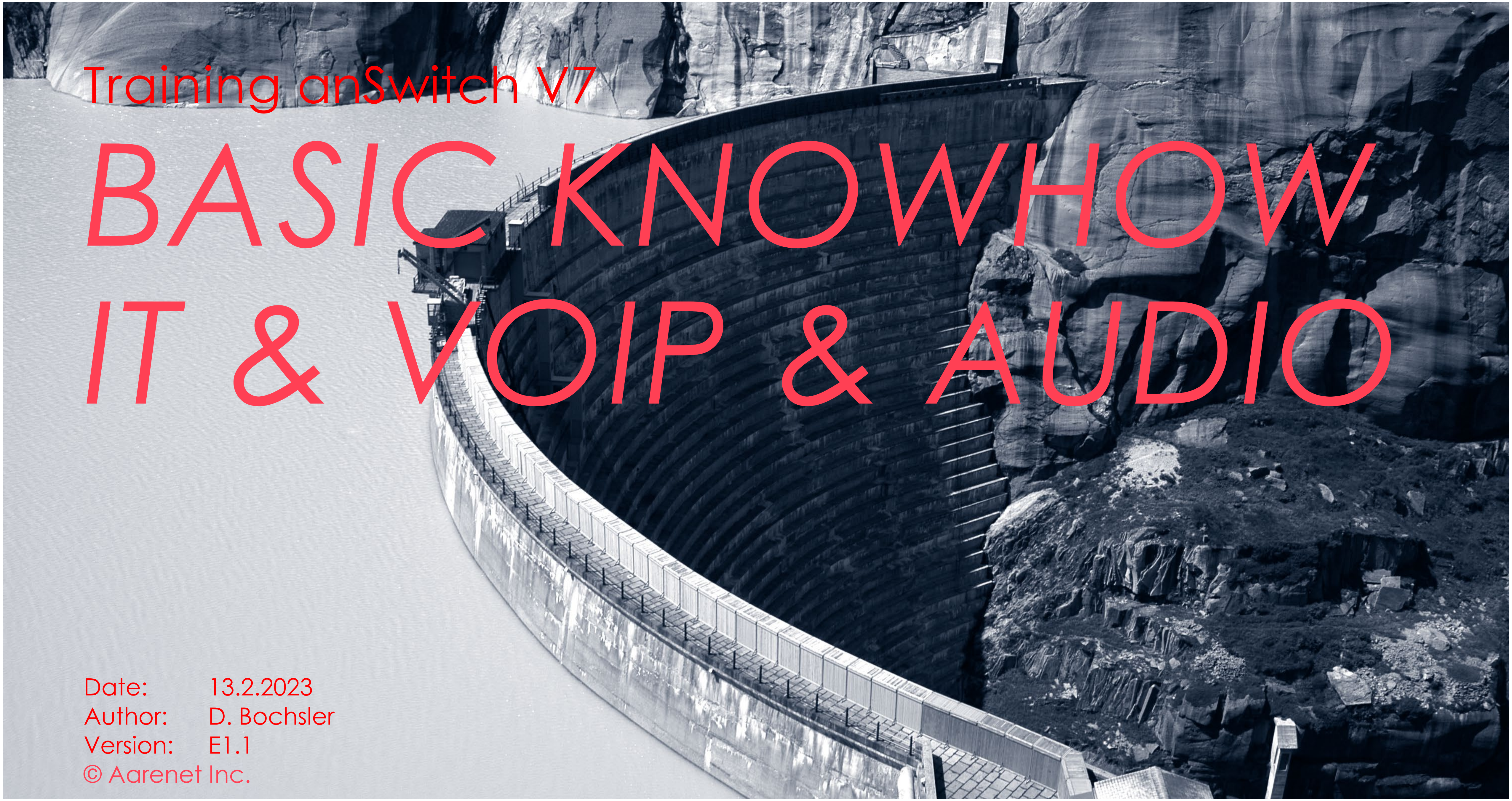


Training anSwitch V7

BASIC KNOWHOW IT & VOIP & AUDIO

Date: 13.2.2023
Author: D. Bochsler
Version: E1.1

INTRODUCTION & MOTIVATION

This training covers the topics:

- SIP & RTP Protocol Basics: SIP Dialogs, Messages, Headers, Flows, SDP, etc.
- Special IT network situations which can cause problems
- Audio transfer topics and transcoding

After this training, the trainee is enabled:

- To understand the SIP & SDP protocol
- To understand IT network problematics
- To understanding codec negotiation and transcoding

IT'S NOT MAGIC

IT'S **"KNOW HOW"**

TABLE OF CONTENTS

1 SIP & SDP: PROTOCOL BASICS

OVERVIEW SIP & SDP PROTOCOL

- The **Session Initiation Protocol SIP** is a communications protocol for signaling and controlling multimedia communication sessions. One of the most common applications of SIP is in Internet telephony for voice and video calls.
- The **Session Description Protocol SDP** is a format for describing streaming media communications parameters.
- The SIP & SDP protocol build the backbone of communication between SIP devices!

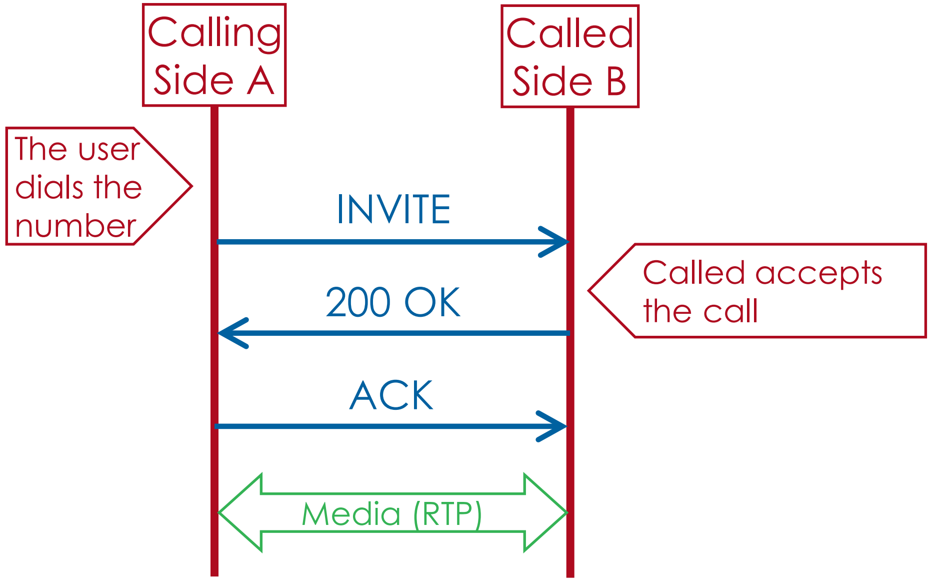
SIP & SDP PROTOCOL LINKS

- For an overview of the SIP & SDP protocol visit:
  - SIP: https://en.wikipedia.org/wiki/Session_Initiation_Protocol
  - SDP: https://en.wikipedia.org/wiki/Session_Description_Protocol

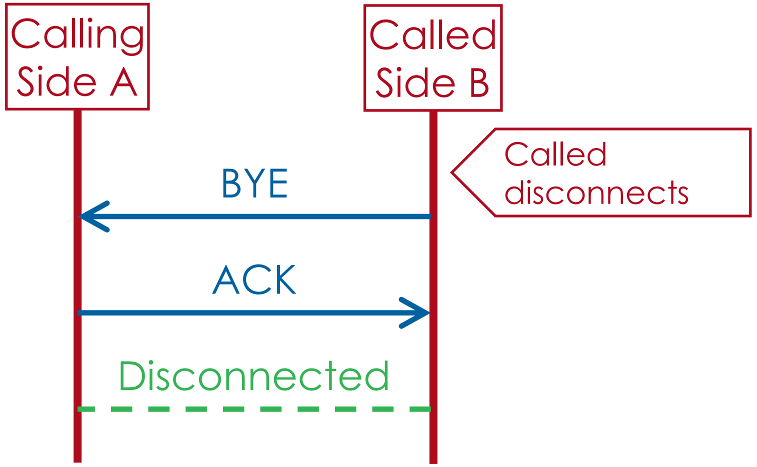
- For the SIP details you must consult the RFC!
  A good entry point is the "Hitchhiker's Guide to the Session Initiation Protocol (SIP)" with lists all relevant SIP RFC's with a short description:
  - http://www.rfcreader.com/#rfc5411

BASICS: "SIP DIALOG"

- SIP is a stateless protocol. It defines just "SIP Dialogs which are supervised.
  - Example of a "SIP Dialog" with the minimal needed messages for a connection setup or connection re-negotiation:

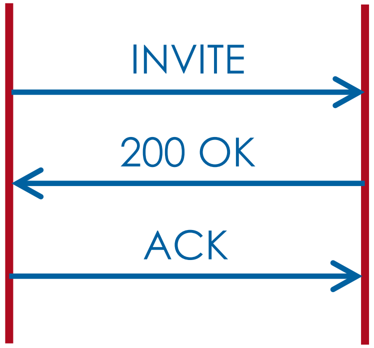


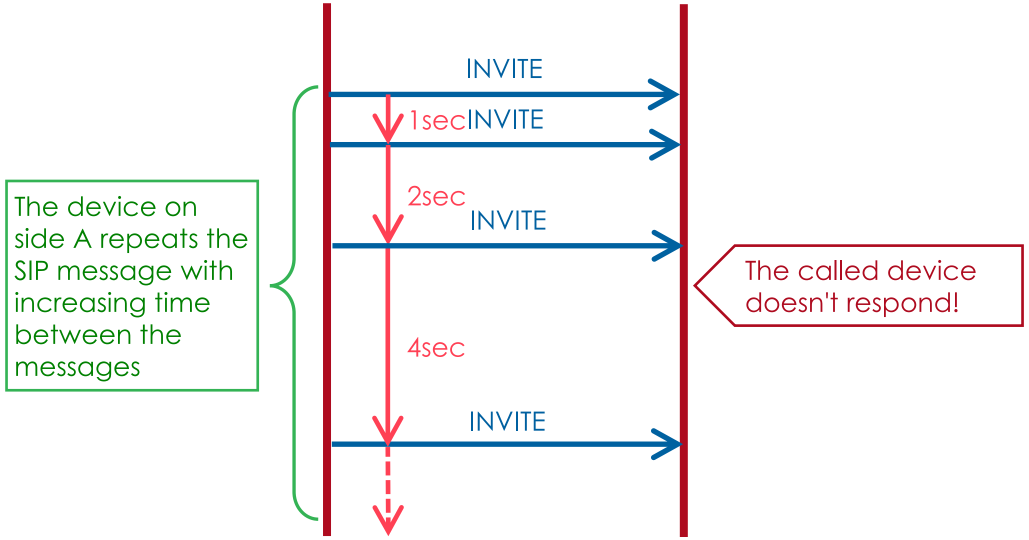
  - Example of a "SIP Dialog" with the minimal needed messages for a connection release:



