Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: March 21, 2019

P. Dawes
Vodafone Group
C. Arunachalam
Cisco Systems

September 17, 2018

# Marking SIP Messages to be Logged
# draft-ietf-insipid-logme-marking-13

Abstract

SIP networks use signaling monitoring tools to diagnose user-reported problems and for regression testing if network or user agent software is upgraded. As networks grow and become interconnected, including connection via transit networks, it becomes impractical to predict the path that SIP signaling will take between user agents, and therefore impractical to monitor SIP signaling end-to-end.

This document describes an indicator for the SIP protocol which can be used to mark signaling as being of interest to logging. Such marking will typically be applied as part of network testing controlled by the network operator and not used in normal user agent signaling. Operators of all networks on the signaling path can agree to carry such marking end-to-end, including the originating and terminating SIP user agents, even if a session originates and terminates in different networks.

Table of Contents

## 1. Introduction

When users experience problems with setting up sessions using SIP, enterprise or service provider network operators often have to identify the root cause by examining the SIP signaling. Also, when network or user agent software or hardware is upgraded, regression testing is needed. Such diagnostics apply to a small proportion of network traffic and can apply end-to-end, even if signaling crosses several networks possibly belonging to several different network operators. It may not be possible to predict the path through those networks in advance, therefore a mechanism is needed to mark a session as being of interest so that SIP entities along the signaling path can provide diagnostic logging. [RFC8123] illustrates this motivating scenario. This document describes a solution that meets the requirements for such "log me" marking of SIP signaling also defined in [RFC8123].

This document defines a new header field parameter "logme" for the "Session-ID" header field [RFC7989]. Implementations of this document MUST implement [RFC7989].

## 2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

## 3. "Log Me" Marking Protocol Aspects

### 3.1. Session-ID logme Parameter

Logging for diagnostic purposes is most effective when it is applied end-to-end in a communication session. This ability requires a "log me" marker to be passed through SIP intermediaries. The Session-ID header field defined in [RFC7989] was chosen to carry the "log me" marker as a "logme" parameter since the session identifier is typically passed through SIP B2BUAs (described in [RFC7092]) or other intermediaries, as per the Session-ID requirement REQ3 in [RFC7206]. The "logme" parameter shown in Figure 1 does not introduce any device-specific or user-specific information and MUST be passed unchanged with the Session-ID header field except for the cases specified in Section 3.4.2 where the "log me" marker may be removed at a network boundary.


Figure 1: "Log Me" marking using the "logme" Session-ID header field parameter

### 3.2. Starting and Stopping Logging

If a dialog is to be "log me" marked then marking MUST start with the SIP request that initiates that dialog (dialog initiating requests are described in Section 12.1 of [RFC3261]). For most effective

testing and troubleshooting, marking continues for the lifetime of the dialog, applies to each request and response in that dialog, and applies uninterrupted end-to-end including user devices. The "log me" marking mechanism described in this document allows for parts of the signaling path to not be marked, for example because an endpoint does not support the "log me" marking mechanism (Section 4.5.2) or because an endpoint or intermediary deliberately removes the "log me" marker (see Section 4.5.2.4). Also, marking errors can terminate marking before the dialog ends (see Section 5.3).

A user agent or intermediary adds a "log me" marker in an unmarked request or response in two cases: firstly because it is configured to add the marking to a dialog-creating request, or secondly because it has received a dialog-creating request that is being "log me" marked, causing it to maintain state to ensure that all requests and responses in the dialog are similarly "log me" marked. Once the "log me" marking is started for a dialog, all subsequent requests and responses in this dialog are "log me" marked and marking is stopped when this dialog and its related dialogs end. It is considered an error (see Section 5.1.2) if "log me" marking is started in a mid-dialog request or response.

For the first case, "log me" marking trigger condition configurations that define whether a user agent or intermediary can initiate "log me" marking for a given dialog are out of scope of this document. As an example of trigger condition configurations, the user agent or intermediary could be configured to add a "log me" marker for all dialogs initiated during a specific time period (e.g., 9:00 am - 10:00 am every day), for specific dialogs that have a particular "User-Agent" header field value, or for a specific set of called party numbers for which users are experiencing call setup failures.

For the second case of a user agent or intermediary detecting that a dialog-initiating request is being "log me" marked, the scope of such marking extends to the lifetime of the dialog. In addition, as discussed in Section 3.7, "log me" marked dialogs that create related dialogs (e.g. REFER) may transfer the marking to the related dialogs. In such cases, the entire "session", identified by the Session-ID header field, is "log me" marked.

## 3.3. Identifying Test Cases

The local Universally Unique Identifier (UUID) portion of Session-ID [RFC7989] in the initial SIP request of a dialog is used as a random test case identifier (described in REQ 5 in [RFC8123]). This provides the ability to collate all logged SIP requests and responses to the initial SIP request in a dialog or standalone transaction.

### 3.4. Passing the Marker

#### 3.4.1. To and From a User Device

When a user device inserts the "log me" marker, the marker MUST be passed unchanged in the Session-ID header field across an edge proxy or a B2BUA adjacent to the user device.

#### 3.4.2. To and From an External Network

An external network is a peer network connected at a network boundary as defined in [RFC8123].

External networks may be connected directly or via a peering network and such networks often have specific connection agreements. Whether "log me" marking is removed depends upon the policy applied at the network to network interface. Troubleshooting and testing will be easier if peer networks endeavor to make agreements to pass "log me" marking unchanged. However, since a "log me" marker may cause a SIP entity to log the SIP header and body of a request or response, if no agreement exists between peer networks then the "log me" marker MUST be removed at a network boundary.

#### 3.4.3. Across a Non-Supporting SIP Intermediary

"Log me" marking is most effective if passed end-to-end. However, intermediaries that do not comply with this document might pass the "log me" marker unchanged or drop it entirely.

### 3.5. Logging Multiple Simultaneous Dialogs

An originating or terminating user agent and SIP entities on the signaling path can log multiple SIP dialogs simultaneously. These dialogs are differentiated by their test case identifier (the local UUID of the Session-ID header field at the originating device).

### 3.6. Format of Logged Signaling

The entire SIP message (SIP request line, response line, header fields and message body) SHOULD be logged since troubleshooting might be difficult if information is missing. Logging SHOULD use common standard formats such as the SIP CLF defined in [RFC6873] and Libpcap [application/vnd.tcpdump.pcap]. If SIP CLF format is used, the entire message is logged using Vendor-ID = 00000000 and Tag = 02 in the <OptionalFields> portion of the SIP CLF record (see [RFC6873] section 4.4). Header fields SHOULD be logged in the form in which they appear in the message, they SHOULD NOT be converted between long and compact forms described in [RFC3261] section 7.3.3.

### 3.7. Marking Related Dialogs

"Log me" marking is done per-dialog and typically begins at dialog creation and ends when the dialog ends. However, dialogs related to a "log me" marked dialog MAY also be "log me" marked for call control features such as call forward, transfer, park, and join. As described in [RFC7989] section 6, related dialogs can occur when an endpoint receives a 3xx message, a REFER that directs the endpoint to a different peer, an INVITE request with Replaces that also potentially results in communicating with a new peer, or an INVITE with a Join header field as described in [RFC3911]. An example is call transfer described in section 6.1 of [RFC5589] and the logged signaling for related dialogs can be correlated using Session-ID values as described in section 10.9 of [RFC7989].

In the example shown in Figure 2, Alice has reported problems making call transfers. Her terminal is placed in debug mode in preparation to log marked signaling from the network administrator Bob. Bob's terminal is configured to "log me" mark and log signaling for calls originated during the troubleshooting session (e.g. for a duration of 15 minutes). Bob, who is troubleshooting the problem, arranges to make a call that Alice can attempt to transfer. Bob calls Alice, which creates initial dialog1, and then Alice transfers the call to connect Bob to Carol. Logged signaling is correlated using the test case identifier, which is the local UUID ab30317f1a784dc48ff824d0d3715d86 in the Session-ID header field of INVITE request F1. Logging by Alice's terminal begins when it receives and echoes the "log me" marker in INVITE F1 and ends when the last request or response in the dialog is sent or received (200 OK F7 of dialog1). Also during dialog1, Alice's terminal logs related REFER dialog2 that it initiates and terminates as part of the call transfer. Alice's terminal inserts a "log me" marker in the REFER request and 200 OK responses to NOTIFY requests in dialog2. Both dialog1 and dialog2 have the same test case identifier.

Logging by Bob's terminal begins when it sends INVITE F1, which includes the "log me" marker, and ends when dialog3, initiated by Bob, ends. Logging by Carol's terminal begins when it receives the INVITE F5 with the "log me" marker and ends when dialog3 ends.

dialog3 is not logged by Alice's terminal, however the test case identifier ab30317f1a784dc48ff824d0d3715d86 is also the test case identifier (local-uuid) in INVITE F5. Also, the test case identifier of dialog2, which is logged by Alice's terminal, can be linked to dialog1 and dialog3 because the remote-uuid component of dialog2 is the test case identifier ab30317f1a784dc48ff824d0d3715d86.

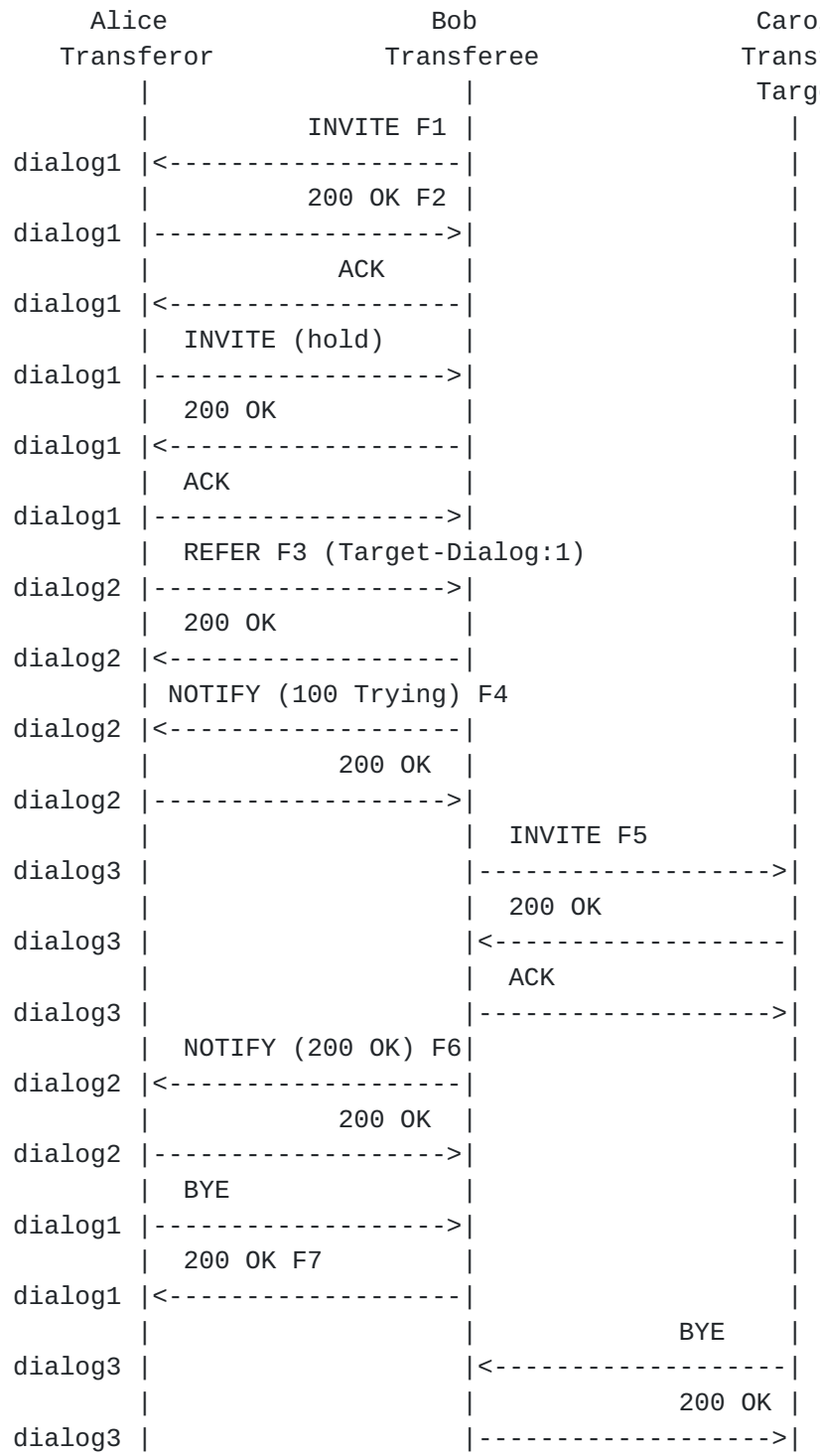



Figure 2: "Log me" marking related dialogs in call transfer

F1 - Bob's UA inserts the "logme" parameter in the Session-ID header field of the INVITE request that creates dialog1.

