Response Code for Indication of Terminated Dialog
draft-ietf-sip-199-02.txt

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 April 23, 2009.

Abstract

This specification defines a new SIP response code, 199 Early Dialog Terminated, which a SIP entity can use to indicate upstream that an early dialog has been terminated. The response code can be used by a SIP entity to indicate to the UAC that an early dialog has been terminated, before a final response is sent to the UAC.
Table of Contents

1. Introduction ................................................. 3
2. Conventions ................................................... 4
3. Requirements .................................................. 4
4. Client behavior .............................................. 4
   4.1. Examples of resource types ............................. 5
5. Server behavior ............................................... 6
6. Proxy behavior ............................................... 6
7. Backward compatibility ....................................... 7
8. 199 Early Dialog Terminated ................................. 7
9. Usage with SDP offer/answer ................................. 7
10. Usage with 100rel ........................................... 7
11. Example ....................................................... 8
12. Security Considerations ..................................... 9
13. IANA Considerations ........................................ 9
   13.1. IANA Registration of the 199 response code .......... 9
   13.2. IANA Registration of the 199 Option Tag ............. 9
14. Acknowledgements ............................................ 9
15. Normative References ....................................... 9
Author’s Address ............................................... 10
Intellectual Property and Copyright Statements ............. 11
1. Introduction

As defined in SIP (Session Initiation Protocol) specification [RFC3261], an early SIP dialog is created when a non-100 provisional response is sent to the dialog initiation request (e.g. INVITE). The dialog is considered to be in early state until a final response is sent.

When a proxy receives an initial request (outside an existing dialog, and without a pre-defined route set), it can forward it towards multiple remote destinations. When the proxy does that, it performs forking.

When a forking proxy receives non-100 provisional responses, it forwards the responses upstream towards the sender of the associated request. When a forking proxy receives a 2xx final response, it forwards the response upstream towards the sender of the associated request. At that point the proxy normally sends a CANCEL request downstream towards all remote destinations where it previously sent the request associated with the 2xx final response, and from which it has not received a final response, in order to terminate associated outstanding early dialogs. It is possible to receive multiple 2xx final responses. When SIP entities upstream receive the first 2xx final response, and they do not intend to accept subsequent 2xx final responses, they will automatically terminate other associated outstanding early dialogs. If additional 2xx final responses are received, for INVITE initiated dialogs those SIP entities will normally send a BYE request using the dialog identifier retrieved from the subsequent 2xx final response.

NOTE: A UAC can use the Request-Disposition header [RFC3841] to request that proxies do not send CANCEL requests downstream once they have received the first final 2xx 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 forking proxy "stores" it and waits for further final responses from remote destinations where the forked request was forwarded. At some point the proxy uses a specified mechanism to determine the "best" final response code, and forwards that final response upstream towards the sender of the associated request. When SIP entities upstream receive the non-2xx final response they will automatically terminate the session setup and all associated early dialogs.

Since the forking proxy does not always immediately forward non-2xx final responses, SIP entities upstream (including the UAC that initiated the request) do not know that a specific early dialog has
been terminated, and the SIP entities keep possible resources associated with the early dialog until they receive a final response from the forking proxy.

This specification defines a new SIP response code, 199 Early Dialog Terminated, which a forking proxy and a UAS can use to indicate upstream that an early dialog has been terminated. The 199 response can also be sent by an UAS, prior to sending a non-2xx final response. SIP entities that receive the 199 provisional response MAY release resources associated with the specific early dialog.

2. Conventions

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 BCP 14, RFC 2119 [RFC2119].

3. Requirements

REQ 1: It must be possible to indicate to the UAC that an early dialog has been terminated before a final response is sent.

4. Client behavior

When a client sends an initial request it MUST insert the 199 option-tag in the Supported header, which indicates that the client supports the 199 Early Dialog Terminated response code.

When a client receives a 199 response it MAY release resources and procedures associated with the early dialog the 199 response is received on. Examples of resources and procedures are e.g. procedures for the establishment of media plane resources (bandwidth, radio, codecs etc), media security procedures or procedures related to NAT traversal.

If multiple usages [RFC5057] are used within an early dialog, and it is not clear which dialogusage 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 client receives a 199 response on a dialog which has not previously been created (this can happen if a 199 response reaches the client before a 18x response) the client SHALL discard the 199 responses.
4.1. Examples of resource types

Examples which benefit from resource-release are:

1. Codec release - when resources for a specific codec has been reserved only for the stream that is terminated. In that case the resources associated with that codec can be released.

2. Pre-conditions - when the dialog is terminated, procedures and resources associated to the pre-conditions for that dialog can be released.

3. In-band security negotiation - when the dialog is terminated, procedures and resources associated with the in-band security negotiation for that dialog can be released.

4. ICE [ref needed] mechanism - when the dialog is terminated, procedures and resources associated with the ICE related in-band procedures for that dialog can be released.

5. Limited access resources - in case of forking and multiple stream it may not be possible to allow early media on all dialogs, so some dialogs may e.g. be set to "inactive". When a dialog is terminated, media can be allowed on other dialogs.

