SIP Protocol

- What is SIP?
- Real life example
- RFC 8497 - Problem Statement
- Solution Explanation

Prepared by Atul Sharma
Guided by Sudhendu Kumar

SIP

History of SIP

Traditionally, in the circuit-switch network, SS7 signaling was designed. E1/T1 are the examples of SS7 signalling which carries, D-Channel for all signalling to travel from source to destination in IP Telephony. As these links are end to end dedicated in nature, cost is high and only limited transmission speed of 2.048 or 1.544 Mbps can be achieved to carry voice traffic. For any additional capacity, new dedicated E1/T1 links are required to be provisioned.

In packet switched networks, packet can travel with high speed and with low cost, as it travel over the same data path/link without any dedicated links. But there was a need of a signalling control protocol which can run in control plane to control call establishment and tearing down.

Cont....

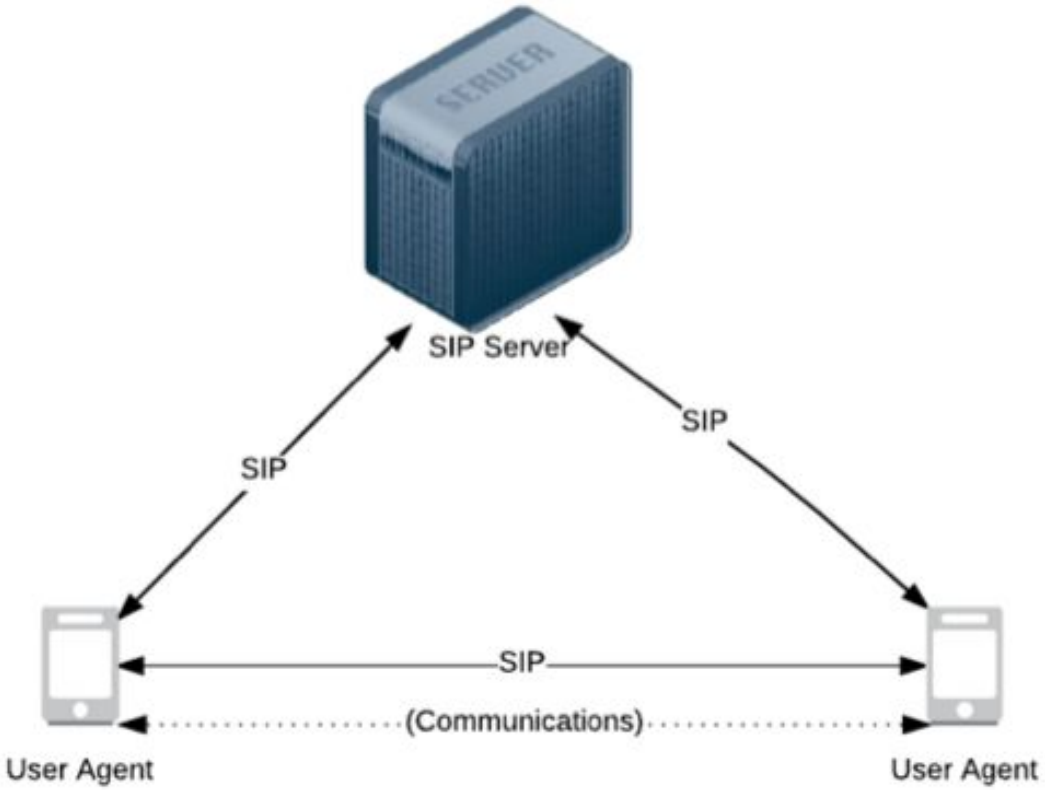
- SIP - Session Initiation Protocol which is an application layer control protocol.
- SIP runs over TCP/UDP protocol
- Source endpoint initiate the SIP invite to destination endpoint.

SIP Protocol is widely being used in Voice over IP Telephony i.e voice communication but also supports Video Conferencing, Instant messaging and File Transfer.

All communication messages between endpoints are controlled and manage by the SIP device called SIP Server.