F3 - Alice's UA inserts the "logme" parameter in the Session-ID header field of the REFER request that creates dialog2 which is related to dialog1.

F5 - Bob's UA inserts the "logme" parameter in the Session-ID header field of the INVITE request that creates dialog3 which is related to dialog1.

F1 INVITE Transferee -> Transferor

```
INVITE sips:transferor@atlanta.example.com SIP/2.0
Via: SIP/2.0/TLS [2001:db8::1];branch=z9hG4bKnas432
Max-Forwards: 70
To: <sips:transferor@atlanta.example.com>
From: <sips:transferee@biloxi.example.com>;tag=7553452
Call-ID: 090459243588173445
Session-ID: ab30317f1a784dc48ff824d0d3715d86
   ;remote=00000000000000000000000000000000;logme
CSeq: 29887 INVITE
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, NOTIFY
Supported: replaces, gruu, tdialog
Contact: <sips:3ld812adkjw@biloxi.example.com;gr=3413kj2ha>
Content-Type: application/sdp
Content-Length: ...
```

F2 200 OK Transferor -> Transferee

```
SIP/2.0 200 OK
Via: SIP/2.0/TLS [2001:db8::1];branch=z9hG4bKnas432
To: <sips:transferor@atlanta.example.com>;tag=31kdl4i3k
From: <sips:transferee@biloxi.example.com>;tag=7553452
Call-ID: 090459243588173445
Session-ID: 47755a9de7794ba387653f2099600ef2
   ;remote=ab30317f1a784dc48ff824d0d3715d86;logme
CSeq: 29887 INVITE
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, NOTIFY
Supported: replaces, gruu, tdialog
Contact: <sips:4889445d8kjtk3@atlanta.example.com;gr=723jd2d>
Content-Type: application/sdp
Content-Length: ...
```

F3 REFER Transferor -> Transferee

```
REFER sips:3ld812adkjw@biloxi.example.com;gr=3413kj2ha SIP/2.0
Via: SIP/2.0/TLS pc33.atlanta.example.com;branch=z9hG4bKna9
```

```
Max-Forwards: 70
To: <sips:3ld812adkjw@biloxi.example.com;gr=3413kj2ha>
From: <sips:transferor@atlanta.example.com>;tag=1928301774
Call-ID: a84b4c76e66710
Session-ID: 47755a9de7794ba387653f2099600ef2
   ;remote=ab30317f1a784dc48ff824d0d3715d86;logme
CSeq: 314159 REFER
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, NOTIFY
Supported: gruu, replaces, tdialog
Require: tdialog
Refer-To: <sips:transfertarget@chicago.example.com>
Target-Dialog: 090459243588173445;local-tag=7553452
 ;remote-tag=31kdl4i3k
Contact: <sips:4889445d8kjtk3@atlanta.example.com;gr=723jd2d>
Content-Length: 0
```

F4 NOTIFY Transferee -> Transferor

```
NOTIFY sips:4889445d8kjtk3@atlanta.example.com
   ;gr=723jd2d SIP/2.0
Via: SIP/2.0/TLS [2001:db8::1];branch=z9hG4bKnas432
Max-Forwards: 70
To: <sips:transferor@atlanta.example.com>;tag=1928301774
From: <sips:3ld812adkjw@biloxi.example.com;gr=3413kj2ha>
 ;tag=a6c85cf
Call-ID: a84b4c76e66710
Session-ID: ab30317f1a784dc48ff824d0d3715d86
   ;remote=47755a9de7794ba387653f2099600ef2;logme
CSeq: 73 NOTIFY
Contact: <sips:3ld812adkjw@biloxi.example.com;gr=3413kj2ha>
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, NOTIFY
Supported: replaces, tdialog
Event: refer
Subscription-State: active;expires=60
Content-Type: message/sipfrag
Content-Length: ...
```

F5 INVITE Transferee -> Transfer Target

```
INVITE sips:transfertarget@chicago.example.com SIP/2.0
Via: SIP/2.0/TLS [2001:db8::1];branch=z9hG4bKnas41234
Max-Forwards: 70
To: <sips:transfertarget@chicago.example.com>
From: <sips:transferee@biloxi.example.com>;tag=j3kso3iqhq
Call-ID: 90422f3sd23m4g56832034
Session-ID: ab30317f1a784dc48ff824d0d3715d86
```

```
   ;remote=00000000000000000000000000000000;logme
CSeq: 521 REFER
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, NOTIFY
Supported: replaces, gruu, tdialog
Contact: <sips:3ld812adkjw@biloxi.example.com;gr=3413kj2ha>
Content-Type: application/sdp
Content-Length: ...


F6 NOTIFY Transferee -> Transferor

NOTIFY sips:4889445d8kjtk3@atlanta.example.com
   ;gr=723jd2d SIP/2.0
Via: SIP/2.0/TLS [2001:db8::1];branch=z9hG4bKnas432
Max-Forwards: 70
To: <sips:transferor@atlanta.example.com>;tag=1928301774
From: <sips:3ld812adkjw@biloxi.example.com;gr=3413kj2ha>
 ;tag=a6c85cf
Call-ID: a84b4c76e66710
Session-ID: ab30317f1a784dc48ff824d0d3715d86
   ;remote=47755a9de7794ba387653f2099600ef2;logme
CSeq: 74 NOTIFY
Contact: <sips:3ld812adkjw@biloxi.example.com;gr=3413kj2ha>
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, NOTIFY
Supported: replaces, tdialog
Event: refer
Subscription-State: terminated;reason=noresource
Content-Type: message/sipfrag
Content-Length: ...
```

### 3.8. Forked Requests

A SIP intermediary is required to copy the "log me" marker into forked requests. SIP request forking is discussed in sections 4 and 16.6 of [RFC3261].

## 4. SIP Entity Behavior

### 4.1. Scope of Marking

"Log me" marking is intended to be limited, in time period and number of dialogs marked, to the minimum needed to troubleshoot a particular problem or perform a particular test.

- SIP entities MUST be configured to "log me" mark only dialogs needed for the current testing purpose e.g. troubleshooting or regression testing. The mechanisms in this section ensure that "log me" marking begins at dialog creation and, other than cases of marking related dialogs or premature ending, ends when the dialog being "log me" marked ends.

- If a dialog is to be marked, the only way to initiate "log me" marking is at the dialog-creating request (e.g. SIP INVITE) sent by an originating endpoint or an intermediary that marks on behalf of the originating endpoint. Marking that appears mid-dialog is an error as described in Section 5.1.2. The final terminating endpoint or an intermediary that marks on behalf of the terminating endpoint cannot initiate marking but takes action as defined in Section 4.2 and Section 4.3 if it receives an incoming "log me" marker.

Note that the error cases described in sections 5.1 and 5.2 cause SIP entities to stop "log me" marking, and the requirements in Section 7 also place requirements on SIP entities, including allowing SIP entities to not log signaling based on local policies (see Section 8.6).

## 4.2. Endpoints

A common scenario is to have both originating and terminating endpoints support "log me" marking with the originating endpoint configured to initiate "log me" marking. In this simplest use case, the originating user agent inserts a "log me" marker in the dialog-creating SIP request and all subsequent SIP requests within that dialog. The "log me" marker is passed through the SIP intermediaries and arrives at the terminating user agent which echoes the "log me" marker in the corresponding responses. If the terminating user agent sends an in-dialog request on a dialog that is being "log me" marked, it inserts a "log me" marker and the originating user agent echoes the "log me" marker in responses. The terminating user agent logs the "log me" marked SIP requests and responses if it is allowed as per policy defined in the terminating network. This basic use case suggests the following rules for originating and terminating user agents.

For originating user agents:

- The originating user agent configured for "log me" marking MUST insert a "log me" marker into the dialog-creating SIP request and subsequent in-dialog SIP requests.

- The originating user agent itself logs marked requests and responses.
- The originating user agent echoes, in responses, the "log me" marker received in in-dialog requests from the terminating side.
- The originating user agent logs the SIP responses that it sends in response to received "log me" marked in-dialog requests.
- The originating user agent MAY also apply these rules to any subsequent related SIP dialogs as described in Section 3.7.

For terminating user agents:

- The terminating user agent detects that a dialog is of interest to logging by the existence of a "log me" marker in an incoming dialog-creating SIP request.
- The terminating user agent itself logs marked requests and corresponding marked responses if allowed as per policy.
- The terminating user agent MUST echo a "log me" marker in responses to a SIP request that included a "log me" marker.
- If the terminating user agent has detected that a dialog is being "log me" marked, it MUST insert a "log me" marker in any in-dialog SIP requests that it sends.
- The terminating user agent itself logs any in-dialog SIP requests that it sends if allowed as per policy.
- The terminating user agent MAY also apply these rules to any subsequent related SIP dialogs as described in Section 3.7.

### 4.3. SIP Intermediaries Acting on Behalf of Endpoints

A network operator may know that some of the user agents connected to the network do not support "log me" marking. Subject to the authorizations in Section 7.1, a SIP intermediary close to the user agent (e.g. edge proxy, B2BUA) on the originating and/or terminating sides inserts the "log me" marker instead in order to test sessions involving such user agents.

The originating and terminating SIP intermediaries are not identified by protocol means but are designated and explicitly configured by the network administrator to "log me" mark on behalf of endpoints. The intermediaries that are known to be closest to the terminals can be configured to "log me" mark on behalf of terminals that do not

support "log me" marking. The originating SIP intermediary is the first one to be traversed by a SIP request sent by the originating endpoint. Similarly, the terminating SIP intermediary is last intermediary traversed before the terminating endpoint is reached.

The SIP intermediary at the originating side is configured to insert the "log me" marker on behalf of the originating endpoint. If the terminating user agent does not echo the "log me" marker in responses to a marked request then the SIP intermediary closest to the terminating user agent, if configured to mark on behalf of the terminating user agent, inserts a "log me" marker in responses to the request. Likewise, if the terminating user agent sends an in-dialog request, the SIP intermediary at the terminating side inserts a "log me" marker and the SIP intermediary at the originating side echoes the "log me" marker in responses to that request. Originating and terminating intermediaries that are configured for "log me" marking on behalf of the endpoint must also mark dialog-creating requests that contain Target-Dialog [RFC4538], Join [RFC3911] and Replaces [RFC3891] header fields and corresponding responses. The SIP intermediaries at the originating and terminating sides log the "log me" marked SIP requests and responses if it is allowed as per policy defined in the originating and terminating networks. This scenario suggests the following rules when a SIP intermediary is configured to initiate or handle "log me" marking on behalf of a user agent.

For the originating SIP intermediary:

- The originating SIP intermediary configured for "log me" marking MUST insert a "log me" marker into the dialog-creating SIP request and subsequent in-dialog SIP requests.

- The originating SIP intermediary itself logs marked requests and responses.

- The originating SIP intermediary detects the "log me" marker received in in-dialog requests and echoes the "log me" marker in the corresponding SIP responses.

- The originating SIP intermediary logs the SIP responses that it sends in response to "log me" marked in-dialog requests.

- The originating SIP intermediary MAY also apply these rules to any subsequent related SIP dialogs as described in Section 3.7).

For the terminating SIP intermediary:

- The terminating SIP intermediary detects that a dialog is of interest to logging by the existence of a "log me" marker in an incoming dialog-creating SIP request.