BASICS: "SIP MESSAGE FLOW SUPERVISION"

- A SIP session is supervised within a "SIP dialog" reliably!
- In a SIP dialog "SIP Requests" must be acknowledged by the peer with well-defined Responses:
  - 1xx: Provisional responses to requests indicate the request was valid and is being processed.
  - 2xx: 200-level responses indicate a successful completion of the request. As a response to an INVITE, it indicates a call is established.
  - 3xx: This group indicates a redirection is needed for completion of the request. The request must be completed with a new destination.
  - 4xx: The request contained bad syntax or cannot be fulfilled at the server.
  - 5xx: The server failed to fulfill an apparently valid request.
  - 6xx: This is a global failure, as the request cannot be fulfilled at any server.
- A not acknowledged SIP message is repeated.
  After 2 – 8 failed retransmissions the session will usually be terminated!



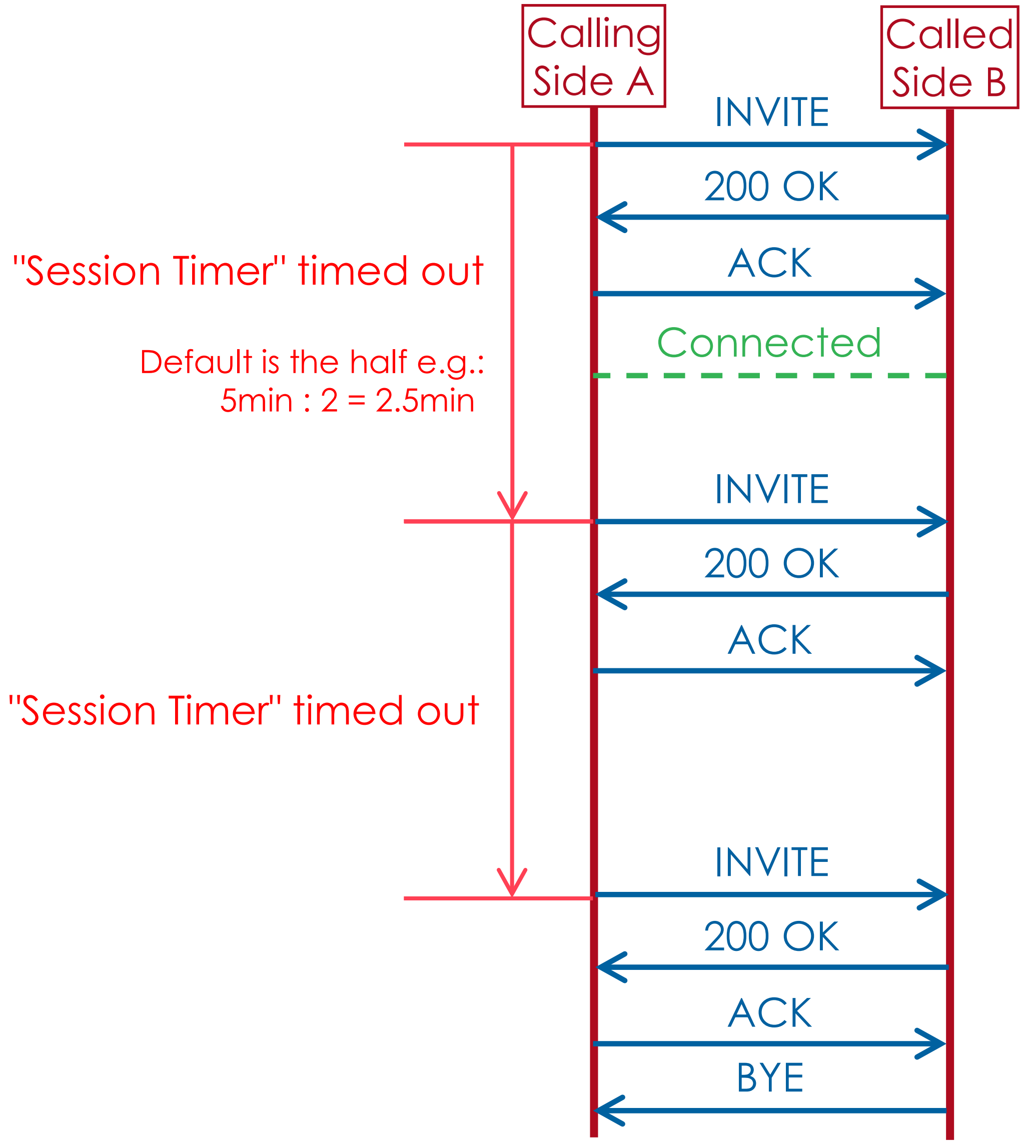


The device on side A repeats the SIP message with increasing time between the messages

The called device doesn't respond!

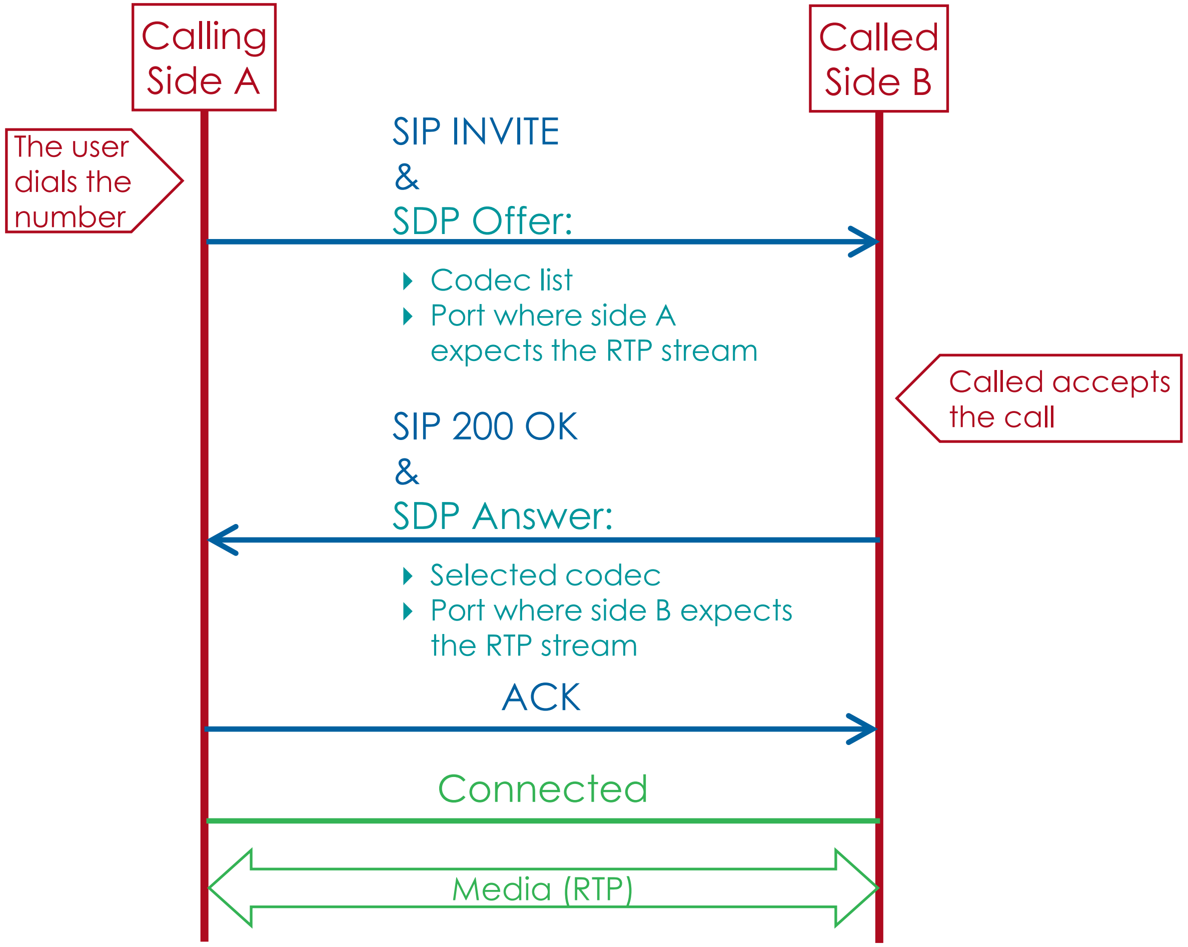
BASICS: "SIP CONNECTION SUPERVISION WITH SESSION TIMER"

- Originally, between the last ACK and the connection releasing BYE, SIP had no connection supervision.
  - ➔ So, it was not possible to supervise for a SIP phone if the peer is still in the connection.
  - ➔ "Hanging" calls were the result, creating high call charges.
- This deficit was eliminated by the introduction of "Session Timer":
  - The calling peer sets a "Session Timer", e.g. 5min.
  - Latest at the timeout of the session timer, the calling side repeats the INVITE
  - The calling side:
    Checks it the 200 OK is received, if none is received it releases the call.
  - The called side:
    Expects an INVITE, if none is received it releases the call.



