which one generates 199

[TOC](#)

The figure shows an example, where a proxy (P1) forks an INVITE request received from UAC. One of the forked requests reaches UAS_2. The other requests reach another proxy (P2), that forks the request to UAS_3 and

UAS_4. UAS_3 and UAS_4 send 18x provisional responses in order to establish early dialogs between themselves and UAC. Later UAS_3 and UAS_4 reject the INVITE request by sending a 4xx error response each. P2 does not support the 199 response code, and forwards a single 4xx response. P1 supports the 199 response code, and when it receives the 4xx response from P2, it also manages to associate the early dialogs from both both UAS_3 and UAS_4 with the response. Therefor it generates and sends two 199 responses to indicate that the early dialogs from UAS_3 and UAS_4 have been terminated. The early dialog leg is shown in parenthesis.

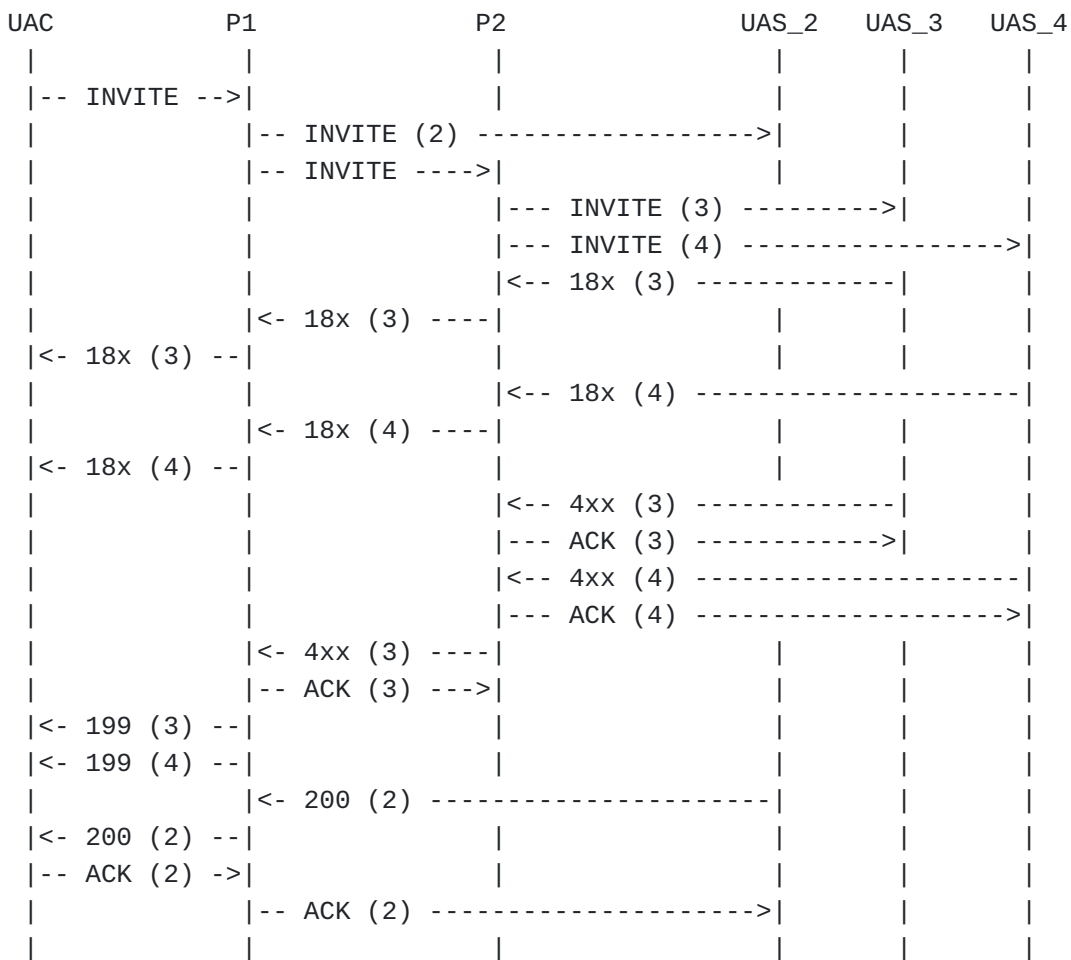


Figure 3: Example call flow

12. Security Considerations

General security issues related to SIP responses are described in [RFC3261] (Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol," June 2002.). Due to the nature of the 199 response, it may be attractive to use it for launching attacks in order to terminate specific early dialogs (other early dialogs will not be affected). In addition, if a man-in-the-middle sends a 199 response to the UAC, which terminates a specific dialog, it can take a while until the UAS finds out that the UAC, and possible stateful intermediates, have terminated the dialog. SIP security mechanisms (e.g. hop-to-hop TLS) can be used to minimize, or eliminate, the risk for such attacks.