- The terminating SIP intermediary itself logs marked requests and corresponding responses if allowed as per policy.

- The terminating SIP intermediary MUST echo a "log me" marker in responses to a SIP request that included a "log me" marker.

- If terminating SIP intermediary has detected that a dialog is being "log me" marked, it MUST insert a "log me" marker in any in-dialog SIP requests from the terminating user agent.

- The terminating SIP intermediary itself logs any in-dialog SIP requests that it sends if allowed as per policy.

- The terminating SIP intermediary MAY also apply these rules to any subsequent related SIP dialogs as described in Section 3.7.

### 4.4. B2BUAs

B2BUA "log me" behavior is specified based on its different signaling plane roles described in [RFC7092].

A Proxy-B2BUA SHOULD copy "log me" marking in requests and responses from its terminating to the originating side without needing explicit configuration to do so.

A dialog on one "side" of the B2BUA may or may not be coupled to a related dialog on the other "side" for "log me" purposes. To allow end-to-end troubleshooting of user problems and regression testing, a signaling-only and SDP-modifying signaling-only B2BUA [RFC7092] SHOULD couple related dialogs for "log me" marking purposes and pass on the received "log me" parameter from the originating side to terminating side and vice versa. For example, a SIP B2BUA handling an end-to-end session between an external caller and an agent in a contact center environment can couple the dialog between itself and an agent with the dialog between itself and external caller and pass on the "log me" marking from originating side to terminating side to enable end-to-end logging of specific sessions of interest.

For dialogs that are being "log me" marked, all B2BUAs MUST "log me" mark in-dialog SIP requests that they generate on their own, without needing explicit configuration to do so. This rule applies to both the originating and terminating sides of a B2BUA.

## 4.5. "Log me" Marker Processing by SIP Intermediaries

### 4.5.1. Stateless processing

Typically, "log me" marking will be done by an originating UA and echoed by a terminating UA. SIP intermediaries on the signaling path between these UAs that do not perform the tasks described in Section 4.3 or Section 4.4 MUST simply log any request or response that contains a "log me" marker in a stateless manner, if it is allowed per local policy.

### 4.5.2. Stateful processing

The originating and terminating SIP intermediaries that "log me" mark on behalf of endpoints and SIP intermediaries that remove "log me" marking at the network boundary must maintain state to enable "log me" marking. Applicable scenarios are as follows.

- The originating UA does not support "log me" marking. This scenario was described in Section 4.3 and requires support by the originating SIP intermediary. "Log me" marker processing is illustrated in Section 4.5.2.1.

- The terminating UA does not support "log me" marking. This scenario was described in Section 4.3 and requires support by the terminating SIP intermediary. "Log me" marker processing is illustrated in Section 4.5.2.2.

- The originating network ensures that it does not pass marking outside its boundaries in order to not impact any external networks. The originating network removes "log me" marking from SIP requests and responses before forwarding them from its network boundary to external networks but adds marking back to any incoming SIP requests and responses belonging to any "log me" marked dialog. This scenario requires support by the SIP intermediary at the originating network boundary and "log me" marker processing is illustrated in Section 4.5.2.3.

- The terminating network ensures that it does not allow "log me" marking from external networks to pass through its boundary to its internal entities. The terminating network removes "log me" marking from SIP requests and responses before forwarding them internally but adds marking back to any outgoing SIP requests and responses belonging to any "log me" marked dialog. This scenario requires support by the SIP intermediary at the terminating network boundary and "log me" marker processing is illustrated in Section 4.5.2.4.

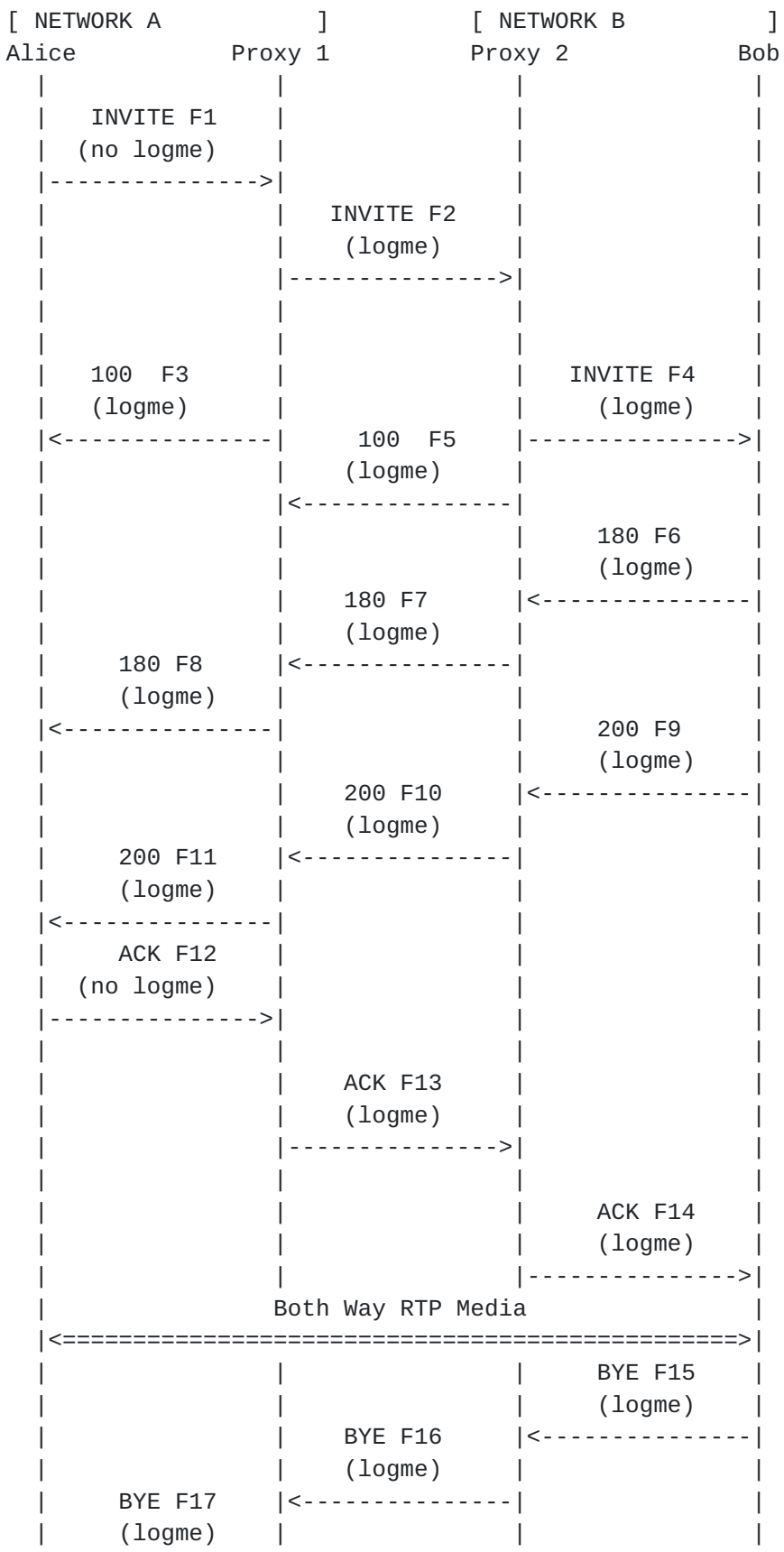
- The terminating network does not support "log me" marking and does not echo marking that it receives. The originating network adds marking back to any incoming SIP requests and responses belonging to the "log me" marked dialog. This scenario requires support by the SIP intermediary at the originating network boundary and "log me" marker processing is illustrated in Section 4.5.2.5.

SIP intermediary behavior in these scenarios is illustrated using [RFC3665] example call flow "Session Establishment Through Two Proxies".

#### 4.5.2.1. "Log Me" marking not supported by Originating UA

Alice's user agent does not support "log me" marking and hence Proxy 1, which is the SIP intermediary closest to Alice, is configured to act on behalf of Alice's user agent to "log me" mark specific dialogs of interest that are created by Alice for troubleshooting purposes.

In Figure 3 below, Proxy 1 in the originating network maintains state of which dialogs are being logged in order to "log me" mark all SIP requests and responses that it receives from Alice's user agent before forwarding them to Proxy 2.


[ NETWORK A ]
[ NETWORK B ]
Alice
Proxy 1
Proxy 2
Bob
INVITE F1
(no logme)
INVITE F2
(logme)
100 F3
(logme)
INVITE F4
(logme)
100 F5
(logme)
180 F6
(logme)
180 F7
(logme)
180 F8
(logme)
200 F9
(logme)
200 F10
(logme)
200 F11
(logme)
ACK F12
(no logme)
ACK F13
(logme)
ACK F14
(logme)
Both Way RTP Media
BYE F15
(logme)
BYE F16
(logme)
BYE F17
(logme)

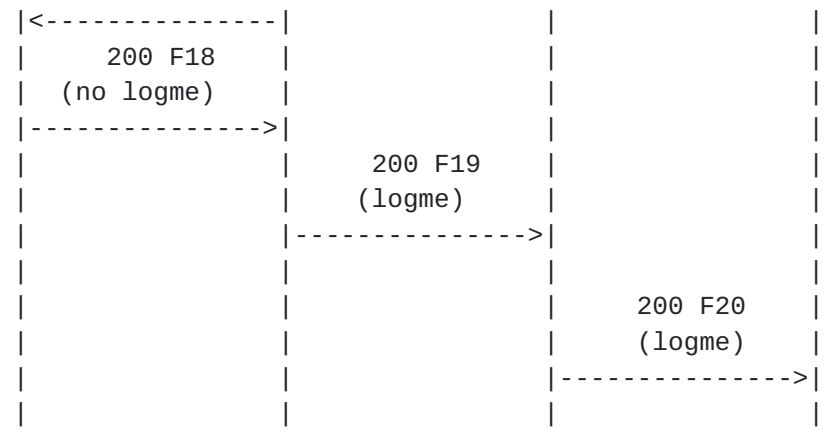



Figure 3: The originating UA does not support "log me" marking

F1 - Alice's UA does not insert a "log me" marker in the dialog-creating INVITE request F1. Nevertheless, Proxy 1 is configured to initiate logging on behalf of Alice. Proxy 1 logs INVITE request F1 and maintains state that this dialog is being logged.

F2 - Proxy 1 inserts a "log me" marker in INVITE request F2 before forwarding it to Proxy 2 and also logs this request.

F3 - Proxy 1 inserts a "log me" marker in 100 response F3 before forwarding it to Alice's UA since this is a response sent on a dialog that is being "log me" marked and also logs this response.

F4 - Bob's UA detects the "log me" marker and logs the INVITE request F4 if allowed as per policy.

F6 - Bob's UA echoes the "log me" marker in INVITE request F4 into 180 response F6. It logs this response if allowed as per policy.

F7 and F8 - Proxy 1 logs the received the "180" response F7 and passes the "log me" marker to Alice's UA in F8.

F12 - Proxy 1 receives ACK with no "log me" marker. It doesn't consider this as an error since it is configured to "log me" mark on behalf of Alice's UA.

F13 - Proxy 1 inserts a "log me" marker in ACK request F13 before forwarding it to Proxy 2 and also logs this request.