BASICS: SESSION DESCRIPTION PROTOCOL SDP

- The SDP protocol is embedded in the SIP messages



EXAMPLE SIP MESSAGE FLOW OF A REGULAR OUTGOING CALL

- Example of a regular outgoing call toward the PSTN:

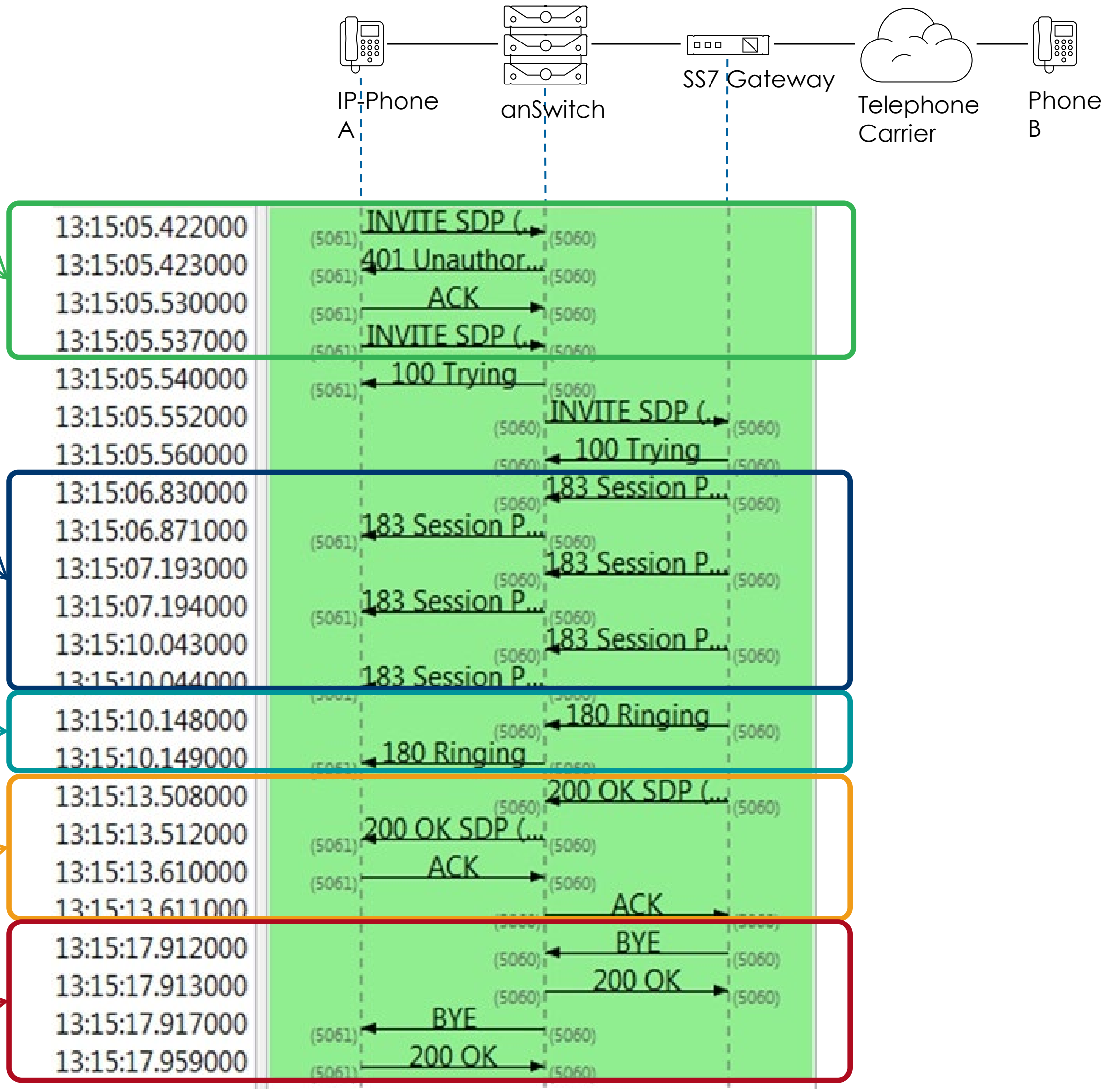
The first INVITE of is challenged! The device must repeat the INVITE with valid SIP credentials!

The repeated SESSION PROGRESS message from the PSTN indicates that the PSTN is engaged with routing the connection.

The PSTN has routed the connection and the peer device B is RINGING.

The peers are connected

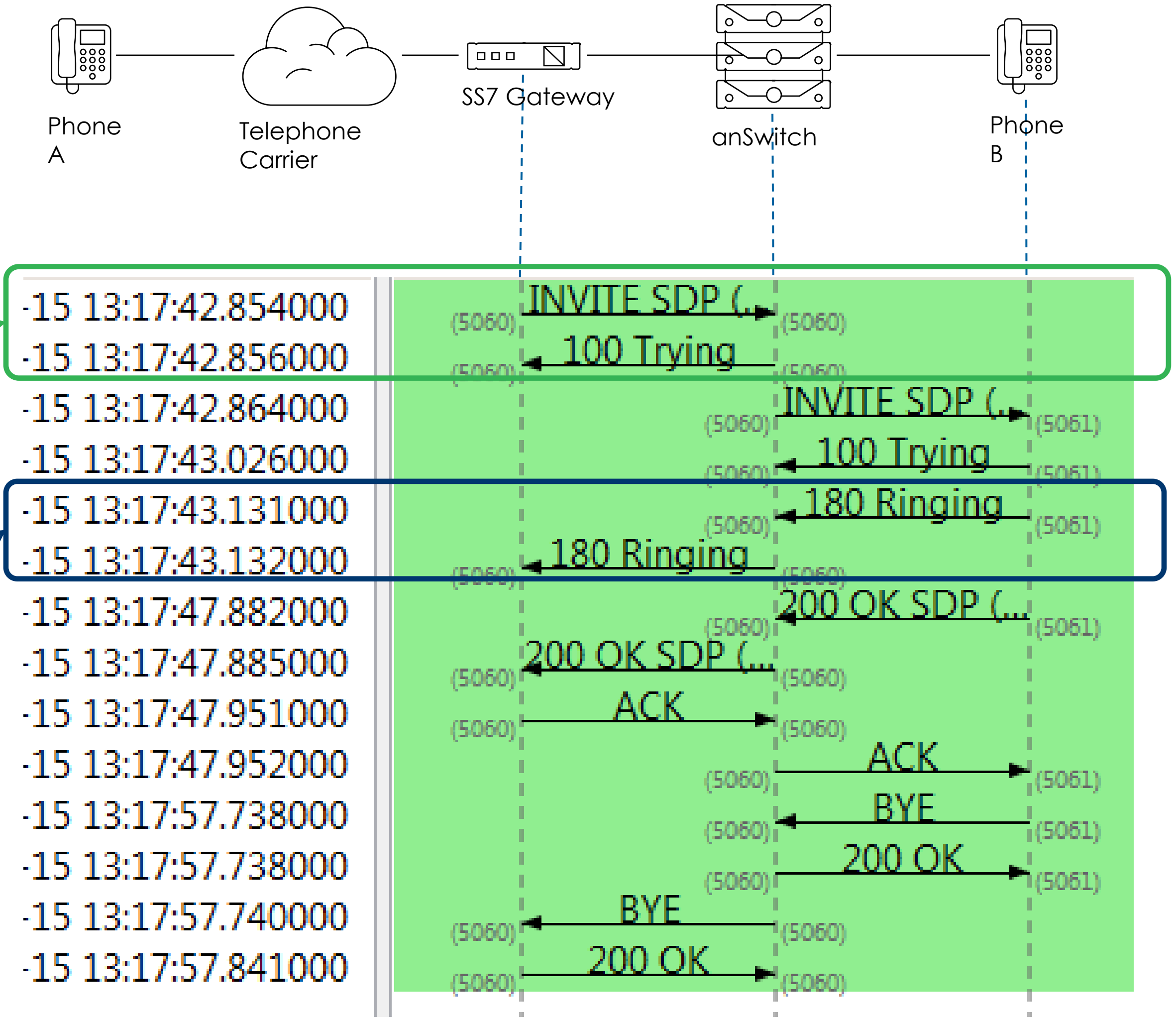
The peer B disconnects the call.



```
13:15:05.422000  (5061) INVITE SDP (...        (5060)
13:15:05.423000  (5061) 401 Unauthor...        (5060)
13:15:05.530000  (5061) ACK                    (5060)
13:15:05.537000  (5061) INVITE SDP (...        (5060)
13:15:05.540000  (5061) 100 Trying             (5060)
13:15:05.552000                         (5060) INVITE SDP (...  (5060)
13:15:05.560000                         (5060) 100 Trying       (5060)
13:15:06.830000                         (5060) 183 Session P... (5060)
13:15:06.871000  (5061) 183 Session P...       (5060)
13:15:07.193000                         (5060) 183 Session P... (5060)
13:15:07.194000  (5061) 183 Session P...       (5060)
13:15:10.043000                         (5060) 183 Session P... (5060)
13:15:10.044000         183 Session P...
13:15:10.148000                         (5060) 180 Ringing      (5060)
13:15:10.149000         180 Ringing
13:15:13.508000                         (5060) 200 OK SDP (...  (5060)
13:15:13.512000  (5061) 200 OK SDP (...        (5060)
13:15:13.610000  (5061) ACK                    (5060)
13:15:13.611000                                ACK
13:15:17.912000                         (5060) BYE              (5060)
13:15:17.913000                         (5060) 200 OK           (5060)
13:15:17.917000  (5061) BYE                    (5060)
13:15:17.959000  (5061) 200 OK                 (5060)
```

EXAMPLE SIP MESSAGE FLOW OF A REGULAR INCOMING CALL

- Example of a regular incoming call from the PSTN:



```
-15 13:17:42.854000   (5060) INVITE SDP (... ->    (5060)
-15 13:17:42.856000   (5060) <- 100 Trying         (5060)
-15 13:17:42.864000                        (5060) INVITE SDP (... ->  (5061)
-15 13:17:43.026000                        (5060) <- 100 Trying       (5061)
-15 13:17:43.131000                        (5060) <- 180 Ringing      (5061)
-15 13:17:43.132000   (5060) <- 180 Ringing        (5060)
-15 13:17:47.882000                        (5060) <- 200 OK SDP (...  (5061)
-15 13:17:47.885000   (5060) <- 200 OK SDP (...    (5060)
-15 13:17:47.951000   (5060) ACK ->                (5060)
-15 13:17:47.952000                        (5060) ACK ->              (5061)
-15 13:17:57.738000                        (5060) <- BYE              (5061)
-15 13:17:57.738000                        (5060) 200 OK ->           (5061)
-15 13:17:57.740000   (5060) <- BYE                (5060)
-15 13:17:57.841000   (5060) 200 OK ->             (5060)
```

Note:
The INVITE from a PSTN Gateway is not challenged!