Simple SIP Architecture -

User Agents - IP Phone, Softphone, Mobile device

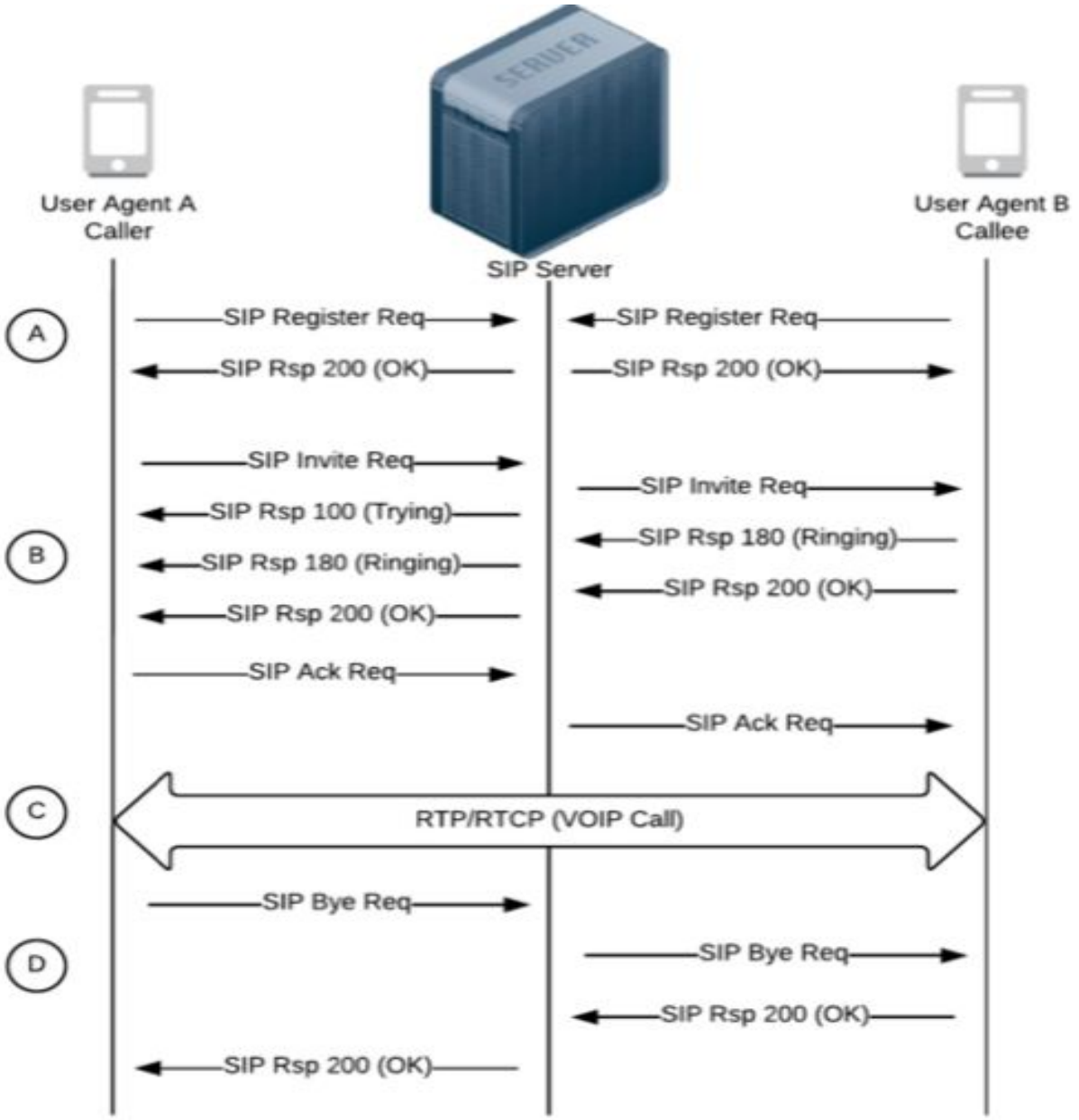


Simple SIP Architecture

Diagrom Ref - https://www.extrahop.com/company/blog/2016/sip-protocol/

SIP Protocol

Message Flow -



Diagrom Ref - https://www.extrahop.com/company/blog/2016/sip-protocol/

Real life example

Enterprise Architecture with remote branch locations connected via Operator network