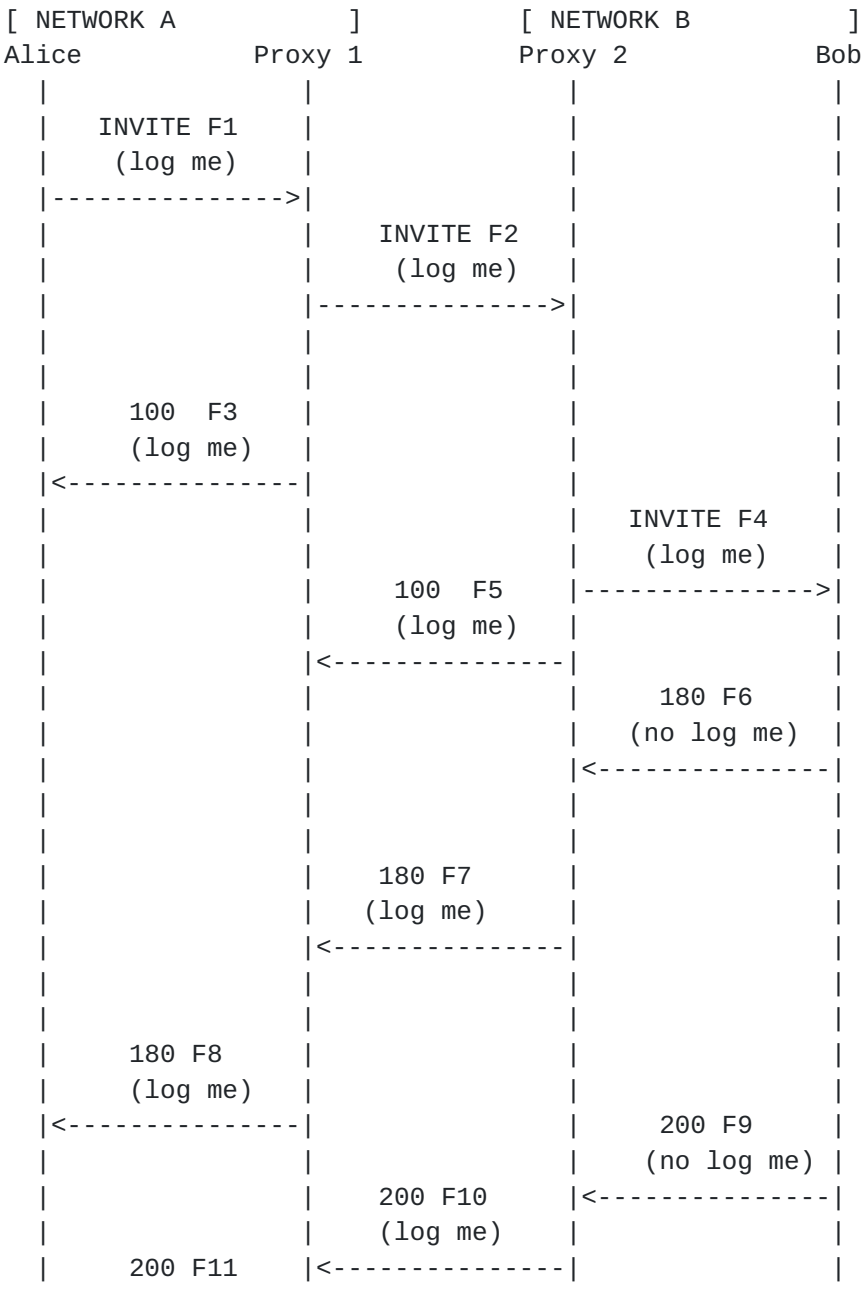
F15 - Bob's UA inserts a "log me" marker in the in-dialog BYE request and this "log me" marker is carried back to Alice's UA in F16 and F17. Bob's UA logs this request if allowed as per policy.

F18 - Alice's UA does not echo the "log me" marker from BYE request F17 into 200 response F18.

F19 - Proxy 1 inserts a "log me" marker in 200 response F19 before forwarding it to Proxy 2 and also logs this response.

#### 4.5.2.2. "Log Me" marking not supported by Terminating UA

In Figure 4 below Bob's UA does not support "log me" marking, so Proxy 2 in the terminating network maintains state to ensure "log me" marking of SIP requests and responses from Bob's UA.



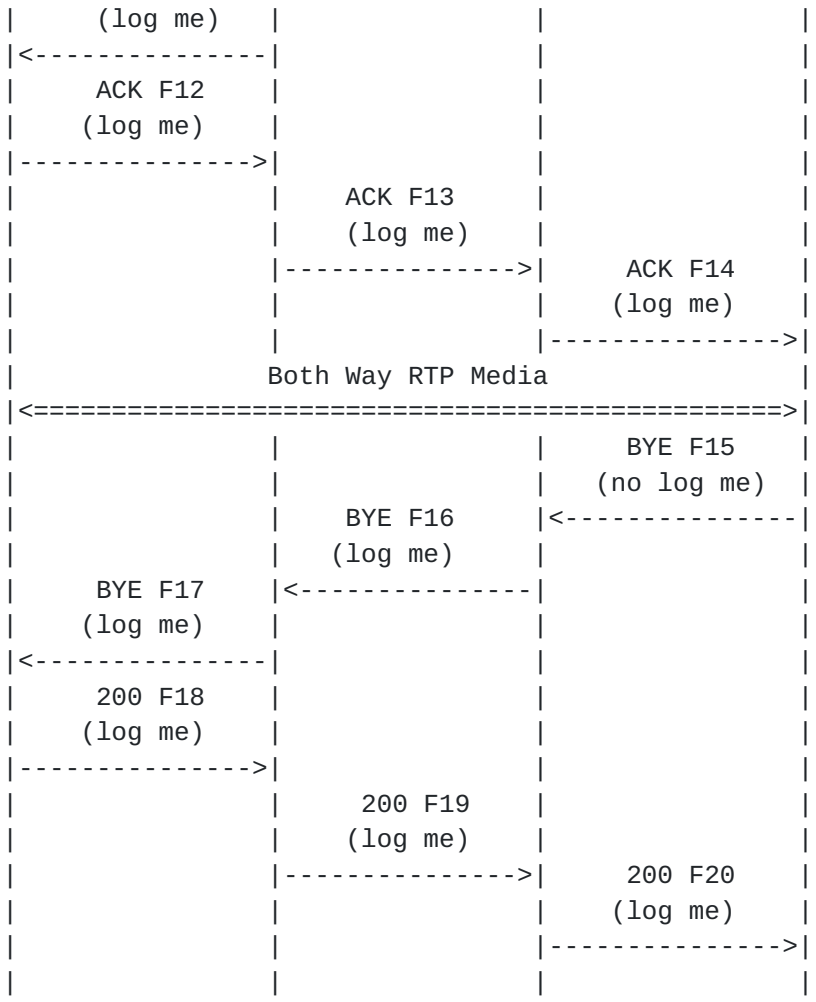



Figure 4: The terminating UA does not support "log me" marking.

F1 - Alice's UA inserts a "log me" marker in the dialog-creating INVITE request F1.

F2 - INVITE F2 is "log me" marked and Proxy 2 therefore maintains state that this dialog is to be logged. Proxy 2 logs the request and responses of this dialog if allowed per policy.

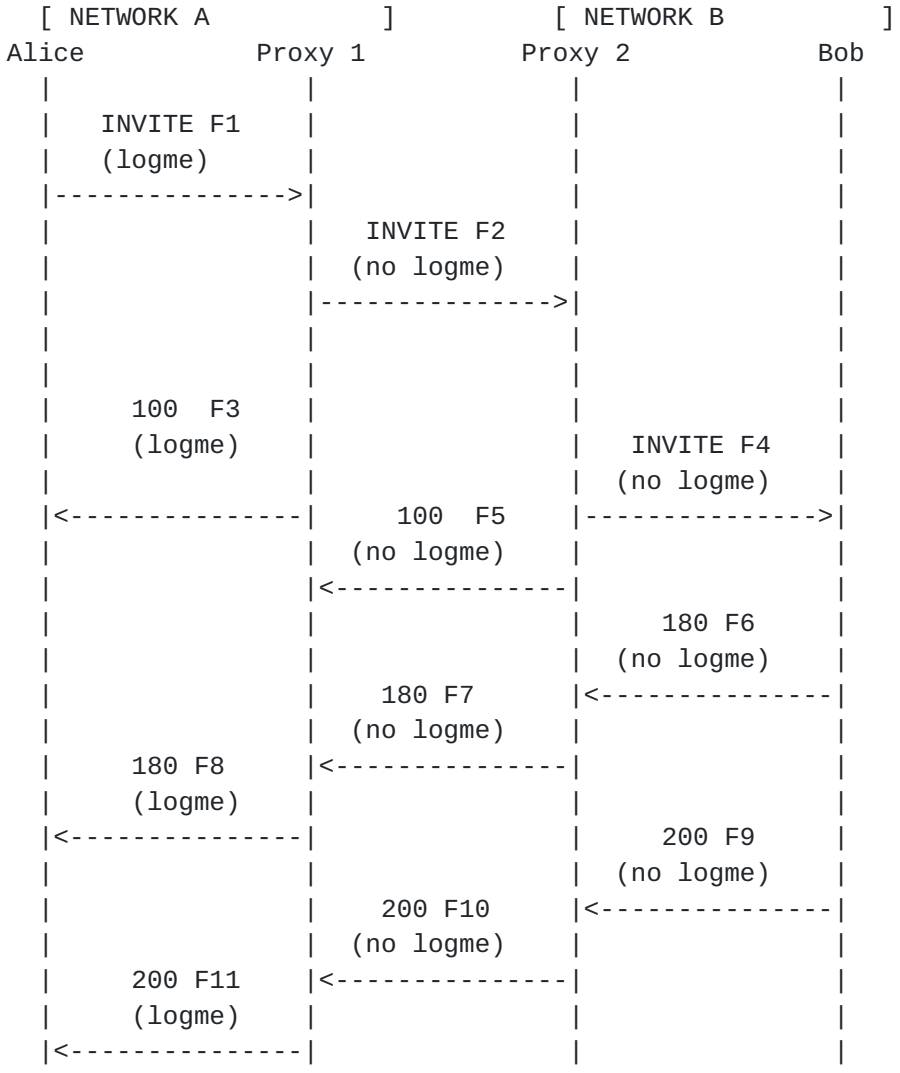
F5 - Proxy 2 inserts a "log me" marker in the 100 response it sends to Proxy 1.

F6 - Bob's UA does not support "log me" marking, therefore the 180 response to the INVITE request doesn't have a "log me" marker.

F7 - Proxy 2 inserts a "log me" marker in the 180 response on behalf of Bob's UA before forwarding it. The same applies to response F10 and the BYE request in F16.

#### 4.5.2.3. "Log Me" marking removed by Originating Network

If network A in Figure 5 below is performing testing independently of network B then network A removes "log me" marking from SIP requests and responses forwarded to network B to prevent triggering unintended logging in network B. Proxy 1 removes "log me" marking from requests and responses that it forwards to Proxy 2 and maintains state of which dialogs are being "log me" marked in order to "log me" mark requests and responses that it forwards from Proxy 2 to Alice's user agent. For troubleshooting purposes, Proxy 1 MAY also log the requests and responses sent to or received from Proxy 2 even though it removed "log me" marker prior to forwarding the messages to Proxy 2.



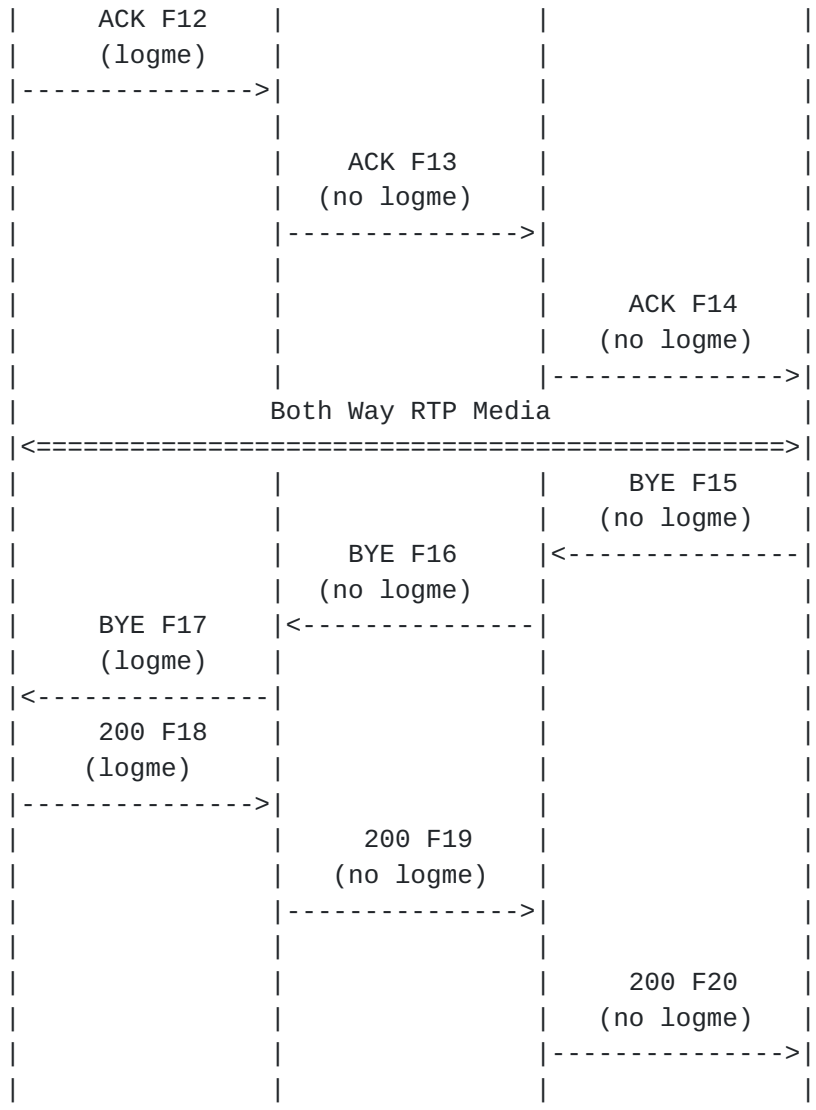



Figure 5: The originating network removes "log me" marking from outgoing SIP messages at its network edge.