The anSwitch has routed the connection and the other peer is RINGING.

EXAMPLE SIP MESSAGE FLOW EXCEPTIONAL SIGNALING SITUATIONS

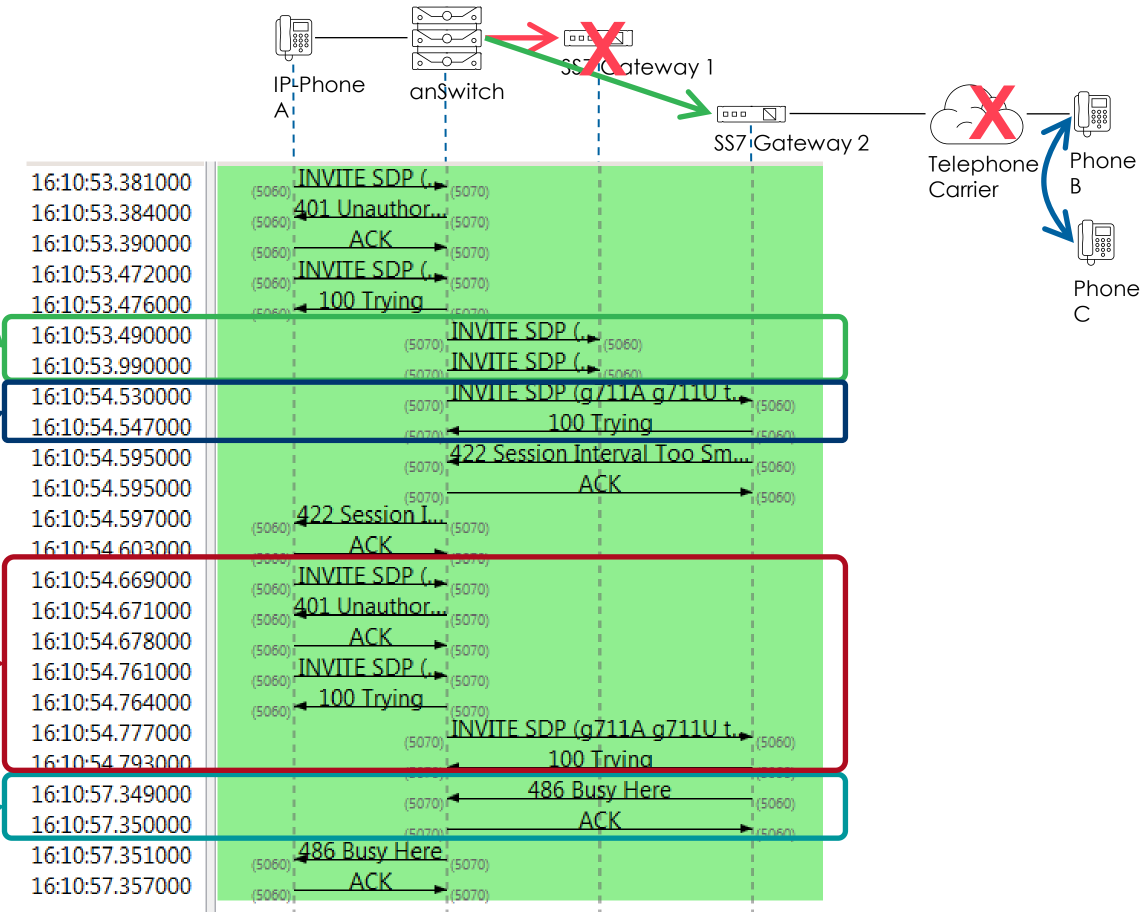
- Example of an outgoing call toward the PSTN with three exceptional signaling situations:
  1. The PSTN Gateway 1 doesn't respond!
     So, the anSwitch must re-route to the PSTN Gateway 2
  2. The telephone on side A offers an invalid "Session Time" value which is refused by the PSTN Gateway 2.
     The telephone on side A must do a re-INVITE with an acceptable "Session Time" value.
  3. End point B is busy with a connection to C.

The anSwitch tries to route the connection via Gateway 1.
→ The GW 1 doesn't respond

The first INVITE to the Gateway 2 is refused due to erroneous "Session Timer" value.

The subsequent re-INVITE to the Gateway 2 is accepted.

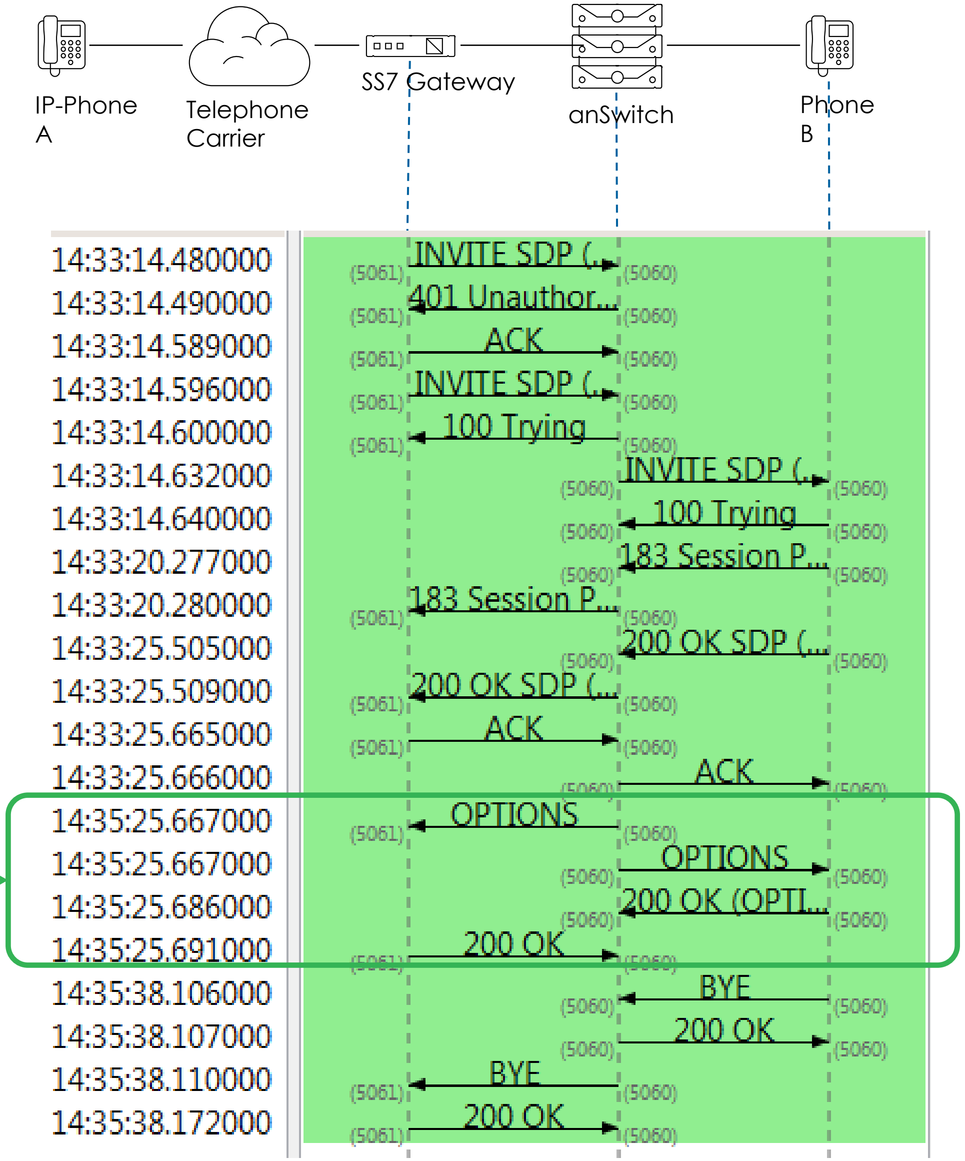
Phone B is busy with phone C. Therefore, the call is rejected with cause 486.



```
16:10:53.381000  (5060) INVITE SDP (...  (5070)
16:10:53.384000  (5060) 401 Unauthor...  (5070)
16:10:53.390000  (5060) ACK  (5070)
16:10:53.472000  (5060) INVITE SDP (...  (5070)
16:10:53.476000  (5060) 100 Trying  (5070)
16:10:53.490000  (5070) INVITE SDP (...  (5060)
16:10:53.990000  (5070) INVITE SDP (...  (5060)
16:10:54.530000  (5070) INVITE SDP (g711A g711U t...  (5060)
16:10:54.547000  (5070) 100 Trying  (5060)
16:10:54.595000  (5070) 422 Session Interval Too Sm...  (5060)
16:10:54.595000  (5070) ACK  (5060)
16:10:54.597000  (5060) 422 Session I...  (5070)
16:10:54.603000  (5060) ACK  (5070)
16:10:54.669000  (5060) INVITE SDP (...  (5070)
16:10:54.671000  (5060) 401 Unauthor...  (5070)
16:10:54.678000  (5060) ACK  (5070)
16:10:54.761000  (5060) INVITE SDP (...  (5070)
16:10:54.764000  (5060) 100 Trying  (5070)
16:10:54.777000  (5070) INVITE SDP (g711A g711U t...  (5060)
16:10:54.793000  (5070) 100 Trying  (5060)
16:10:57.349000  (5070) 486 Busy Here  (5060)
16:10:57.350000  (5070) ACK  (5060)
16:10:57.351000  (5060) 486 Busy Here  (5070)
16:10:57.357000  (5060) ACK  (5070)
```