13. IANA Considerations

[TOC](#)

This section registers a new SIP response code and a new option-tag, according to the procedures of RFC 3261, and updates section 4 of RFC 3262.

13.1. IANA Registration of the 199 response code

[TOC](#)

This section registers a new SIP response code, 199. The required information for this registration, as specified in RFC 3261, is:

RFC Number: RFC XXXX [[NOTE TO IANA: Please replace XXXX with the RFC number of this specification]]

Response Code Number: 199

Default Reason Phrase: Early Dialog Terminated

13.2. IANA Registration of the 199 option-tag

[TOC](#)

This section registers a new SIP option-tag, 199. The required information for this registration, as specified in RFC 3261, is:

Name: 199

Description: This option-tag is for indicating support of the 199 Early Dialog Terminated provisional response code. When present in a Supported header, it indicates that the UA supports the response code. When present in a Require header in a request, it indicates that the UAS MUST support the sending of the response code.

13.3. RFC 3262 Update

[TOC](#)

This document updates section 4 of RFC 3262.

14. Acknowledgements

[TOC](#)

Thanks to Paul Kyzivat, Dale Worley, Gilad Shaham, Francois Audet, Attila Sipos, Robert Sparks, Brett Tate, Ian Elz, Hadriel Kaplan, Timothy Dwight, Dean Willis, Serhad Doken, John Elwell, Gonzalo Camarillo, Adam Roach, Bob Penfield, Tom Taylor, Ya Ching Tan, Keith Drage, Hans Erik van Elburg and Cullen Jennings for their feedback and suggestions.

15. Change Log

[TOC](#)

[RFC EDITOR NOTE: Please remove this section when publishing]
Changes from draft-ietf-sipcore-199-02

- *Usage example section rewritten and clarified
- *Requirement has been removed
- *SIP has been added to document title
- *Acronyms expanded in the abstract and throughout the document
- *Editorial fixes throughout the document
- *Indication added that document is aimed for standards track
- *Some references made informative

- *Additional text added regarding the usage of the Reason header
- *SBC latching text has been removed
- *Usage of Require/Proxy-Require header removed
- *Additional text added regarding sending SDP offer in 199
- *Note added, which clarifies that 199 does not affect other early dialogs
- *References added to Security Considerations
- *Clarification of local ringing tone
- *Clarification that media must not be sent or processed after 199
- *Text regarding sending media on terminated dialogs added to security section
- *Change: UAS must send 199 reliably in case of Require:100rel
- *Change: Section 4 of RFC 3262 updated

16. References

[TOC](#)

16.1. Normative References

[TOC](#)

[RFC2119]	Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels," BCP 14, RFC 2119, March 1997 (TXT , HTML , XML).
[RFC3261]	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 (TXT).
[RFC3262]	Rosenberg, J. and H. Schulzrinne, " Reliability of Provisional Responses in Session Initiation Protocol (SIP) ," RFC 3262, June 2002 (TXT).
[RFC3264]	Rosenberg, J. and H. Schulzrinne, " An Offer/Answer Model with Session Description Protocol (SDP) ," RFC 3264, June 2002 (TXT).
[RFC3326]	

	Schulzrinne, H., Oran, D., and G. Camarillo, " The Reason Header Field for the Session Initiation Protocol (SIP) ," RFC 3326, December 2002 (TXT).
[RFC3840]	Rosenberg, J., Schulzrinne, H., and P. Kyzivat, " Indicating User Agent Capabilities in the Session Initiation Protocol (SIP) ," RFC 3840, August 2004 (TXT).

16.2. Informational References

[TOC](#)

[RFC3841]	Rosenberg, J., Schulzrinne, H., and P. Kyzivat, " Caller Preferences for the Session Initiation Protocol (SIP) ," RFC 3841, August 2004 (TXT).
[RFC5057]	Sparks, R., " Multiple Dialog Usages in the Session Initiation Protocol ," RFC 5057, November 2007 (TXT).
[3GPP. 24.182]	3GPP, " IP Multimedia Subsystem (IMS) Customized Alerting Tones (CAT); Protocol specification ," 3GPP TS 24.182.
[3GPP. 24.628]	3GPP, " Common Basic Communication procedures using IP Multimedia (IM)Core Network (CN) subsystem; Protocol specification ," 3GPP TS 24.628.

Author's Address

[TOC](#)

	Christer Holmberg
	Ericsson
	Hirsalantie 11
	Jorvas 02420
	Finland
Email:	christer.holmberg@ericsson.com