F1 - Alice's UA inserts a "log me" marker in the dialog-creating INVITE request and Proxy 1 therefore maintains state that this dialog is to be logged.

F2 - Proxy 1 removes "log me" marking from INVITE request before forwarding it to Proxy 2. Proxy 1 logs INVITE request F2.

F3 - Proxy 1 inserts a "log me" marker in 100 response sent to Alice's user agent and logs this response.

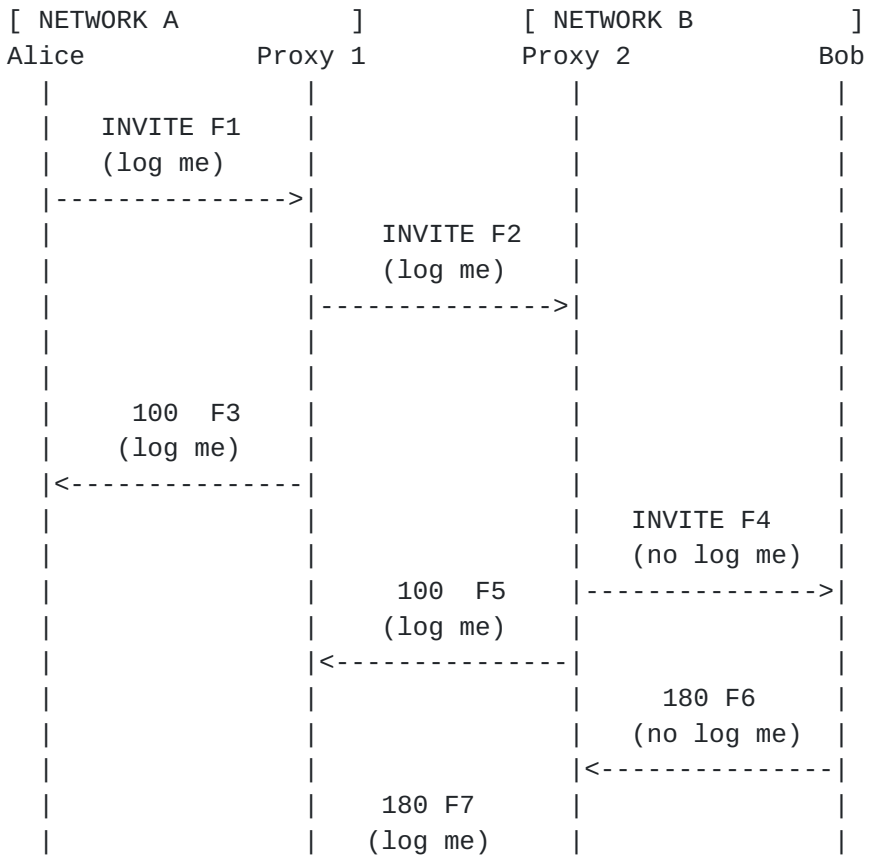
F8 - Proxy 1 inserts a "log me" marker in 180 response before forwarding it to Alice's user agent and logs this response. The same applies to responses F11, F17.

F13 - Proxy 1 removes "log me" marking from ACK request and logs this request before forwarding it to Proxy 2.

F19 - Proxy 1 removes "log me" marking from the 200 response of the BYE request and logs this response before forwarding it to Proxy 2.

#### 4.5.2.4. "Log Me" marking removed by Supporting Terminating Network

In Figure 6 below Proxy 2 removes "log me" marking from all SIP requests and responses entering network B. However, Proxy 2 supports maintaining the marking state of the dialog and "log me" marks requests and responses that it sends towards Proxy 1. For troubleshooting purposes, Proxy 2 MAY also log the requests and responses received from or sent to Bob even though it removed "log me" marker prior to forwarding the messages to Bob. This scenario might be used for troubleshooting a signaling path between two enterprise or carrier networks, or across a transit network, with minimal logging (i.e., only at the network boundaries).



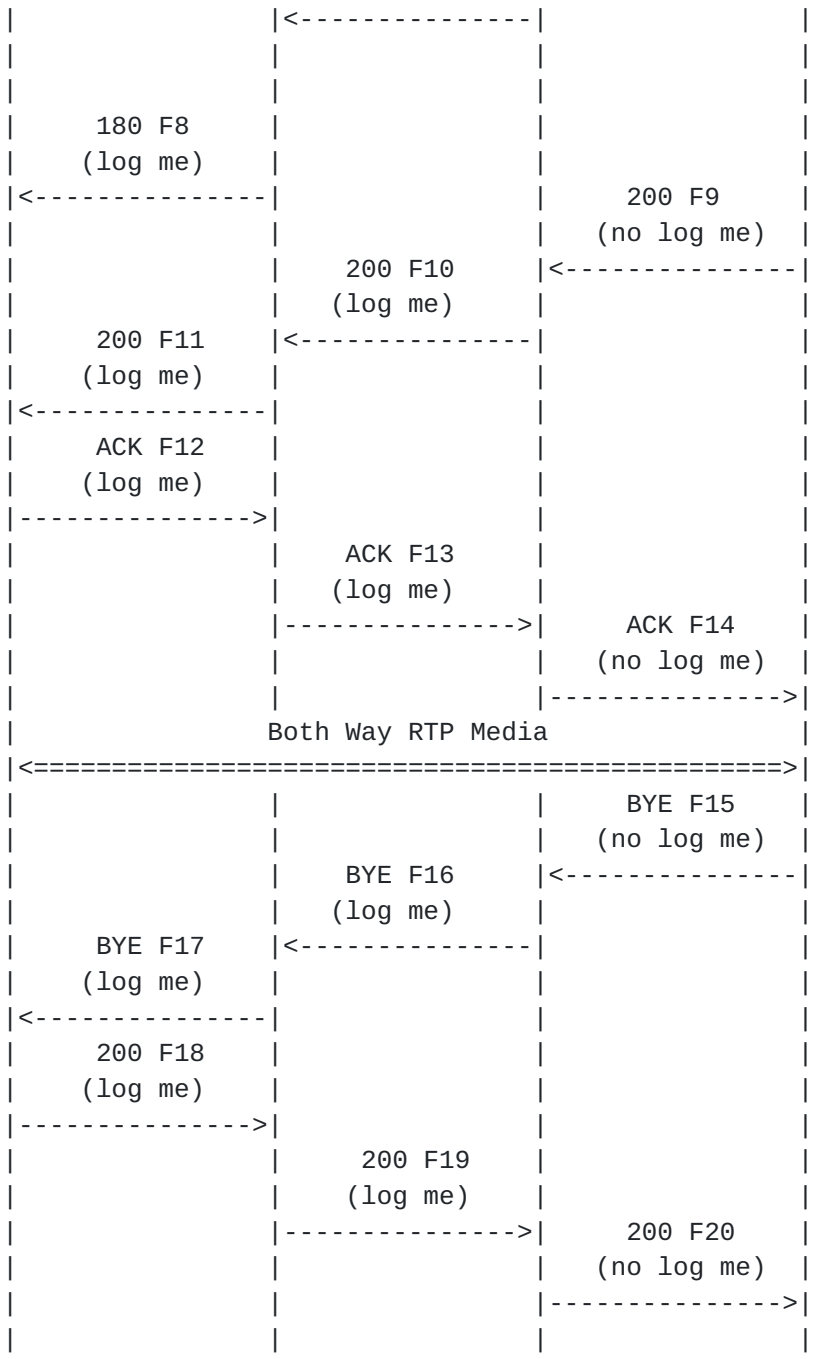



Figure 6: The terminating network removes "log me" marking from incoming SIP messages at its network edge.

F1 - Alice's UA inserts a "log me" marker in the dialog-creating INVITE request F1. Proxy 1 detects the "log me" marker, logs the request and maintains state that this dialog is to be logged.

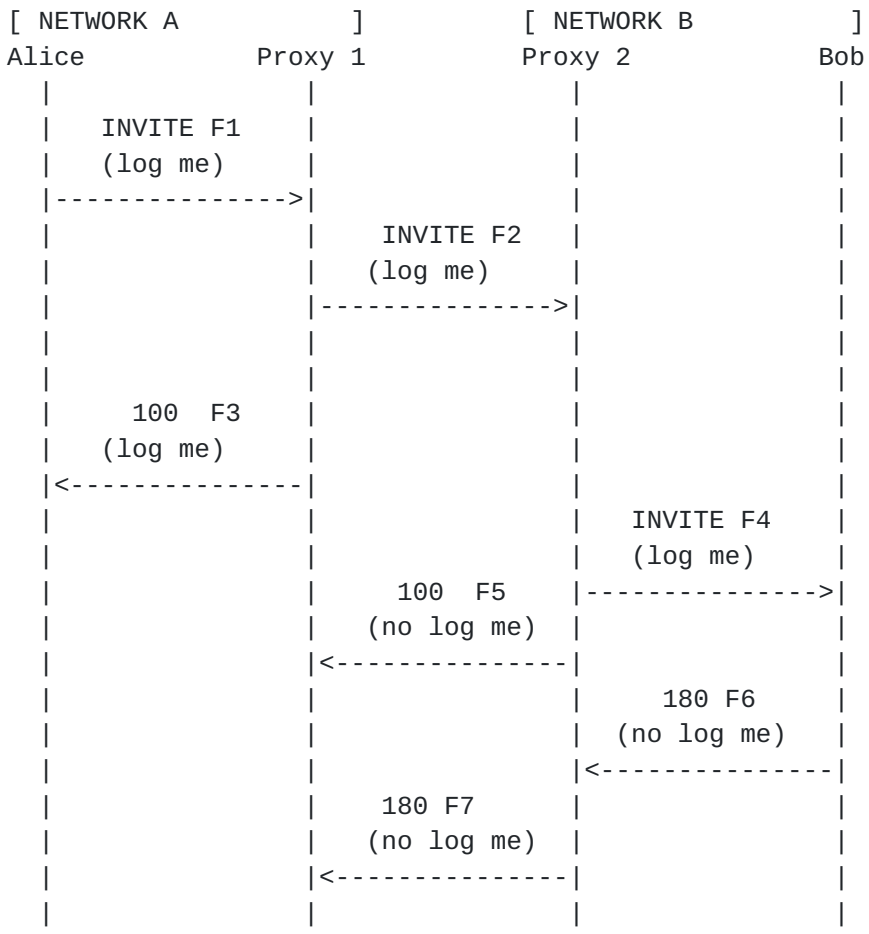
F2 - Proxy 2 removes "log me" marker in the INVITE request F2 before forwarding it as F4. The same applies to responses F13, F19.

F6 - Proxy 2 inserts a "log me" marker in 180 response to the INVITE request and logs the request before forwarding it as F7. The same applies to response F9 and the BYE request in F15.