EXAMPLES SIP MESSAGE FLOW OF OPTION MESSAGES

- Example of an OPTION messages from the anSwitch sent toward the SIP peers for checking their presence:

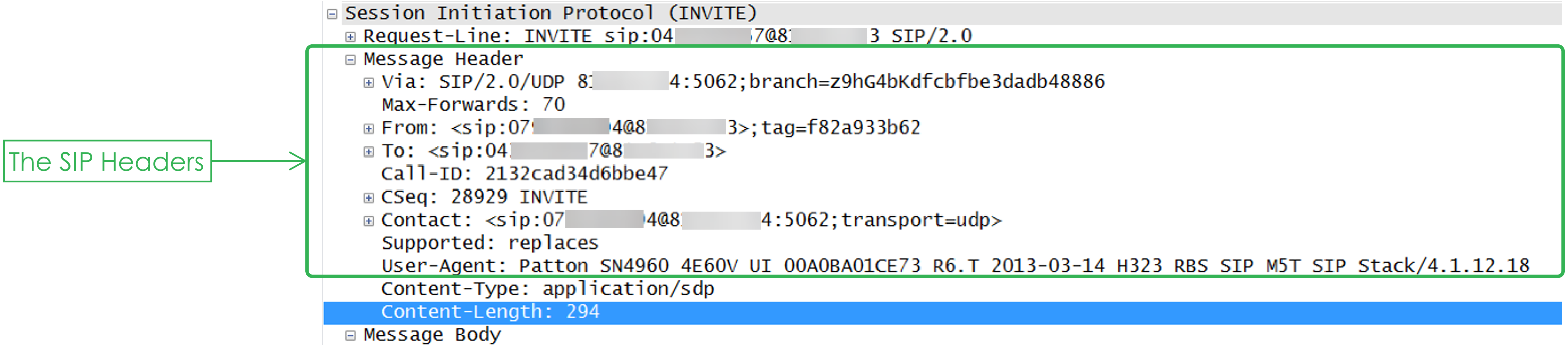


```
14:33:14.480000   (5061) INVITE SDP (...      (5060)
14:33:14.490000   (5061) 401 Unauthor...      (5060)
14:33:14.589000   (5061) ACK                  (5060)
14:33:14.596000   (5061) INVITE SDP (...      (5060)
14:33:14.600000   (5061) 100 Trying           (5060)
14:33:14.632000                        (5060) INVITE SDP (...   (5060)
14:33:14.640000                        (5060) 100 Trying        (5060)
14:33:20.277000                        (5060) 183 Session P...  (5060)
14:33:20.280000   (5061) 183 Session P...     (5060)
14:33:25.505000                        (5060) 200 OK SDP (...   (5060)
14:33:25.509000   (5061) 200 OK SDP (...      (5060)
14:33:25.665000   (5061) ACK                  (5060)
14:33:25.666000                        (5060) ACK               (5060)
14:35:25.667000   (5061) OPTIONS              (5060)
14:35:25.667000                        (5060) OPTIONS           (5060)
14:35:25.686000                        (5060) 200 OK (OPTI...   (5060)
14:35:25.691000   (5061) 200 OK               (5060)
14:35:38.106000                        (5060) BYE               (5060)
14:35:38.107000                        (5060) 200 OK            (5060)
14:35:38.110000   (5061) BYE                  (5060)
14:35:38.172000   (5061) 200 OK               (5060)
```

The anSwitch checks the presence of the end points OPTION messages.

OVERVIEW SIP HEADER

- The SIP Header in a SIP message contains the information that are needed for establishing, maintain and end a connection.



```
⊟ Session Initiation Protocol (INVITE)
  ⊞ Request-Line: INVITE sip:04[redacted]7@8[redacted]3 SIP/2.0
  ⊟ Message Header
    ⊞ Via: SIP/2.0/UDP 8[redacted]4:5062;branch=z9hG4bKdfcbfbe3dadb48886
      Max-Forwards: 70
    ⊞ From: <sip:07[redacted]4@8[redacted]3>;tag=f82a933b62
    ⊞ To: <sip:04[redacted]7@8[redacted]3>
      Call-ID: 2132cad34d6bbe47
    ⊞ CSeq: 28929 INVITE
    ⊞ Contact: <sip:07[redacted]4@8[redacted]4:5062;transport=udp>
      Supported: replaces
      User-Agent: Patton SN4960 4E60V UI 00A0BA01CE73 R6.T 2013-03-14 H323 RBS SIP M5T SIP Stack/4.1.12.18
      Content-Type: application/sdp
      Content-Length: 294
  ⊟ Message Body
```

The SIP Headers

Note	The order of the SIP headers within a SIP Message is not important.

MANDATORY AND IMPORTANT SIP HEADER

- A list of the most important SIP Headers in a SIP Request:

Request Line: → Mandatory

Via: → Mandatory
The VIA header keeps track of all the proxies a request has traversed.

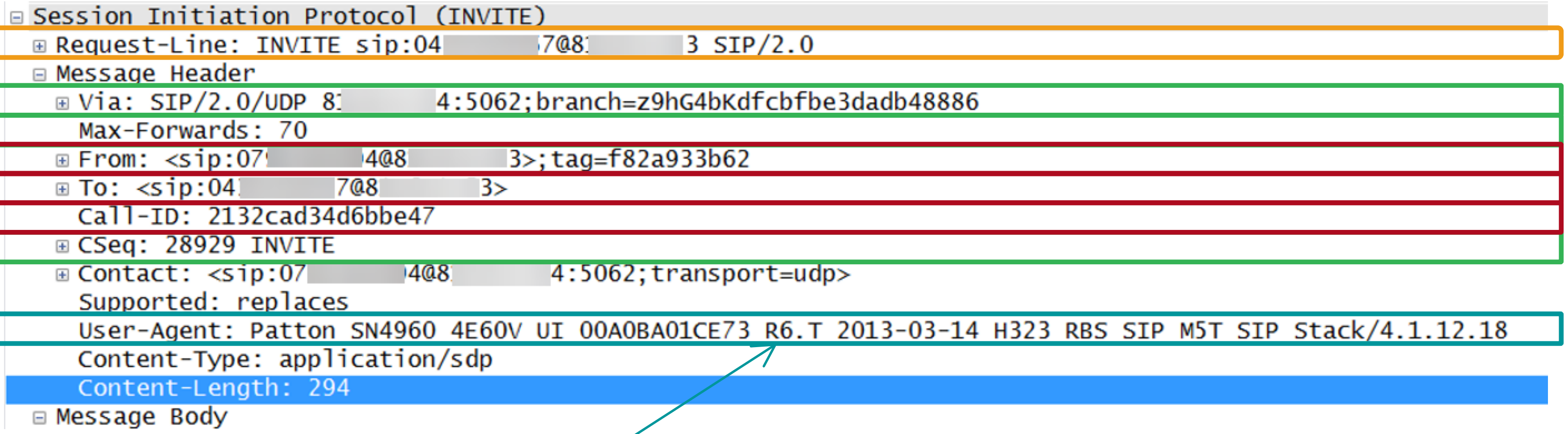
Max-Forwards: → Mandatory
The Max-Forward header is used to avoid routing loops.

From: → Mandatory
The From header contains the URI of the originator of the request.

To: → Mandatory
The To header contains the URI of the destination of the request.

Call-ID: → Mandatory
The Call-ID header provides a unique identifiers for a SIP message exchange.

CSeq: → Mandatory
The Cseq header contains a sequence number and a method name. They are used to match requests and responses.