If the client is able to associate the 199 response with a specific media stream, it MAY choose to discard media on that specific media stream, it MAY release all resources associated with that media stream and it MAY start to process media streams received on other early dialogs. When the P-Early-Media header is used, a UA may trigger different actions depending on whether the header has been used for the terminated dialog. How the association between the dialog and the associated media stream is done is outside the scope of this document.

NOTE: When using SRTP [RFC3261], the secure media stream is bound to the crypto context setup for the dialog, and can be identified using the MKI of SRTP.

If the client only has a single early dialog (other early dialogs may not have been established, or they may have been established and later terminated) when a 199 response is received for that early dialog, after the client has terminated the early dialog associated with the 199 response it will act as before the first early dialog was established.
5. Server behavior

If the received initial request contains an 199 option tag, and the server has established an early dialog with the initiator of the request, the server MAY send a 199 response prior to sending a non-200 final response towards the initiator of the request.

If the server intends to send 199 responses, and if the server supports the procedures defined in [RFC3840], it MAY during the registration procedure use the sip.extensions feature tag [RFC3840] to indicate support of the 199 response code.

OPEN ISSUE: We still need to describe when a UAS sends a 199 response. One potential solution is to define an indicator which a forking proxy that supports 199 can insert in the request, indicating that the UAS does not need to send 199.

6. Proxy behavior

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

When a forking proxy receives a non-2xx final response which terminates an early dialog and the proxy does not intend to forward the final response immediately (due to the rules for a forking proxy), and the UAC has indicated support of the 199 response code, the proxy MUST generate and send a 199 provisional response upstream for that early dialog, unless the proxy prior has received and forwarded a 199 response for that early dialog. The 199 provisional response MUST contain a To header tag parameter, which identifies the early dialog that has been terminated.

A proxy which supports generating of 199 response codes 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.

NOTE: If the non-2xx final response is received from another forking proxy upstream, the final response may terminate multiple early dialogs. If the forking proxy which receives the final response is able to associate the final response with multiple early dialogs, the forking proxy can generate 199 responses for all of those early
7. Backward compability

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 may receive SIP messages from either direction). If such request is received by a UA, it MUST reply to such requests with a 481 final response. 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 [RFC3261], even if the proxy has knowledge that the early dialog has been terminated.

The 199 Early Dialog Terminated response code MUST NOT "replace" a final response. A final response MUST always be sent, after one or many 199 responses have been sent.

8. 199 Early Dialog Terminated

The 199 Early Dialog Terminated response code allows a SIP entity to indicate upstream that a specific dialog has been terminated, before a final response is sent by the entity. The To header tag value is used to identify the dialog.

9. Usage with SDP offer/answer

A 199 Early Dialog Terminated provisional response MUST NOT contain a new SDP offer/answer message body, but the sender of the response MAY insert a copy of a previously sent offer/answer message body as otherwise allowed by the offer/answer rules for a provisional response.

10. Usage with 100rel

When a 199 Early Dialog Terminated provisional response is sent by a UAS, since the provisional response is only used for information purpose, the UAS SHOULD send it unreliably even if the 100rel option tag [RFC3262] is present in the Require header of the associated request.
When a forking proxy triggers a 199 response, the response MUST NOT be sent reliably.

NOTE: The 199 response MUST NOT be sent reliably if it would be required to insert a new SDP offer/answer message body in the response, according to the rules in [RFC3264].

11. Example

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 create early dialogs between themselves and the UAC. UAS_2 and UAS_3 rejects the INVITE by sending a 4xx error response. When P1 receives the 4xx responses it immediately sends 199 Early Dialog Terminated responses, associated with the dialogs where the 4xx responses were received, towards the UAC.

```
UAC             P1             UAS_2     UAS_3     UAS_4
  --- INVITE ----->
  --- INVITE (leg 2) -->
  --- INVITE (leg 3) --------->
  --- INVITE (leg 4) -------------->
  <--- 18x (leg 2) ------
    <--- 18x (leg 3) ---------
    <--- 18x (leg 4) --------------
    <--- 18x (leg 4) ---------------
      <--- 4xx (leg 2) ----- 
      --- ACK (leg 2) ---->
    <--- 199 (leg 2) --
      <--- 4xx (leg 3) ---------- 
      --- ACK (leg 3) ---------->
    <--- 199 (leg 3) --
      <--- 200 (leg 4) ---------------
    <--- 200 (leg 4) --
      --- ACK (leg 4) --->
                    --- ACK (leg 4) -------------->
```

Figure 1: Example call flow
12. Security Considerations

TBD

13. IANA Considerations

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

13.1. IANA Registration of the 199 response code

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

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.

14. Acknowledgements

Thanks to Paul Kyzivat, Dale Worley, Gilad Shaham, Francois Audet, Attila Sipos, Robert Sparks, Brett Tate and Ian Elz for their feedback and suggestions.

15. Normative References


[RFC3420] Sparks, R., "Internet Media Type message/sipfrag", RFC 3420, November 2002.


Author’s Address

Christer Holmberg
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

Email: christer.holmberg@ericsson.com