#### 4.5.2.5. "Log Me" marking passed by Non-Supporting Terminating Network

In Figure 6 below Proxy 2 is not "log me" aware and therefore passes marking in all SIP requests and responses entering network B according to the rules in Section 16.6 and 16.7 of [RFC3261]. Proxy 2 does not log requests and responses in the dialog. Proxy 1 supports maintaining the marking state of the dialog. When Proxy 1 observes that requests and responses received from Proxy 2 are not marked it adds the marking.

For troubleshooting purposes, Proxy 1 MAY also log the requests and responses received from or sent to Proxy 2 even though Proxy 2 didn't add "log me" to messages sent to Proxy 1.



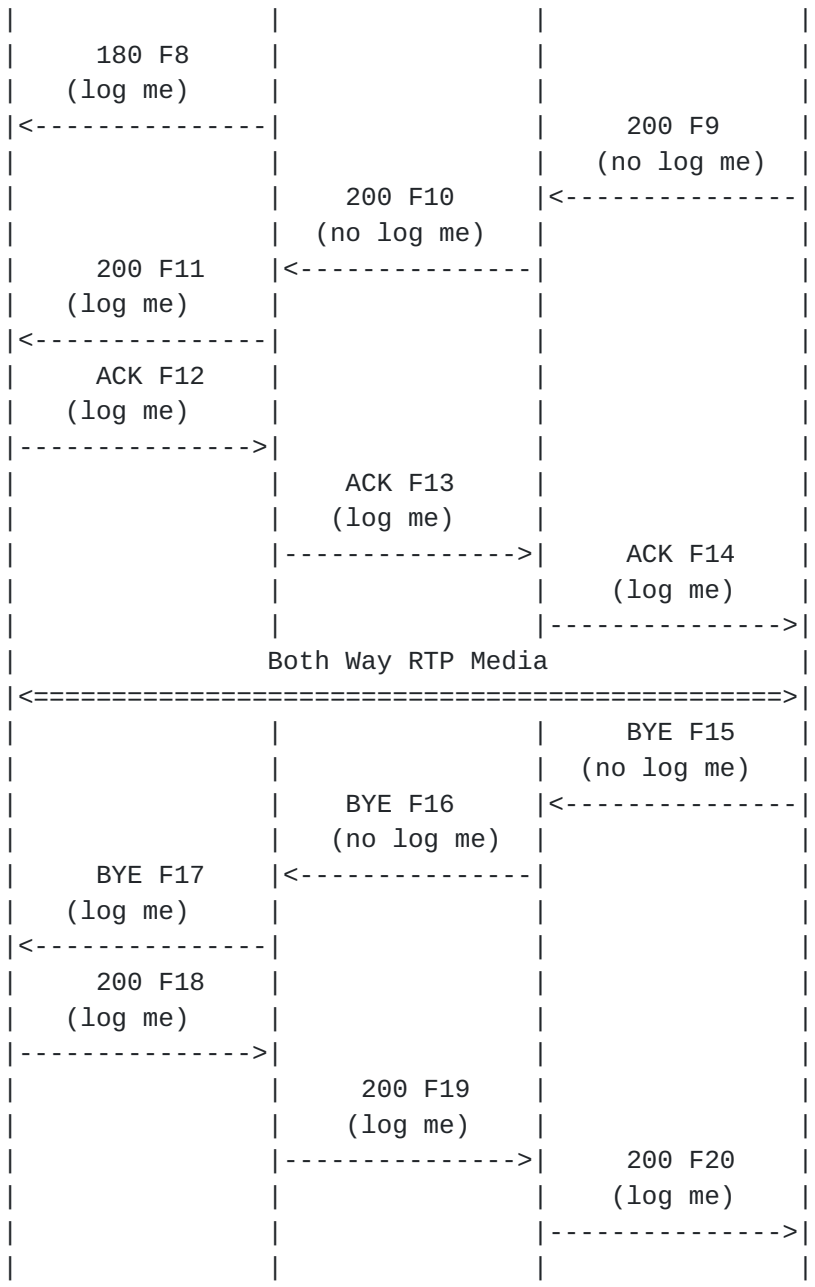



Figure 7: The terminating network removes "log me" marking from incoming SIP messages at its network edge.

F1 - Alice's UA inserts a "log me" marker in the dialog-creating INVITE request F1. Proxy 1 detects the "log me" marker, logs the request and maintains state that this dialog is to be logged.

F2 - Proxy 2 passes the "log me" marker in the INVITE request F2 before forwarding it as F4. The same applies to request F13 and response F19.

F6 - Bob's UA does not support "log me" marking and does not echo the "log me" marker in response F6. The same applies to response F9 and the BYE request F15.

F7 - Proxy 1 inserts a "log me" marker in 180 response of the INVITE request before forwarding it as F8. The same applies to response F10 and the BYE request F16.

## 5. Errors

### 5.1. Error Cases

The following error cases are possible for "log me" marking.

1. A "log me" marker is unexpectedly missing from a dialog that is being logged.

2. A "log me" marker unexpectedly appears in a dialog that is not being logged

3. A "log me" marker unexpectedly disappears and then reappears in a dialog being logged. This is treated in the same way as case 1.

4. A "log me" marker is unexpectedly missing from a retransmission in a dialog being logged. This is treated in the same way as case 1.

These cases apply to any request or response sent by any entity and in any direction in a dialog being "log me" marked. Detection of these error cases is described in this section.

#### 5.1.1. Missing "Log Me" Marker Error Case

Since "log me" marking is per dialog, if a dialog is being marked and marking is missing from a request or response then this is an error.

However, detecting such errors is not as simple as checking for missing markers because of cases such as non-supporting terminals where it is normal that marking is not done.

Detecting errors must be evaluated separately for each neighbor. It is an error if a particular neighbor has previously sent logme in the dialog and then stops, independently of what has been happening with other neighbors.

User agents and intermediaries that are stateless with respect to "log me" marking are not able detect such errors. User agents and intermediaries that are stateful with respect to "log me" marking are able to detect that a marker is missing from a dialog that has previously been "log me" marked. Error cases are illustrated in this section, and non-error cases in Section 5.2.1.

The following figures illustrate missing "Log me" Marker errors.

Figure 8 shows an error detected at Proxy 1, where an expected "log me" marker is missing.

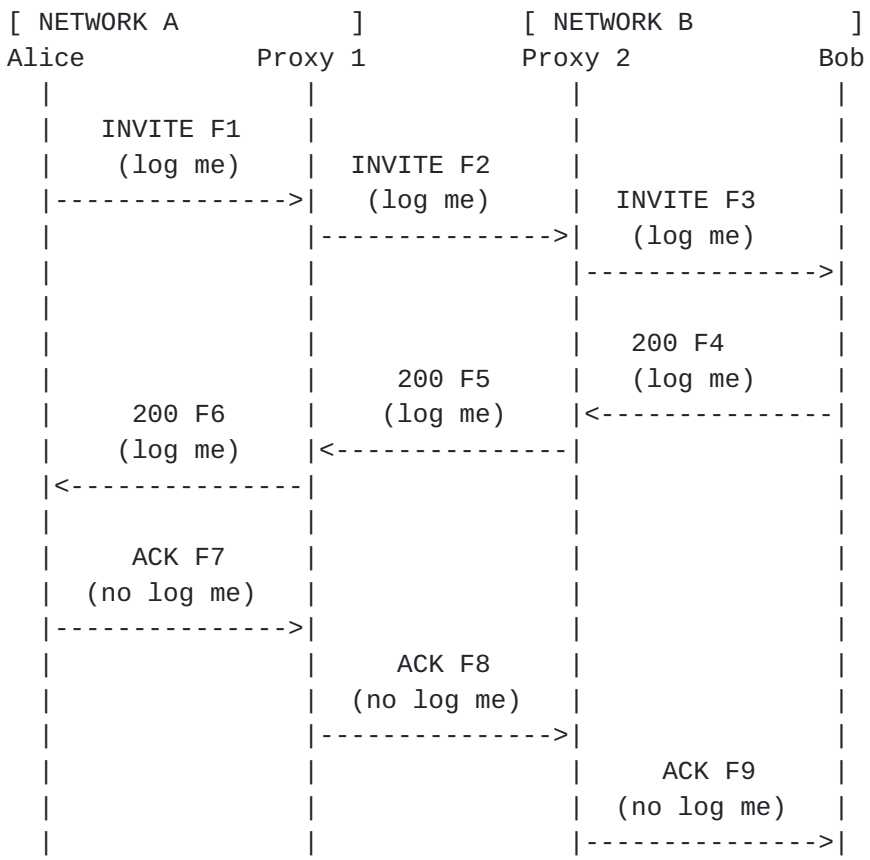



Figure 8: Error case: missing "log me" marker

F1 - Proxy 1 detects the "log me" marker and maintains state that this dialog is to be logged.

F7 - Proxy 1 detects that the expected "log me" marker is missing, considers it as an error and stops "log me" marking in subsequent requests and responses in this dialog.

Figure 9 shows an error detected at Proxy 2 and Bob's user agent.

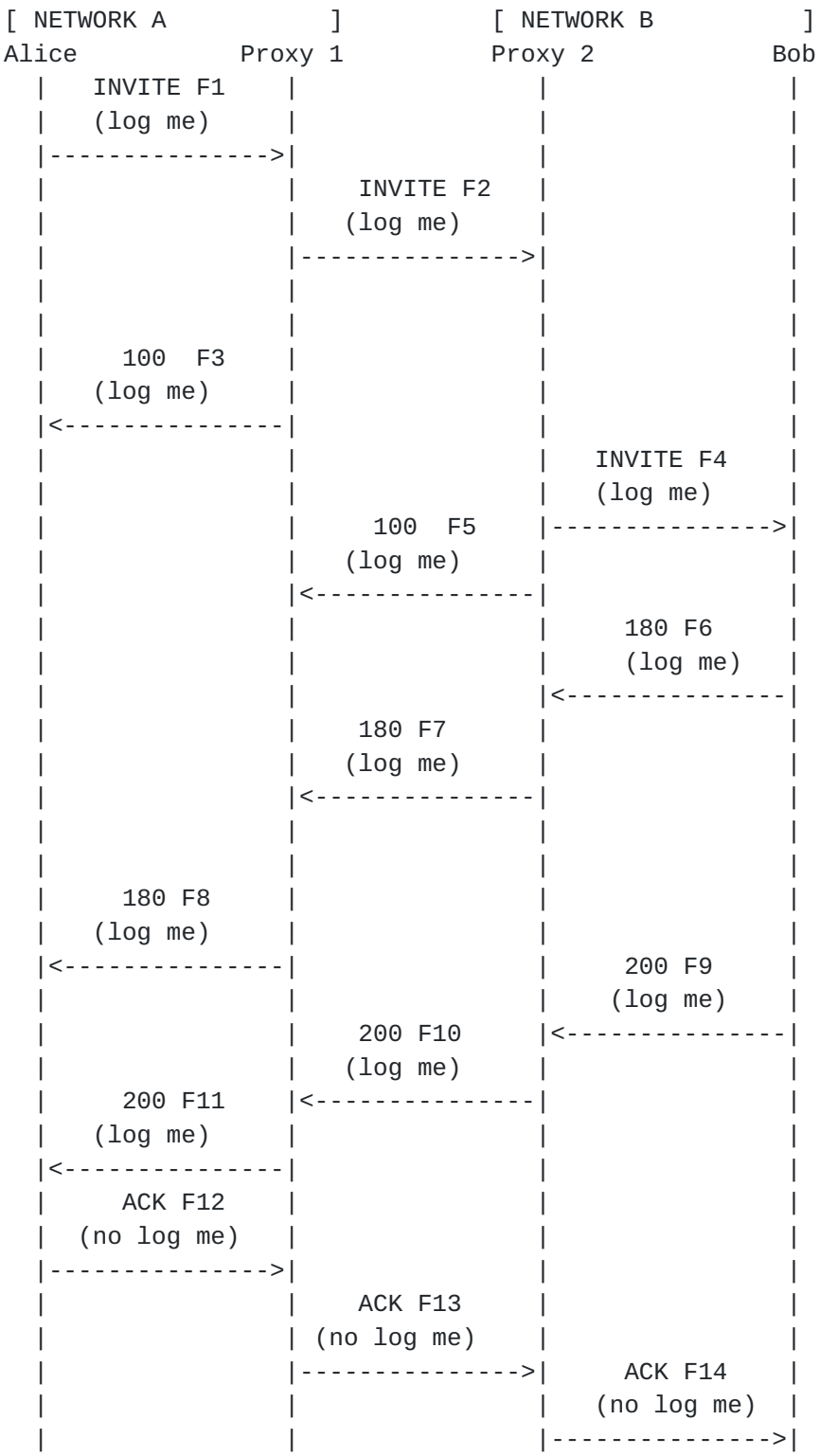



Figure 9: Error case: missing "log me" marker

F2 - Proxy 2 detects the "log me" marker and maintains state that this dialog is to be logged.