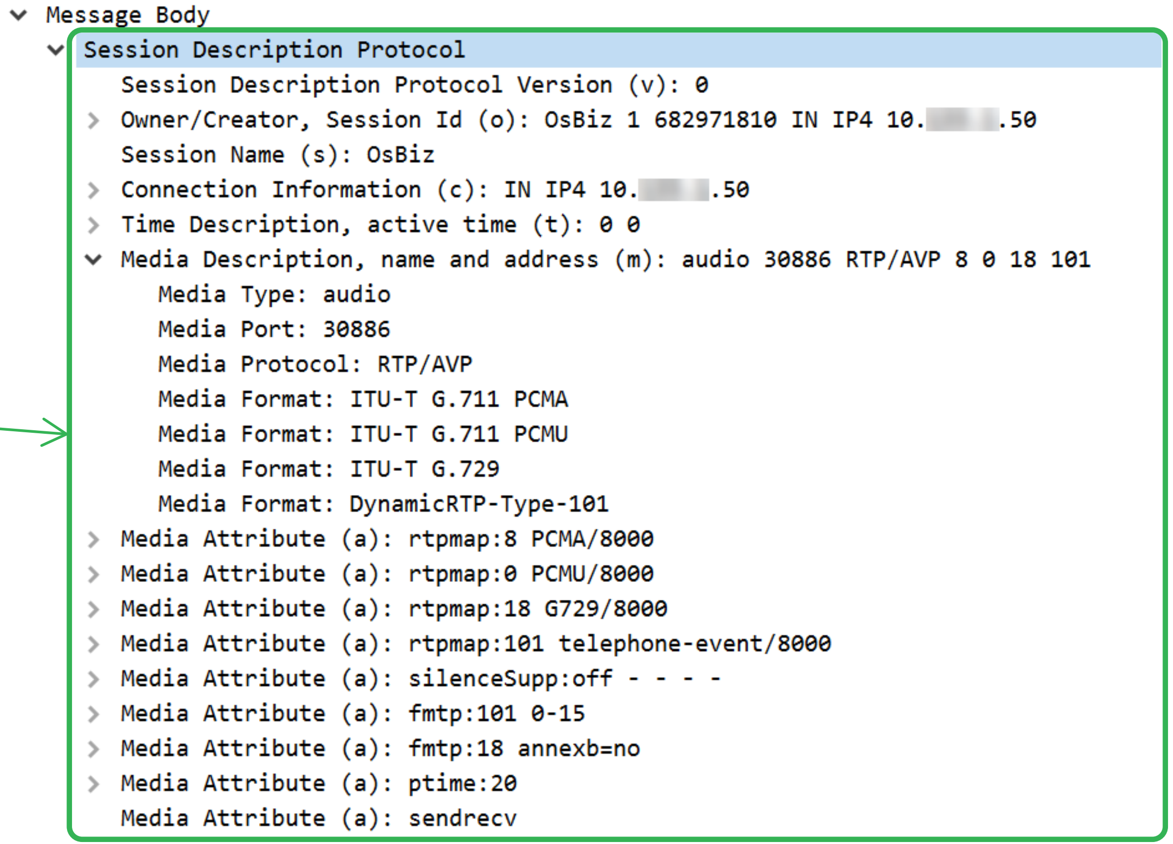


```
⊟ Session Initiation Protocol (INVITE)
  ⊞ Request-Line: INVITE sip:04[redacted]7@8[redacted]3 SIP/2.0
  ⊟ Message Header
    ⊞ Via: SIP/2.0/UDP 8[redacted]4:5062;branch=z9hG4bKdfcbfbe3dadb48886
      Max-Forwards: 70
    ⊞ From: <sip:07[redacted]4@8[redacted]3>;tag=f82a933b62
    ⊞ To: <sip:04[redacted]7@8[redacted]3>
      Call-ID: 2132cad34d6bbe47
    ⊞ CSeq: 28929 INVITE
    ⊞ Contact: <sip:07[redacted]4@8[redacted]4:5062;transport=udp>
      Supported: replaces
      User-Agent: Patton SN4960 4E60V UI 00A0BA01CE73 R6.T 2013-03-14 H323 RBS SIP M5T SIP Stack/4.1.12.18
      Content-Type: application/sdp
      Content-Length: 294
  ⊟ Message Body
```

User-Agent:
The User-Agent header contains information about the SIP device.

OVERVIEW OF THE SDP DESCRIPTION

- The SDP Description is embedded in the "Message Body" of a SIP message.
- The SDP Description contains the information that are needed for establishing, maintain and end a RTP media stream.

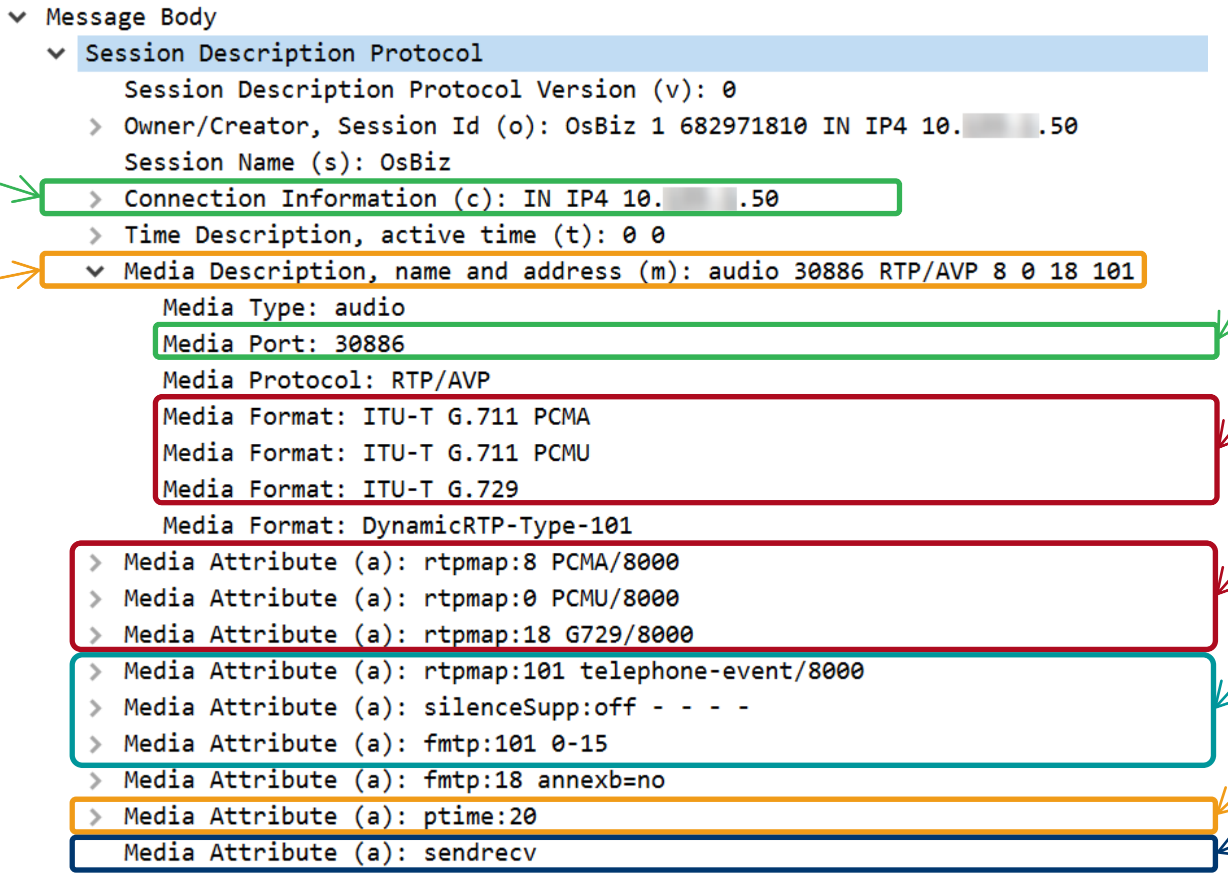


```
⌄ Message Body
  ⌄ Session Description Protocol
      Session Description Protocol Version (v): 0
    > Owner/Creator, Session Id (o): OsBiz 1 682971810 IN IP4 10.[redacted].50
      Session Name (s): OsBiz
    > Connection Information (c): IN IP4 10.[redacted].50
    > Time Description, active time (t): 0 0
    ⌄ Media Description, name and address (m): audio 30886 RTP/AVP 8 0 18 101
        Media Type: audio
        Media Port: 30886
        Media Protocol: RTP/AVP
        Media Format: ITU-T G.711 PCMA
        Media Format: ITU-T G.711 PCMU
        Media Format: ITU-T G.729
        Media Format: DynamicRTP-Type-101
    > Media Attribute (a): rtpmap:8 PCMA/8000
    > Media Attribute (a): rtpmap:0 PCMU/8000
    > Media Attribute (a): rtpmap:18 G729/8000
    > Media Attribute (a): rtpmap:101 telephone-event/8000
    > Media Attribute (a): silenceSupp:off - - - -
    > Media Attribute (a): fmtp:101 0-15
    > Media Attribute (a): fmtp:18 annexb=no
    > Media Attribute (a): ptime:20
      Media Attribute (a): sendrecv
```

The SDP protocol is embedded in the SIP message

INFORMATION IN THE SDP DESCRIPTION

- The SDP Description detailed:
  - Specification of the RTP session: v-, o-, s- and c- lines
  - Time description: t- line
  - Media description: m- and a- lines



```
⌄ Message Body
  ⌄ Session Description Protocol
      Session Description Protocol Version (v): 0
    › Owner/Creator, Session Id (o): OsBiz 1 682971810 IN IP4 10.[redacted].50
      Session Name (s): OsBiz
    › Connection Information (c): IN IP4 10.[redacted].50
    › Time Description, active time (t): 0 0
    ⌄ Media Description, name and address (m): audio 30886 RTP/AVP 8 0 18 101
        Media Type: audio
        Media Port: 30886
        Media Protocol: RTP/AVP
        Media Format: ITU-T G.711 PCMA
        Media Format: ITU-T G.711 PCMU
        Media Format: ITU-T G.729
        Media Format: DynamicRTP-Type-101
    › Media Attribute (a): rtpmap:8 PCMA/8000
    › Media Attribute (a): rtpmap:0 PCMU/8000
    › Media Attribute (a): rtpmap:18 G729/8000
    › Media Attribute (a): rtpmap:101 telephone-event/8000
    › Media Attribute (a): silenceSupp:off - - - -
    › Media Attribute (a): fmtp:101 0-15
    › Media Attribute (a): fmtp:18 annexb=no
    › Media Attribute (a): ptime:20
      Media Attribute (a): sendrecv
```

IP address where the RTP audio stream is expected

Summary of RTP data

UDP port where the RTP audio stream is expected

List of offered or accepted codec for voice transfer. Codec priority:
- First preferred