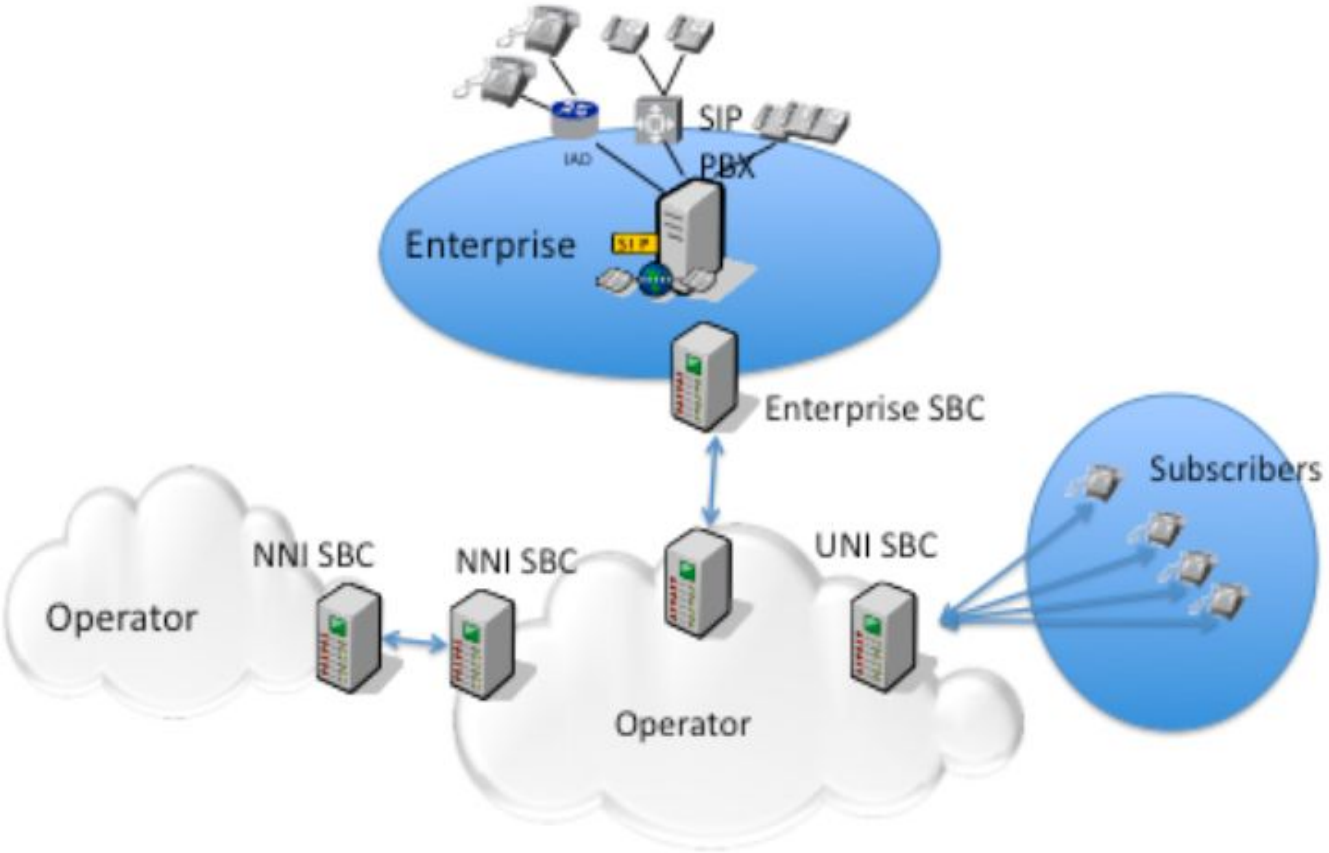


Diagram Ref - https://www.frafos.com/resources/white-papers/understanding-session-border-controllers/

#RFC 8497 Problem Statement

- When UA1 initiate SIP call to UA2, both are connected via Operator Network, may experience SIP signalling issue.

- In such scenario, troubleshooting and finding out root cause of any kind of issue is not easy task. Such kind of regressive testing is required even if any hardware or software upgrade.

- Troubleshooting in such scenarios can not be conducted on the large network traffic, however it can performed over small portion of traffic.

- Even, predicting path may not be possible when traffic traverse over multiple Operator networks.

Proposed Solution

- To identify path and dialog, a mechanism has to be devised to mark dialog/session so that SIP Entities along the path can provide diagnostic logging. Such marking is "***log me***", marking in SIP Signalling.

#RFC 8123 - discussed about the 12 requirement of adding an indicator in the SIP PDU.

“Log me” marker Explained

- “Log me” is more effective when used end to end in sessions.
- “Log me” marker should be passed unchanged through SIP intermediaries.
- Session-ID (chosen parameter) header passes through SIP B2BUA.
- UUID of the session id header could be used for the test case.

- Marking should start at the beginning of the dialog.
- User Agent (UA) or Intermediary can start the marking on behalf of the UA if needed.

SIP Entity Behaviour :

- SIP entities only mark current dialog wich are needed for testing purpose i.e troubleshooting and testing.
- Log me marking starts at the creation of the dialog and end with the dialog ends.
- It only initiated by SIP endpoint or intermediary that marks on behalf of originating endpoint.
- Intermediary can only initiate Log me when originating endpoint define the action.

Cont..

➢ Two cases to mark a request or response by UA or Intermediary -

(a) First, it has been configured to do so on the UA or intermidary.