F4 - Bob's user agent detects the "log me" marker and maintains state that this dialog is to be logged.

F12 - Proxy 1 detects that the expected "log me" marker is missing, considers it as an error and stops "log me" marking in subsequent requests and responses in this dialog. Hence it does not insert a "log me" marker in F13.

F13 - Proxy 2 detects that the expected "log me" marker is missing, considers it as an error and stops "log me" marking in subsequent requests and responses in this dialog.

F14 - Proxy 2 does not insert a "log me" marker because it has stopped "log me" marking due to an error observed in F13. Bob's UA detects that the expected "log me" marker is missing, considers it as an error and stops "log me" marking in subsequent requests and responses in this dialog.

### 5.1.2. "Log Me" Marker Appears Mid-Dialog Error Case

SIP endpoints, intermediaries acting on behalf of endpoints, and B2BUAs that can perform "log me" marking are stateful. Such entities will expect a "log me" marker only for dialogs where the initial dialog-creating request was "log me" marked, either by themselves or an upstream entity. "Log me" marking that subsequently begins mid-dialog is an error.

Figure 10 illustrates a "log me" marking error observed in the middle of a dialog. Alice's UA supports "log me" marking but the call is not initially marked for logging i.e. INVITE F1 is not "log me" marked. But Alice's UA starts to "log me" mark at the ACK request F7. Proxy 1 supports "log me" marking at the originating network boundary and therefore detects the error, does not log signaling, and removes the "log me" marker before forwarding the ACK request F8.

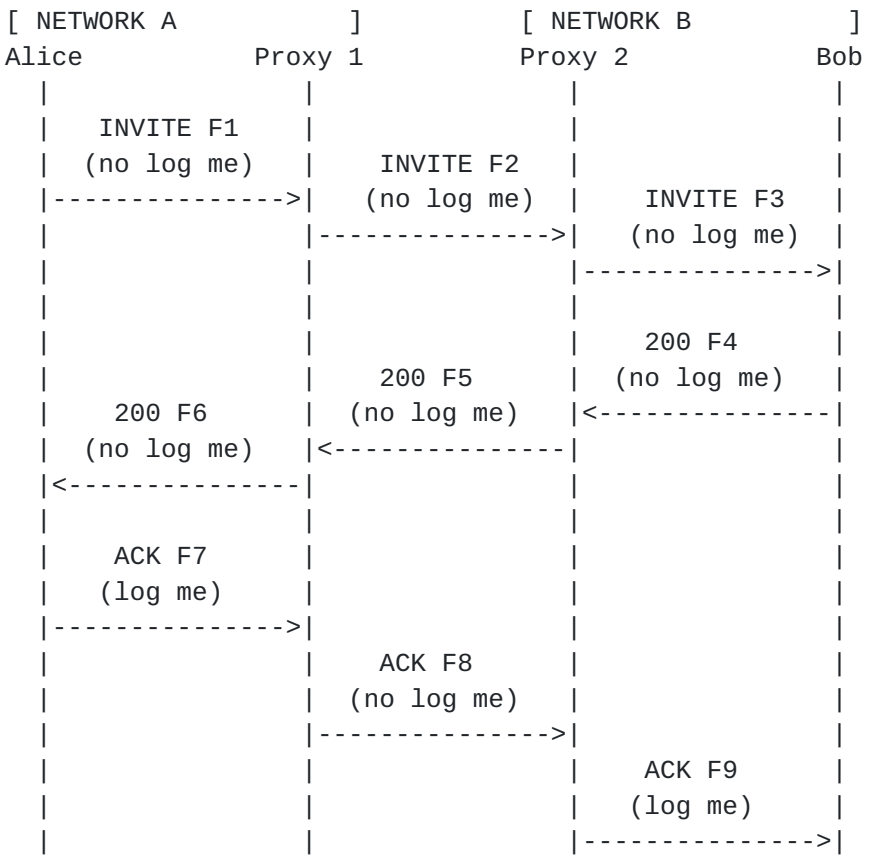



Figure 10: Error case: "log me" marker begins mid-dialog

## 5.2. Non-Error Cases

### 5.2.1. Missing "Log me" Marker Non-Error Case

The following figure illustrates a non-error case.

Figure 11 shows Proxy 2 receiving a response with no "log me" marker that is not an error case. Proxy 2 is configured by network B to perform "log me" marking on behalf of Bob's UA, which does not support "log me" marking. Proxy 2 does not therefore expect responses from Bob to include a "log me" marker.

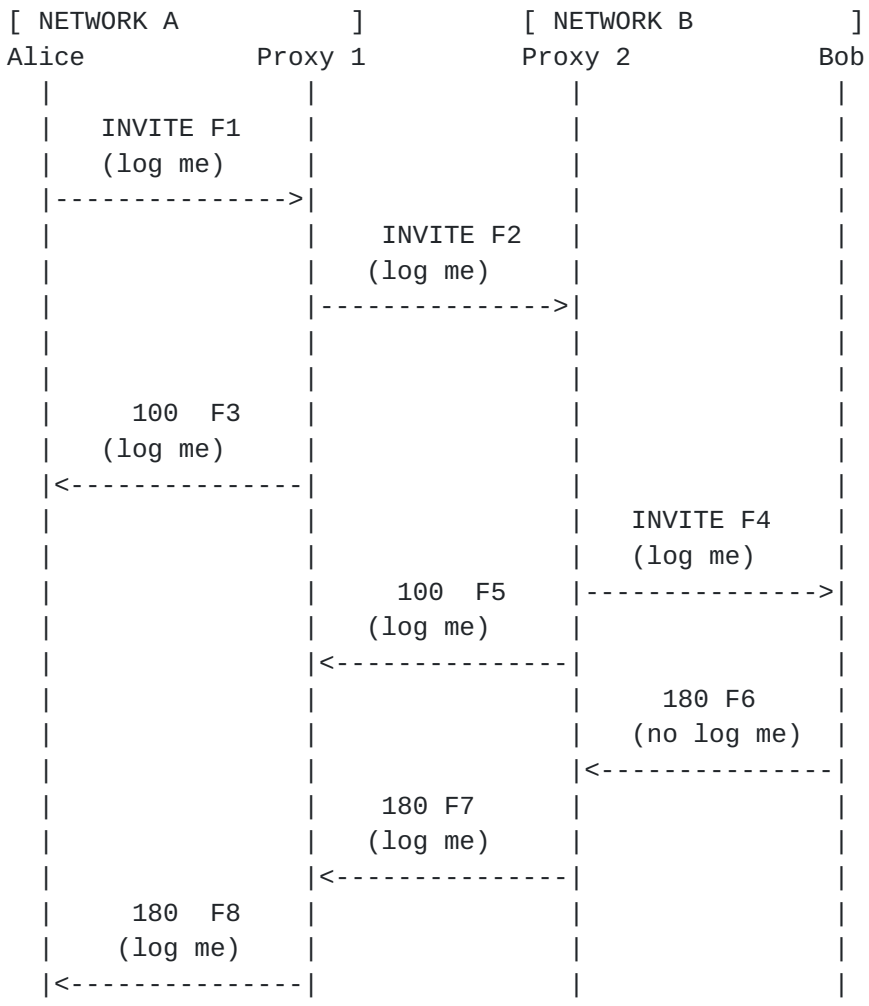



Figure 11: Non-error case: missing "log me" marker

F2 - Proxy 2 detects the "log me" marker and maintains state that this dialog is to be logged. Proxy 2 inserts "log me" markers on behalf of Bob's user agent such as in F7.

F6 - Proxy 2 detects that the "log me" marker is missing from the response but considers "log me" marking to be ongoing as a marker was not expected.

F7 - Proxy 2 continues to "log me" mark requests and responses on behalf of Bob's user agent.

### 5.2.2. "Log Me" Marker Appears Mid-Dialog Non-Error Case

A SIP intermediary that can perform "log me" marking on behalf of an endpoint MAY optionally mark a request or response towards a non-supporting endpoint, such as the 100 response F3 in Figure 3. In this case the endpoint will receive a "log me" marker mid-dialog and is not considered an error.

Another use case is a network in which some but not all endpoints support "log me" marking that wants to avoid treating endpoints differently by always managing "log me" marking at a SIP intermediary. In this case, the endpoint that supports "log me" is not configured to mark a dialog, instead the SIP intermediary is configured to perform "log me" marking on behalf of that endpoint. This case still requires authorization as described in Section 7.1. This SIP intermediary MAY optionally mark a request or response towards the endpoint, such as the 100 response F3 in Figure 3. The endpoint will receive a "log me" marker mid-dialog and this is not considered an error.

### 5.2.3. Combining Dialogs Non-Error Case

When troubleshooting call flows that involve the SIP Join header field specified in [RFC3911], the ideal scenario is to have "log me" marking enabled on all UAs and intermediaries participating in the end-to-end session. If the ideal scenario is not feasible, the following rules apply.

- If a "log me"-aware endpoint or intermediary that is already "log me" marking a dialog receives a SIP INVITE with a Join header field and without a "log me" marker, it MUST NOT "log me" mark responses and requests exchanged within the new dialog established as a result of processing the SIP INVITE.

- If a "log me"-aware endpoint or intermediary that is not "log me" marking a dialog receives a SIP INVITE with a Join header field and with a "log me" marker, it MUST "log me" mark responses and requests exchanged within the new dialog established as a result of processing the SIP INVITE as per Section 4 of this document.

## 5.3. Error Handling

The two error types that SIP entities must handle are defined in Section 5.1: a missing marker error and an error of "log me" marking that begins mid-dialog. Section 5.2 gives exceptions which have a missing marker or marking that begins mid-dialog but are not errors.

If a missing marker error is detected by a UA, SIP intermediary, or B2BUA, it SHOULD consider this as an error condition in the "log me" functionality. It MUST NOT mark subsequent requests and responses and MUST stop logging messages in the same dialog. Any previously logged messages SHOULD be retained, for the time period defined in Section 8.5, and not deleted.

If a "log me" marking that begins mid-dialog error is detected by a UA, SIP intermediary, or B2BUA, it SHOULD consider this as an error condition in the "log me" functionality. It MUST NOT forward the "log me" marker and MUST NOT log the message. It MUST NOT mark subsequent requests and responses and MUST NOT log subsequent messages in the same dialog.

"Log me" marking errors can be detected and handled only by supporting UAs or B2BUAs. A SIP proxy as defined in [RFC3261] cannot detect or handle marking errors and will simply forward any "log me" marker it receives.

## 6. Augmented BNF for the "logme" Parameter

ABNF is described in [RFC5234]. This document introduces a new "logme" parameter for the Session-ID header field defined in Section 5 of [RFC7989].

```
sess-id-param       =/ logme-param

logme-param         = "logme"
```

Figure 12: Augmented BNF for the "logme" Parameter

## 7. Security Considerations

### 7.1. "Log Me" Authorization

"Log me" marking MUST be disabled by default both at the endpoints and intermediaries and MUST be enabled only by authorized users. For example, an end user or network administrator must give permission for a terminal that supports "log me" marking in order to initiate marking. Similarly, a network administrator must enable a configuration at the SIP intermediary to perform "log me" marking on behalf of a terminal that does not support "log me" marking. The permission MUST be limited to only specific calls of interest that are originated in a given time duration.

Activating a debug mode affects the operation of a terminal, therefore debugging configuration MUST be supplied by an authorized party to an authorized terminal through a secure communication channel.