Voice codec details

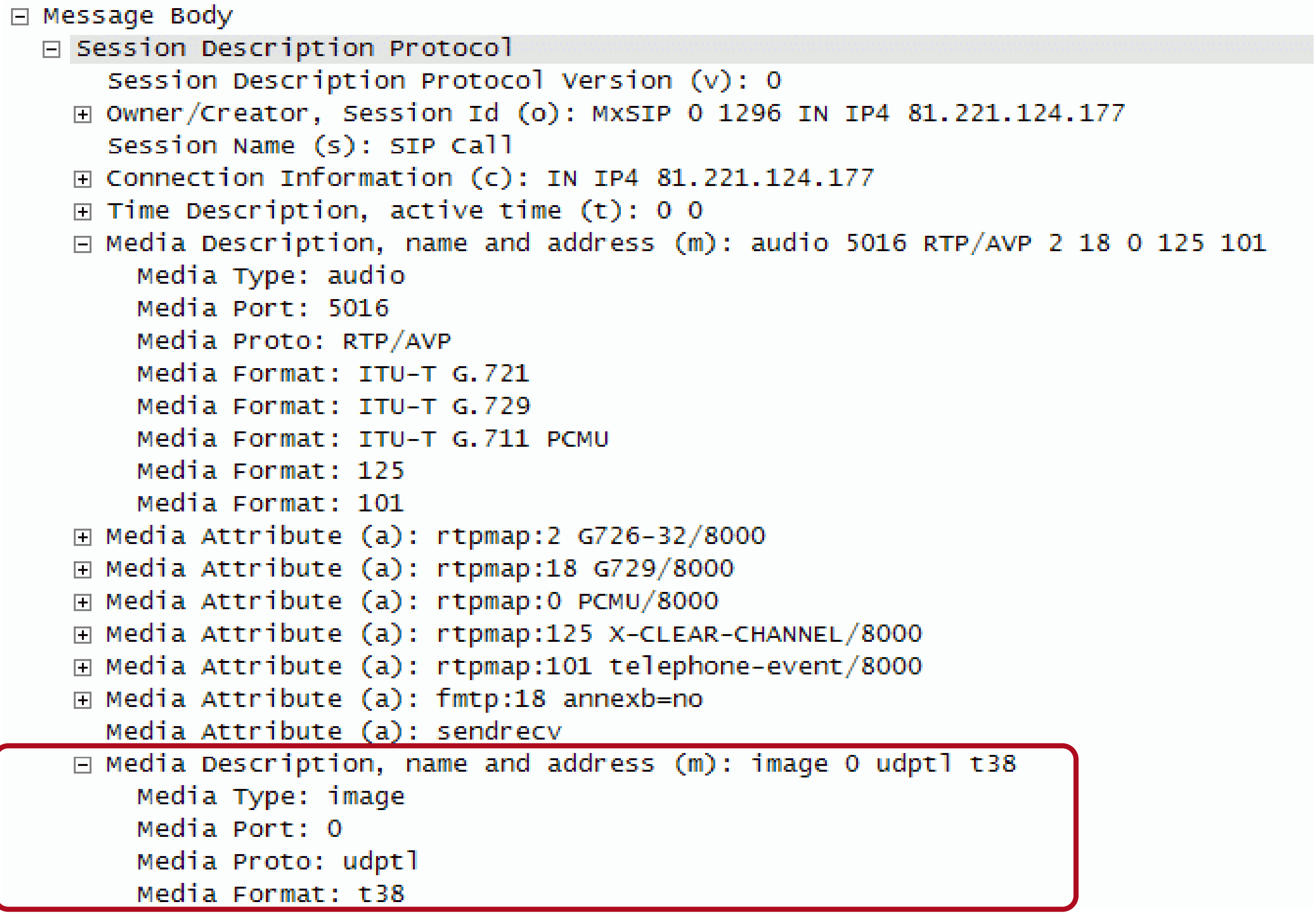
The "codec 101" for DTMF transfer, according RFC 2833

For this call leg the packet time is 20ms.

Directive to send and/or receive RTP packets

SDP DESCRIPTION FOR FAX T.38

- The SDP Description FAX transmission with T.38:



```
⊟ Message Body
  ⊟ Session Description Protocol
      Session Description Protocol Version (v): 0
    ⊞ Owner/Creator, Session Id (o): MxSIP 0 1296 IN IP4 81.221.124.177
      Session Name (s): SIP Call
    ⊞ Connection Information (c): IN IP4 81.221.124.177
    ⊞ Time Description, active time (t): 0 0
    ⊟ Media Description, name and address (m): audio 5016 RTP/AVP 2 18 0 125 101
        Media Type: audio
        Media Port: 5016
        Media Proto: RTP/AVP
        Media Format: ITU-T G.721
        Media Format: ITU-T G.729
        Media Format: ITU-T G.711 PCMU
        Media Format: 125
        Media Format: 101
    ⊞ Media Attribute (a): rtpmap:2 G726-32/8000
    ⊞ Media Attribute (a): rtpmap:18 G729/8000
    ⊞ Media Attribute (a): rtpmap:0 PCMU/8000
    ⊞ Media Attribute (a): rtpmap:125 X-CLEAR-CHANNEL/8000
    ⊞ Media Attribute (a): rtpmap:101 telephone-event/8000
    ⊞ Media Attribute (a): fmtp:18 annexb=no
      Media Attribute (a): sendrecv
    ⊟ Media Description, name and address (m): image 0 udptl t38
        Media Type: image
        Media Port: 0
        Media Proto: udptl
        Media Format: t38
```

Fax transfer T.38

→ If a Fax transfer is offered with T.38 then always it is always executed with T.38! No fall back or in-band negotiation with G.711 codec takes place.

2 SIP SESSION TIMER

OVERVIEW SIP SESSION TIMER

- The SIP does not define a keepalive mechanism for the sessions it establishes.
  - User agents, e.g. SIP phones, may be able to determine whether the session has timed out by using session specific mechanisms, e.g. by sending re-INVITE.
  - Proxies, e.g. the anSwitch, will not be able to do so. The result is that the anSwitch will not always be able to determine whether a session is still active.

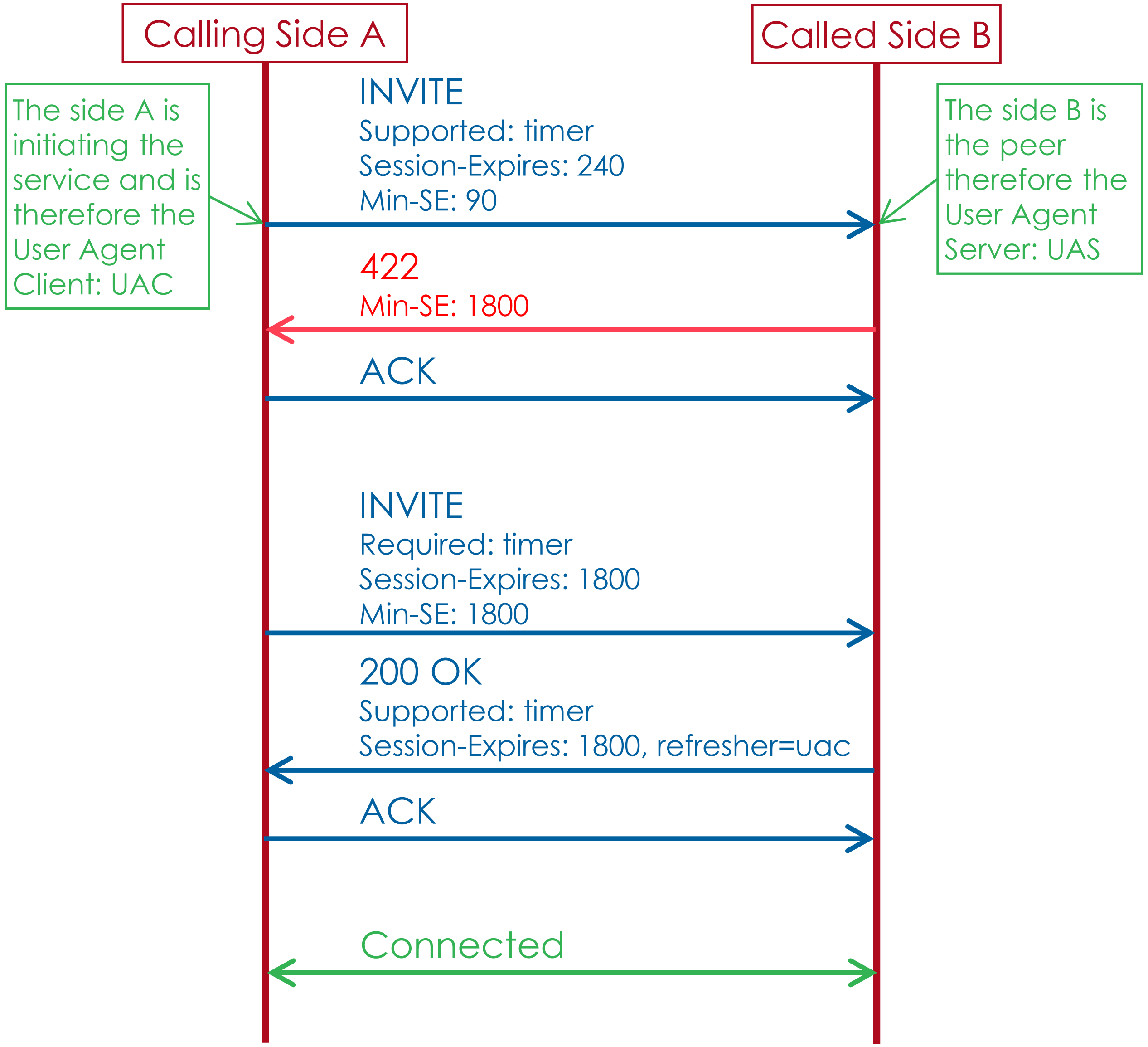
    Example:

    When a SIP phone fails to send a BYE message at the end of a session, or when the BYE message gets lost due to network problems, a call stateful proxy will not know when the session has ended. In this situation, the call stateful proxy will retain state for the call and has no method to determine when the call state information no longer applies.