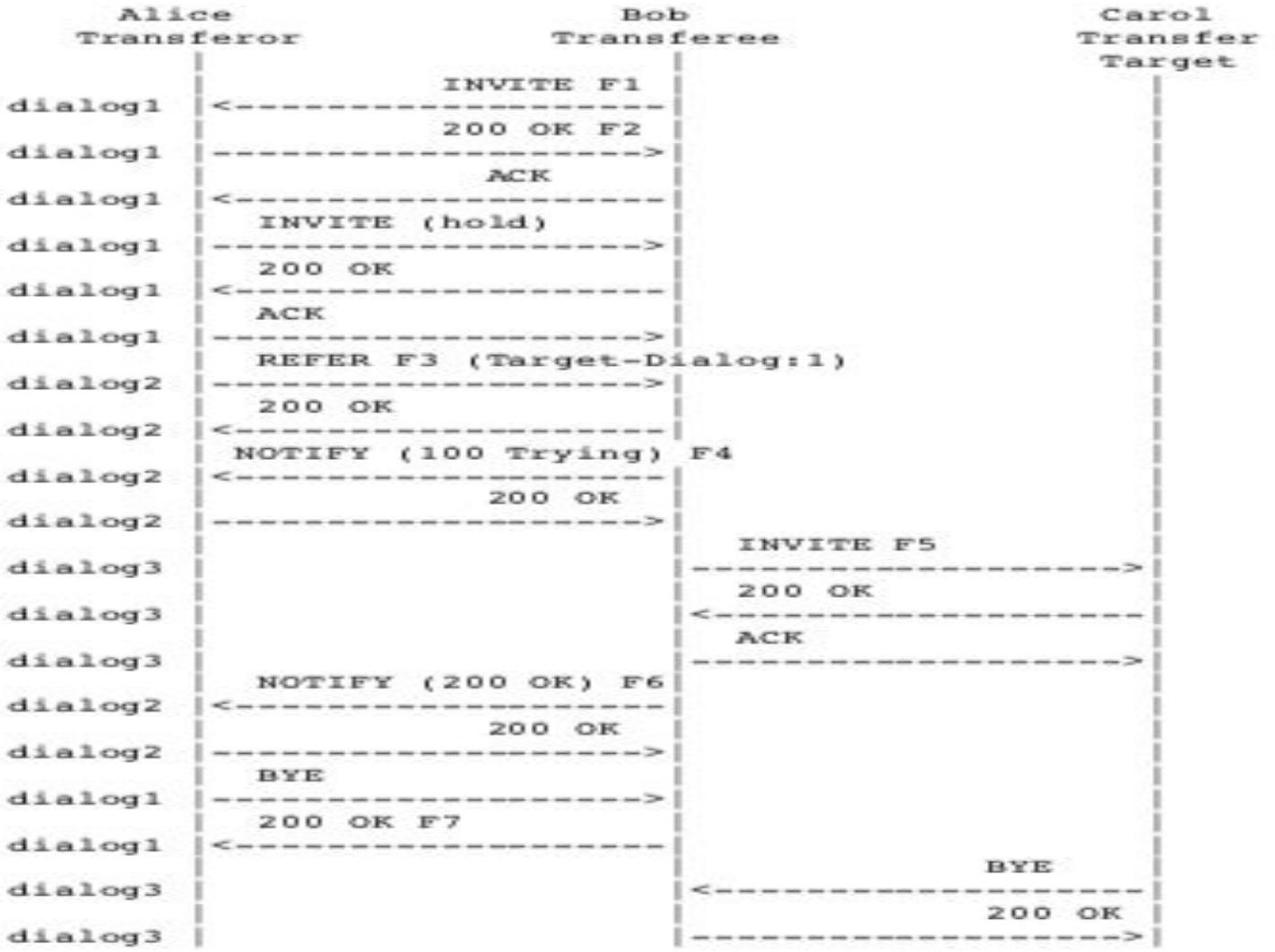
(b) Second, it has been maintained the state of session-id hearder to be “log me” mark if already marked.

Passing Marker:

- To/From a User Device
- To/From an External Network
- Across a non supporting SIP Intermediary

Multiple simultaneous dialogs can be initiated by using different Test Case Identifier (UUID in Session ID header) e.g. conference calls or call transfer.

Example


Alice
Transferor
Bob
Transferee
Carol
Transfer
Target
INVITE F1
dialog1
200 OK F2
dialog1
ACK
dialog1
INVITE (hold)
dialog1
200 OK
dialog1
ACK
dialog1
REFER F3 (Target-Dialog:1)
dialog2
200 OK
dialog2
NOTIFY (100 Trying) F4
dialog2
200 OK
dialog2
INVITE F5
dialog3
200 OK
dialog3
ACK
dialog3
NOTIFY (200 OK) F6
dialog2
200 OK
dialog2
BYE
dialog1
200 OK F7
dialog1
BYE
dialog3
200 OK
dialog3

Originating UA or Intermediary on behalf of originating UA -

Must insert Log Me in dialog creating SIP request.

Below trace has “logme” marker captured in the SIP call trace.

Source Domain - biloxi.example.com

Destination Domain - atlanta.example.com

```
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: 0904592435881734445
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: ...
```

Destination UA or destination Intermediary:

Can not initiate marking but should detect that dialog is enabled with “log me” marker or not.

And accordingly take action on the request and response and mark with “logme” marker in the same dialog.

```
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: ...
```

Log me marker Processing by SIP Intermediaries between Networks

Network boundary are the entity which are in different administrative domains. When SIP signalling crosses the network boundary or traverse from one network to other operator network, it prevents the end-to-end testing/troubleshooting.

These network boundary devices are called ***Session Border Controller***.