### 7.2. "Log Me" Marker Removal

The log me marker is not sensitive information, although it will sometimes be inserted because a particular device is experiencing problems.

The presence of a log me marker will cause some SIP entities to log signaling messages. Therefore, this marker MUST be removed at the earliest opportunity if it has been incorrectly inserted, such as appearing mid-dialog in a dialog that was not being logged or outside the configured start and stop of logging.

If SIP requests and responses are exchanged with an external network with which there is no agreement to pass "log me" marking, then the "log me" marking is removed as mandated in Section 3.4.2. This behavior applies to incoming and outgoing requests and responses.

### 7.3. Denial of Service Attacks

Maliciously configuring a large number of terminals to simultaneously mark dialogs with a "log me" marker will cause high processor load on SIP entities that are logging signaling. Since "log me" marking is for the small number of dialogs subject to troubleshooting or regression testing, the number of dialogs that can be simultaneously logged can be statically limited without adversely affecting the usefulness of "log me" marking. Also, the SIP intermediary closest to the terminal and SIP intermediary at network edge (e.g Session Border Controllers) can be configured to screen-out "log me" markers when troubleshooting or regression testing is not in progress.

### 7.4. Data Protection

A SIP entity that has logged information MUST protect the logs. Storage of the log files are subject to the security considerations specified in [RFC6872].

## 8. Privacy Considerations

Logging includes all SIP header fields. The SIP privacy mechanisms defined in [RFC3323] can be used to ensure that logs do not divulge personal identity information in the core SIP header fields specified in [RFC3261].

Privacy mechanisms might also need to be applied to header fields defined by SIP extensions and for managing the confidentiality of the Request URI and SIP header fields and bodies.

## 8.1. Personal Identifiers

"Log me" marking is defined for the SIP Protocol, and SIP has header fields such as From, Contact, P-Asserted-Identity that can carry personal identifiers. Different protocol interactions can be correlated using the Session-ID and Call-ID header fields, but such correlation is limited to a single end-to-end session.

In order to protect user privacy during logging, privacy settings can be enabled or requested by the terminal used by the end user. [RFC3323] suggests two mechanisms:

- By using the value anonymous in the From header field
- By requesting header- and session-level privacy from SIP intermediaries using the Privacy header

Endpoints that support Globally Routable User Agent URIs (GRUUs) can use a temporary GRUU (see Section 3.1.2 of [RFC5627]) assigned by the Registrar in order to protect user privacy as discussed in Section 10.3 of [RFC5627].

Intermediaries that perform "log me" marking on behalf of the endpoints (see Section 4.3) may also be configured to apply privacy (as defined in Section 3.3 of [RFC3323]) on messages that belong to a dialog that is "log me" marked.

Complete anonymization (e.g. the Request URI and the "username" field in the "o=" parameter of an SDP body) may not be possible in all circumstances and therefore administrators of the originating and terminating networks should consider how privacy will be ensured when providing consent for "log me" marking.

"Log me" marking is typically used for troubleshooting and regression testing, and in some cases a service provider owned device with a dummy account can be used instead of a customer device. In such cases, no personal identifiers are included in the logged signaling messages.

## 8.2. Data Stored at SIP Intermediaries

SIP endpoints and intermediaries that honor the "log me" request store all the SIP messages that are exchanged within a given dialog. SIP messages can contain the personal identifiers listed in

Section 8.1 and additionally a user identity, calling party number, IP address, hostname, and other user and device related items. The SIP message bodies describe the kind of session being set up by the identified end user and device.

"Log me" marking does not introduce any additional user or device data to SIP but might indicate that a specific user is experiencing a problem.

If the SIP SDP parameters [sdp-parameters] contain sensitive security information (e.g. encryption keys) such as "crypto" [RFC4568], 3GPP-Integrity-Key, or 3GPP-SRTP-Config [RFC6064] attributes then the attribute value MUST be masked with a dummy value prior to storing the message in a log file. For example, the attribute value can be replaced with a string of special characters like "X", "*" and "#" as shown in the example below.

```
a=crypto:XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 XXXXXXXXXXXXX
```

## 8.3. Data Visible at Network Elements

SIP messages that are logged due to "log me" requests are stored only by the SIP initiators, intermediaries and recipients. Enablers as defined in section 3.1 of [RFC6973], such as firewalls and DNS servers do not log messages due to the "log me" marking.

## 8.4. Preventing Fingerprinting

"Log me" functionality is typically used to troubleshoot a given problem and hence it can be used as an method to identify users and devices that are experiencing issues. The best way to prevent fingerprinting of users is to enable or request SIP privacy for the logged dialog.

## 8.5. Retaining Logs

The lifetime of "log me" marking is equivalent to the lifetime of the dialog that initiated the "log me" request. When "log me" is extended to related dialogs the lifetime is extended until there is no more related dialog for the end-to-end session.

"Log me" automatically expires at the end of the dialog and there is no explicit mechanism to turn off logging within a dialog.

The scope of "log me" Marking is limited i.e. an user or the network administrator has to enable it on a per session basis or for a

specific time period. This minimizes the risk of exposing user data for an indefinite time.

The retention time period for logged messages SHOULD be the minimum needed for each particular troubleshooting or testing case. The retention period is configured based on the data retention policies of service providers and enterprises.

### 8.6. User Control of Logging

Consent to turn on "log me" marking for a given session MUST be provided by the end user or by the network administrator. It is handled outside of the protocol through user interface or application programming interfaces at the end point, call control elements and network management systems.

Originating and terminating endpoints that are "log me" aware and have a user interface MUST indicate (using text, icon etc.) to the user that a session is being logged.

SIP entities across the communication path MAY be configured to pass through the "log me" marking but not honor the request i.e. not log the data based on local policies.

### 8.7. Recommended Defaults

The recommended defaults for "log me" marking are:

- turn on SIP privacy as described in Section 8 or use a service provider owned device with a dummy user identity for test calls

- use the local UUID of Session-ID header field at the originating device as the test case identifier as described in Section 3.3

## 9. IANA Considerations

### 9.1. Registration of the "logme" Parameter

The following parameter is to be added to the "Header Field Parameters and Parameter Values" section of the SIP parameter registry:

| Header Field | Parameter Name | Predefined Values | Reference |
|---|---|---|---|
| Session-ID | logme | No (no values are allowed) | [RFCXXXX] |

Table 1

## 10. Acknowledgments

The authors wish to thank Paul Giralt, Paul Kyzivat, Jorgen Axell, Christer Holmberg, Vijay Gurbani, Ben Campbell, Gonzalo Salgueiro, Francesca Palombini, Adam Roach, Mirja Kuhlewind, Benjamin Kaduk, Eric Rescorla, Alissa Cooper, Warren Kumari, and Alexey Melnikov for their constructive review comments and guidance while developing this document.

## 11. References

### 11.1. Normative References

[application/vnd.tcpdump.pcap] Harris, G., "MIME type application/vnd.tcpdump.pcap", March 2011, <https://www.iana.org/assignments/media-types/application/vnd.tcpdump.pcap>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, DOI 10.17487/RFC3261, June 2002, <https://www.rfc-editor.org/info/rfc3261>.

[RFC3323] Peterson, J., "A Privacy Mechanism for the Session Initiation Protocol (SIP)", RFC 3323, DOI 10.17487/RFC3323, November 2002, <https://www.rfc-editor.org/info/rfc3323>.

[RFC3891] Mahy, R., Biggs, B., and R. Dean, "The Session Initiation Protocol (SIP) "Replaces" Header", RFC 3891, DOI 10.17487/RFC3891, September 2004, <https://www.rfc-editor.org/info/rfc3891>.

[RFC3911] Mahy, R. and D. Petrie, "The Session Initiation Protocol (SIP) "Join" Header", RFC 3911, DOI 10.17487/RFC3911, October 2004, <https://www.rfc-editor.org/info/rfc3911>.

[RFC4538] Rosenberg, J., "Request Authorization through Dialog Identification in the Session Initiation Protocol (SIP)", RFC 4538, DOI 10.17487/RFC4538, June 2006, <https://www.rfc-editor.org/info/rfc4538>.

[RFC4568] Andreasen, F., Baugher, M., and D. Wing, "Session Description Protocol (SDP) Security Descriptions for Media Streams", RFC 4568, DOI 10.17487/RFC4568, July 2006, <https://www.rfc-editor.org/info/rfc4568>.

[RFC5627] Rosenberg, J., "Obtaining and Using Globally Routable User Agent URIs (GRUUs) in the Session Initiation Protocol (SIP)", RFC 5627, DOI 10.17487/RFC5627, October 2009, <https://www.rfc-editor.org/info/rfc5627>.

[RFC6064] Westerlund, M. and P. Frojdh, "SDP and RTSP Extensions Defined for 3GPP Packet-Switched Streaming Service and Multimedia Broadcast/Multicast Service", RFC 6064, DOI 10.17487/RFC6064, January 2011, <https://www.rfc-editor.org/info/rfc6064>.

[RFC6872] Gurbani, V., Ed., Burger, E., Ed., Anjali, T., Abdelnur, H., and O. Festor, "The Common Log Format (CLF) for the Session Initiation Protocol (SIP): Framework and Information Model", RFC 6872, DOI 10.17487/RFC6872, February 2013, <https://www.rfc-editor.org/info/rfc6872>.

[RFC6873] Salgueiro, G., Gurbani, V., and A. Roach, "Format for the Session Initiation Protocol (SIP) Common Log Format (CLF)", RFC 6873, DOI 10.17487/RFC6873, February 2013, <https://www.rfc-editor.org/info/rfc6873>.

[RFC7989] Jones, P., Salgueiro, G., Pearce, C., and P. Giralt, "End-to-End Session Identification in IP-Based Multimedia Communication Networks", RFC 7989, DOI 10.17487/RFC7989, October 2016, <https://www.rfc-editor.org/info/rfc7989>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[sdp-parameters]
"Session Description Protocol (SDP) Parameters", June 2001, <https://www.iana.org/assignments/sdp-parameters/sdp-parameters.xhtml>.

### 11.2. Informative References

[RFC3665] Johnston, A., Donovan, S., Sparks, R., Cunningham, C., and K. Summers, "Session Initiation Protocol (SIP) Basic Call Flow Examples", BCP 75, RFC 3665, DOI 10.17487/RFC3665, December 2003, <https://www.rfc-editor.org/info/rfc3665>.

[RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <https://www.rfc-editor.org/info/rfc5234>.

[RFC5589] Sparks, R., Johnston, A., Ed., and D. Petrie, "Session Initiation Protocol (SIP) Call Control - Transfer", BCP 149, RFC 5589, DOI 10.17487/RFC5589, June 2009, <https://www.rfc-editor.org/info/rfc5589>.

[RFC6973] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., Hansen, M., and R. Smith, "Privacy Considerations for Internet Protocols", RFC 6973, DOI 10.17487/RFC6973, July 2013, <https://www.rfc-editor.org/info/rfc6973>.

[RFC7092] Kaplan, H. and V. Pascual, "A Taxonomy of Session Initiation Protocol (SIP) Back-to-Back User Agents", RFC 7092, DOI 10.17487/RFC7092, December 2013, <https://www.rfc-editor.org/info/rfc7092>.

[RFC7206] Jones, P., Salgueiro, G., Polk, J., Liess, L., and H. Kaplan, "Requirements for an End-to-End Session Identification in IP-Based Multimedia Communication Networks", RFC 7206, DOI 10.17487/RFC7206, May 2014, <https://www.rfc-editor.org/info/rfc7206>.

[RFC8123] Dawes, P. and C. Arunachalam, "Requirements for Marking SIP Messages to be Logged", RFC 8123, DOI 10.17487/RFC8123, March 2017, <https://www.rfc-editor.org/info/rfc8123>.

Authors' Addresses

Peter Dawes
Vodafone Group
The Connection
Newbury, Berkshire RG14 2FN
UK

Email: peter.dawes@vodafone.com

Chidambaram Arunachalam
Cisco Systems
7200-12 Kit Creek Road
Research Triangle Park, NC 27709
US

Email: carunach@cisco.com