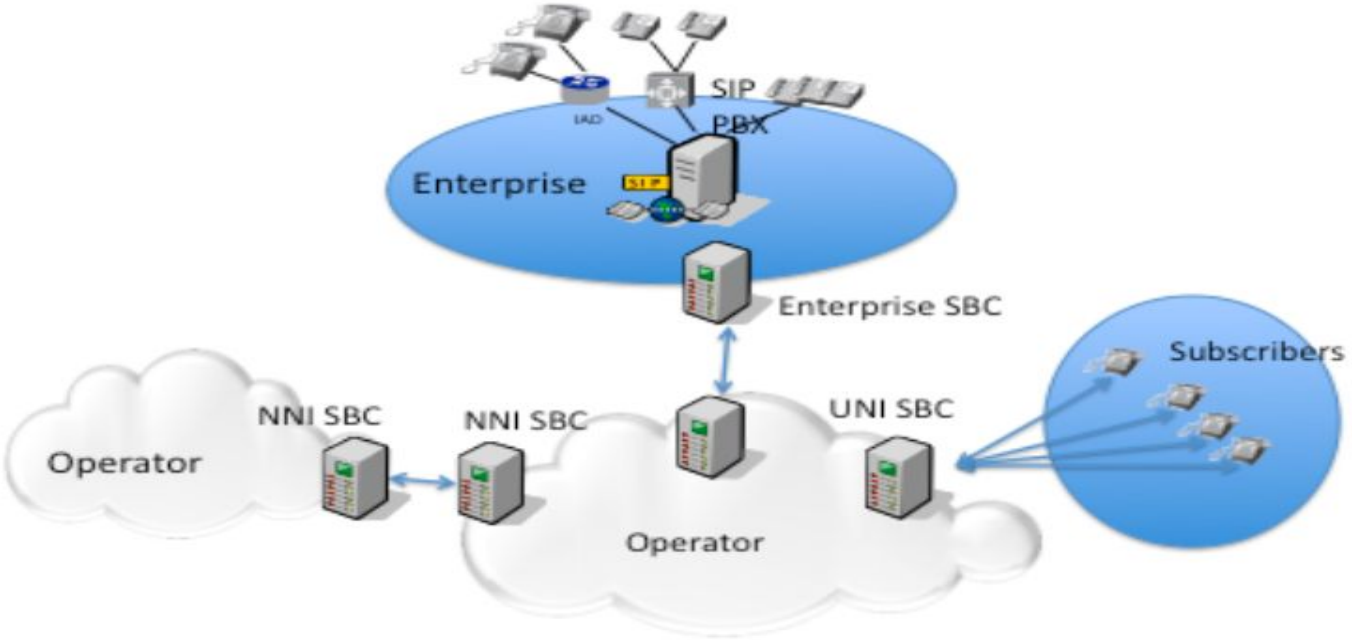


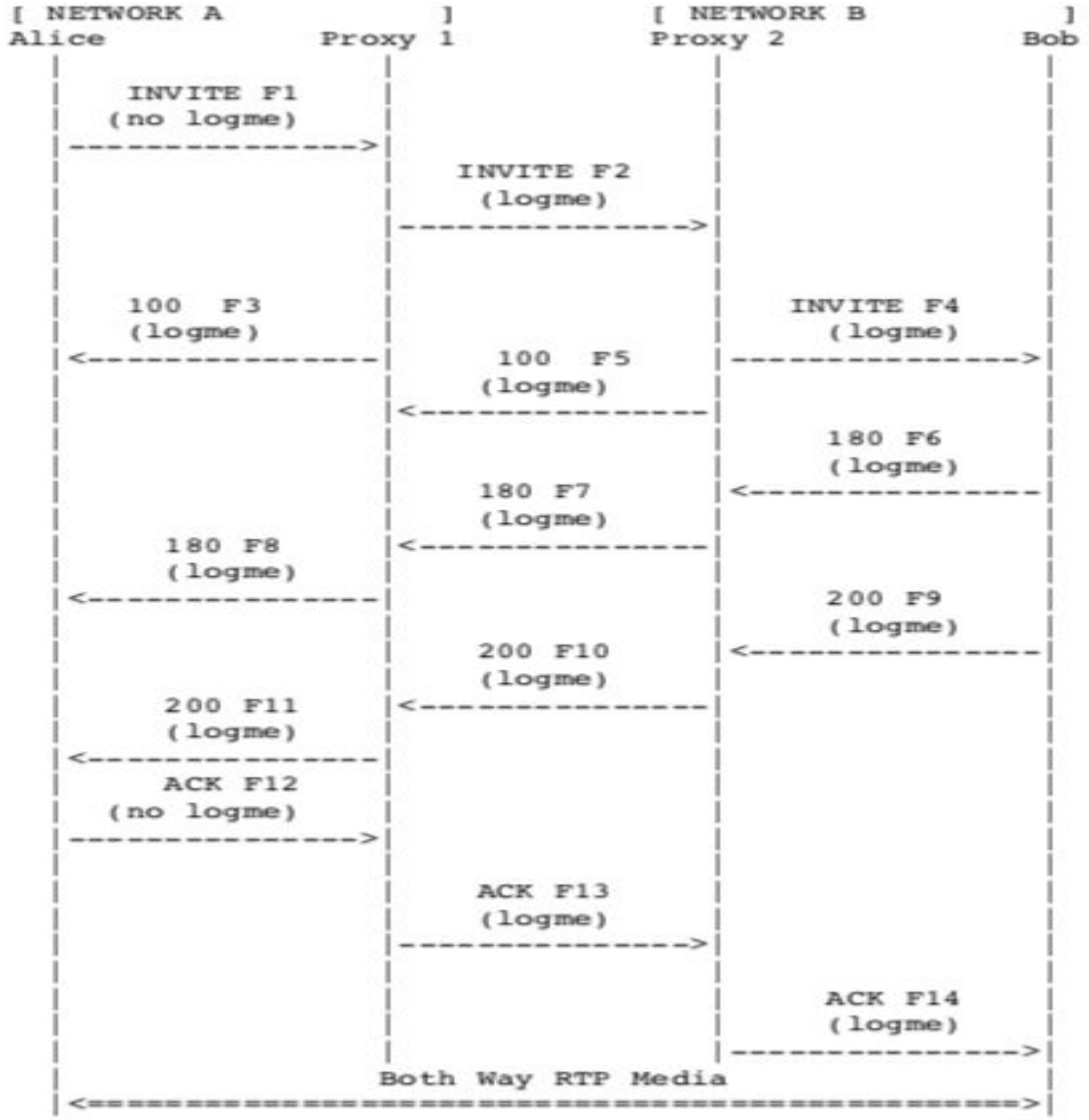
Diagram Ref - https://www.frafos.com/resources/white-papers/understanding-session-border-controllers/

Points to be noted

- Marker should not cross the originating or terminating network boundary.
- All SIP requests and responses should be removed by the network boundary device before forwarding external network and add log me marker to the same SIP dialog.

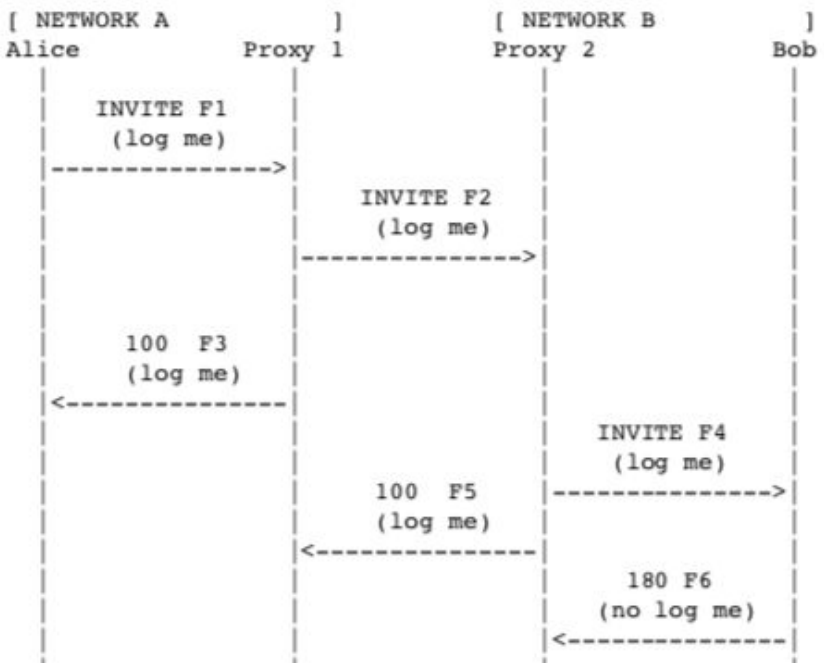
- Similarly, terminating network boundary device should remove the log me marker from the dialog incoming or outgoing SIP dialog before forwarding to/from UA.
- Add log me marker before sending to the terminating UA and external network.

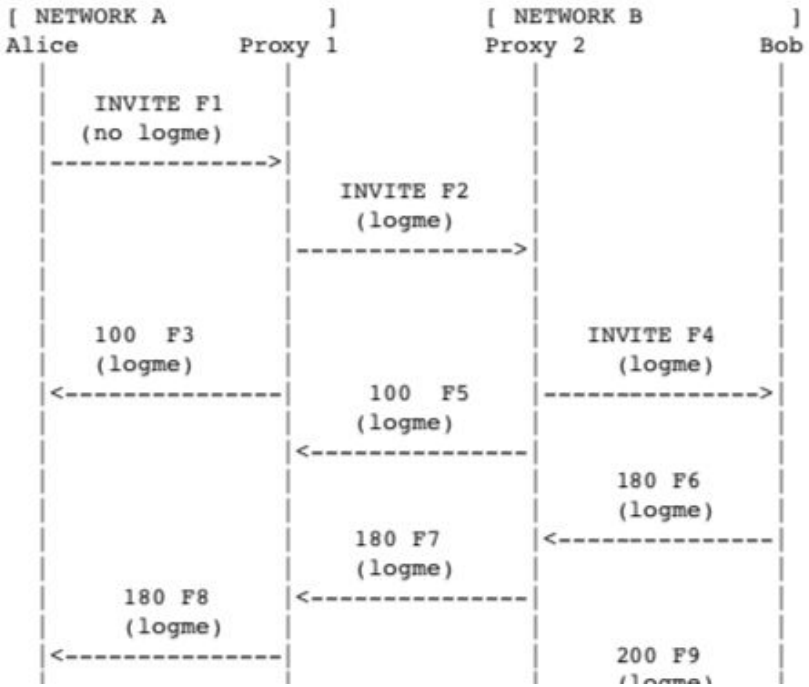
Example -


[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

Stateful Processing Cases -

1. Log me marker not supported by Originating UA

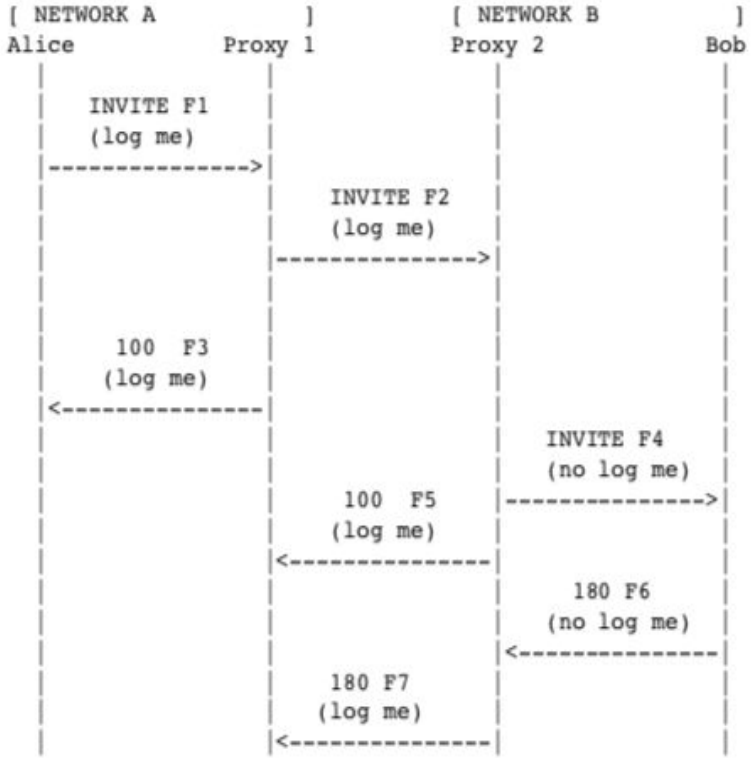


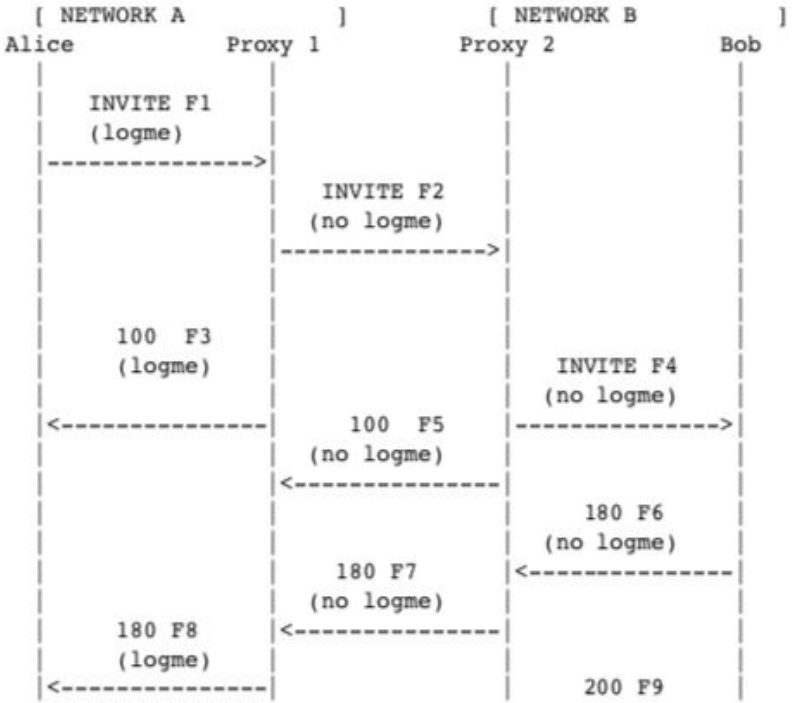


<---- 2. Log me marker not supported by

Terminating UA

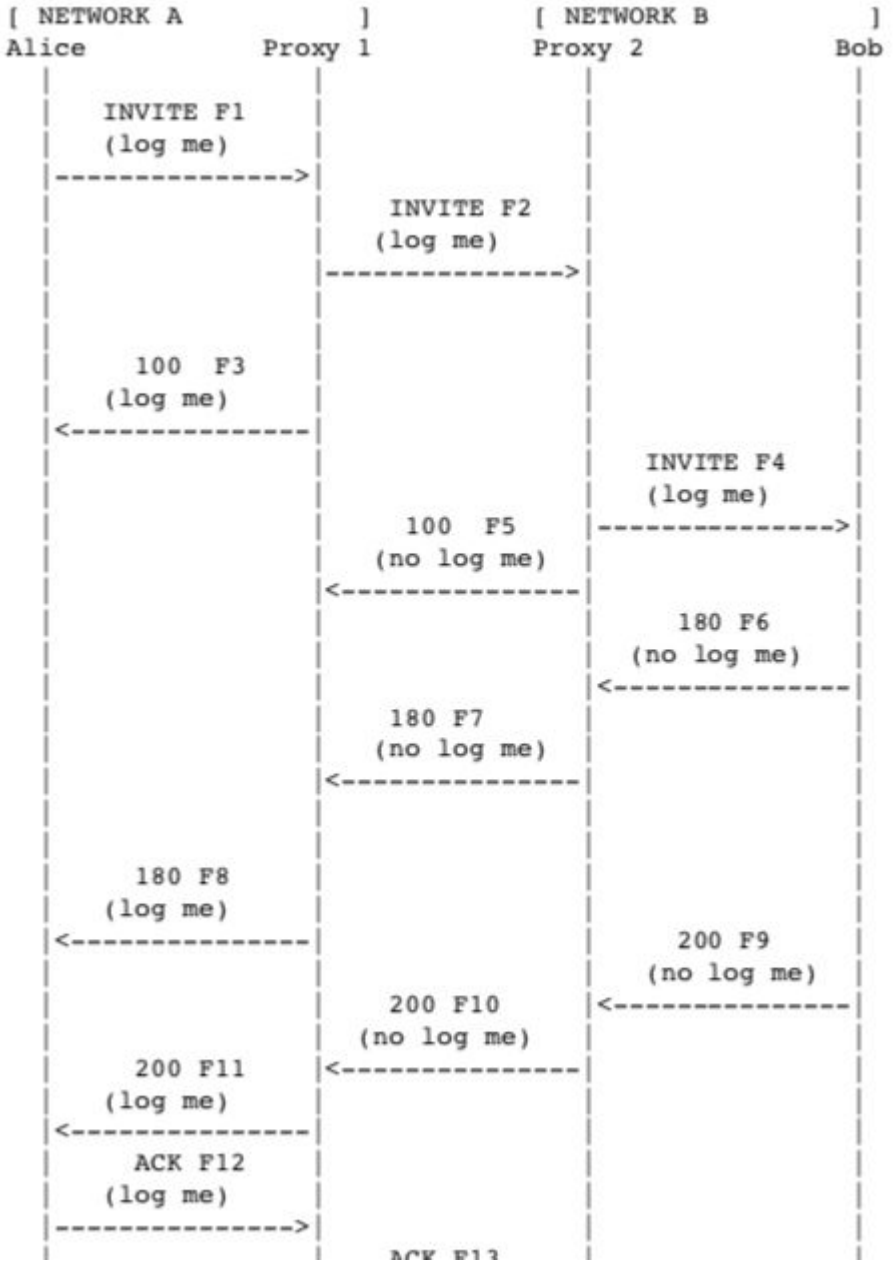
3. Log me marker removed by Originating Network ------>





←----- 4. Log me marker removed by Terminating Network

5. Log me marking passed by non supporting Terminating Network



Error Cases

If log me marker is

1. Missing
2. Appeared in mid dialog
3. Appear and disappeared (treated as case 1)
4. Missing in retransmission (treated as case 1)

Note -

Log me marker can be detected and handled by supporting UA or B2BUA. SIP Proxy can not perform this task and simply forward any log me marker it receives.

Security Considerations

Log me Authorization

Log me marker Removal

DOS Attack

Data Protection

SIP entity that has logs, should be stored safely.

Privacy Considerations

Logs should have any personal identity information in the SIP header.

IANA consideration of Log me parameter:-

Header Field	Parameter Name	Predefined Values	Reference
Session-ID	logme	No (no values are allowed)	[RFC8497]

Source Code

```
2941        if ((bytes->len == 16) && (g_ascii_strcasecmp(param_name, "remote") == 0) &&
2942            tvb_get_string_bytes(tvb, equals_offset + 1, line_end_offset - equals_offset - 1,
2943                                 ENC_UTF_8|ENC_STR_HEX|ENC_SEP_NONE, uuid, NULL) &&
2944            (uuid->len == 16)) {
2945            /* Decode header as draft-ietf-insipid-session-id
2946             *
2947             * session-id          = "Session-ID" HCOLON session-id-value
2948             * session-id-value    = local-uuid *(SEMI sess-id-param)
2949             * local-uuid          = sess-uuid / null
2950             * remote-uuid         = sess-uuid / null
2951             * sess-uuid           = 32(DIGIT / %x61-66)  ;32 chars of [0-9a-f]
2952             * sess-id-param       = remote-param / generic-param
2953             * remote-param        = "remote" EQUAL remote-uuid
2954             *
2955             * ==========Sudhendu's comment starts==========
2956             * Look for the semi-colon. Parse 5 Bytes "logme".
2957             * Confusion while display:
2958             * "Log Me: logme"?
2959             *
```

```
2960            * logme              = "logme"
2961            * ==========Sudhendu's comment ends============
2962            * null               = 32("0")
2963            *
2964            */
2965           e_guid_t guid;
2966
2967           proto_item_set_hidden(pi);
2968           guid.data1 = (bytes->data[0] << 24) | (bytes->data[1] << 16) |
2969                        (bytes->data[2] <<  8) |  bytes->data[3];
2970           guid.data2 = (bytes->data[4] <<  8) |  bytes->data[5];
2971           guid.data3 = (bytes->data[6] <<  8) |  bytes->data[7];
2972           memcpy(guid.data4, &bytes->data[8], 8);
2973           proto_tree_add_guid(tree, hf_sip_session_id_local_uuid, tvb,
2974                               start_offset, semi_colon_offset - start_offset, &guid);
2975           guid.data1 = (uuid->data[0] << 24) | (uuid->data[1] << 16) |
2976                        (uuid->data[2] <<  8) |  uuid->data[3];
2977           guid.data2 = (uuid->data[4] <<  8) |  uuid->data[5];
2978           guid.data3 = (uuid->data[6] <<  8) |  uuid->data[7];
2979           memcpy(guid.data4, &uuid->data[8], 8);
2980           proto_tree_add_guid(tree, hf_sip_session_id_remote_uuid, tvb,
2981                               equals_offset + 1, line_end_offset - equals_offset - 1, &guid);
```

Thank you