➔ To resolve this problem, the Session Timer defines a keepalive mechanism for SIP sessions. User agents UAs send periodic re-INVITE or UPDATE requests to keep the session alive (details see RFC4028).

INITIATING SESSION TIMER

1. The calling side A requests a session timer by including Session-Expires:
   - Header Supported: timer
     - The calling side supports session timer.
   - Header Session-Expires: 240
     - The calling side defines 240sec as upper bound of for the session refresh interval.
   - Header Min-SE: 90
     - Defines the lower bound of the session refresh interval. The minimal value is 90sec. If this header is not provided, then the default is 90sec.
2. The called side B may reject the session if the refreshing interval is too short. It will add the desired minimal value:
   - Header Min-SE: 1800
3. The calling side A re-send the INVITE with adjusted values.
4. The called side B acknowledges and adds who is responsible for the refreshing:
   - Side B defines side A as the responsible:
     - User Agent Client: UAC



Calling Side A

Called Side B

The side A is initiating the service and is therefore the User Agent Client: UAC

The side B is the peer therefore the User Agent Server: UAS

INVITE
Supported: timer
Session-Expires: 240
Min-SE: 90

422
Min-SE: 1800

ACK

INVITE
Required: timer
Session-Expires: 1800
Min-SE: 1800

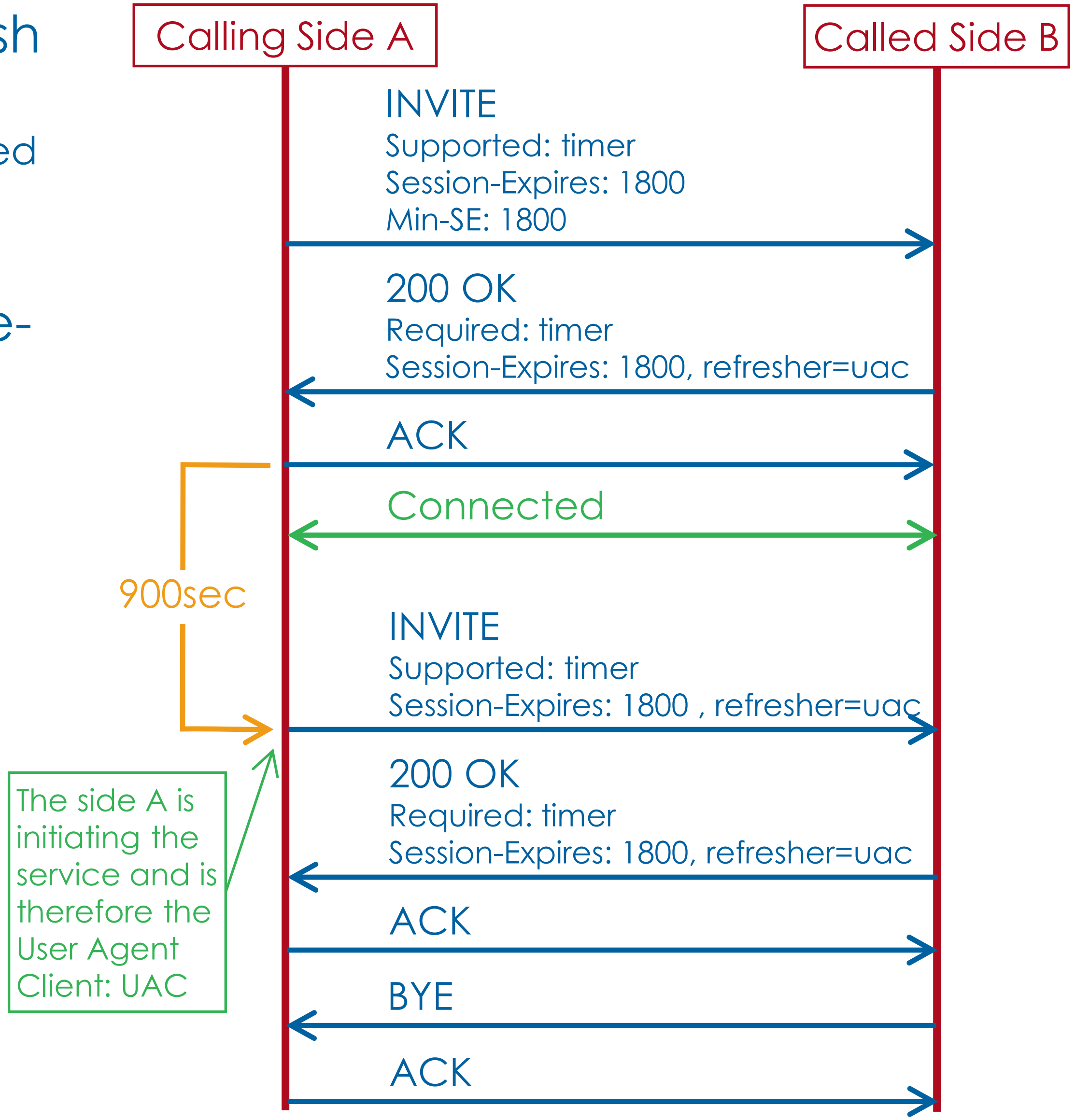
200 OK
Supported: timer
Session-Expires: 1800, refresher=uac

ACK

Connected

REFRESHING SESSION TIMER

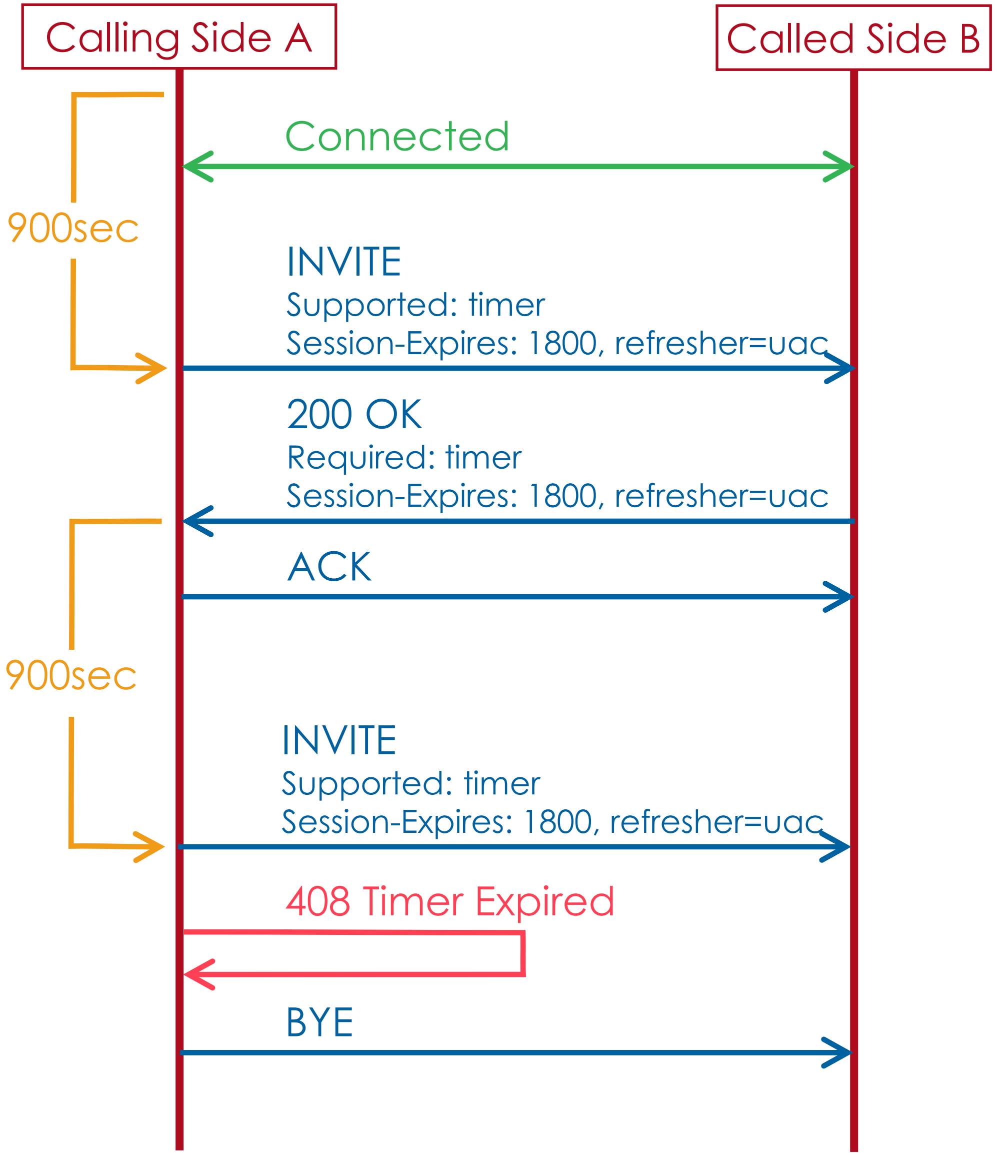
1. The UAC side A starts 900sec session refresh timer:
   - The RFC4028 recommends the half of the negotiated session expiry duration e.g.: 1800sec/2=900sec
2. After 900sec side A sends the refreshing re-INVITE.
3. The refreshing is repeated until one side ends the session.


Calling Side A
Called Side B
INVITE
Supported: timer
Session-Expires: 1800
Min-SE: 1800
200 OK
Required: timer
Session-Expires: 1800, refresher=uac
ACK
Connected
900sec
INVITE
Supported: timer
Session-Expires: 1800 , refresher=uac
200 OK
Required: timer
Session-Expires: 1800, refresher=uac
ACK
BYE
ACK
The side A is initiating the service and is therefore the User Agent Client: UAC

FAILED REFRESHING SESSION TIMER TERMINATES THE SESSION

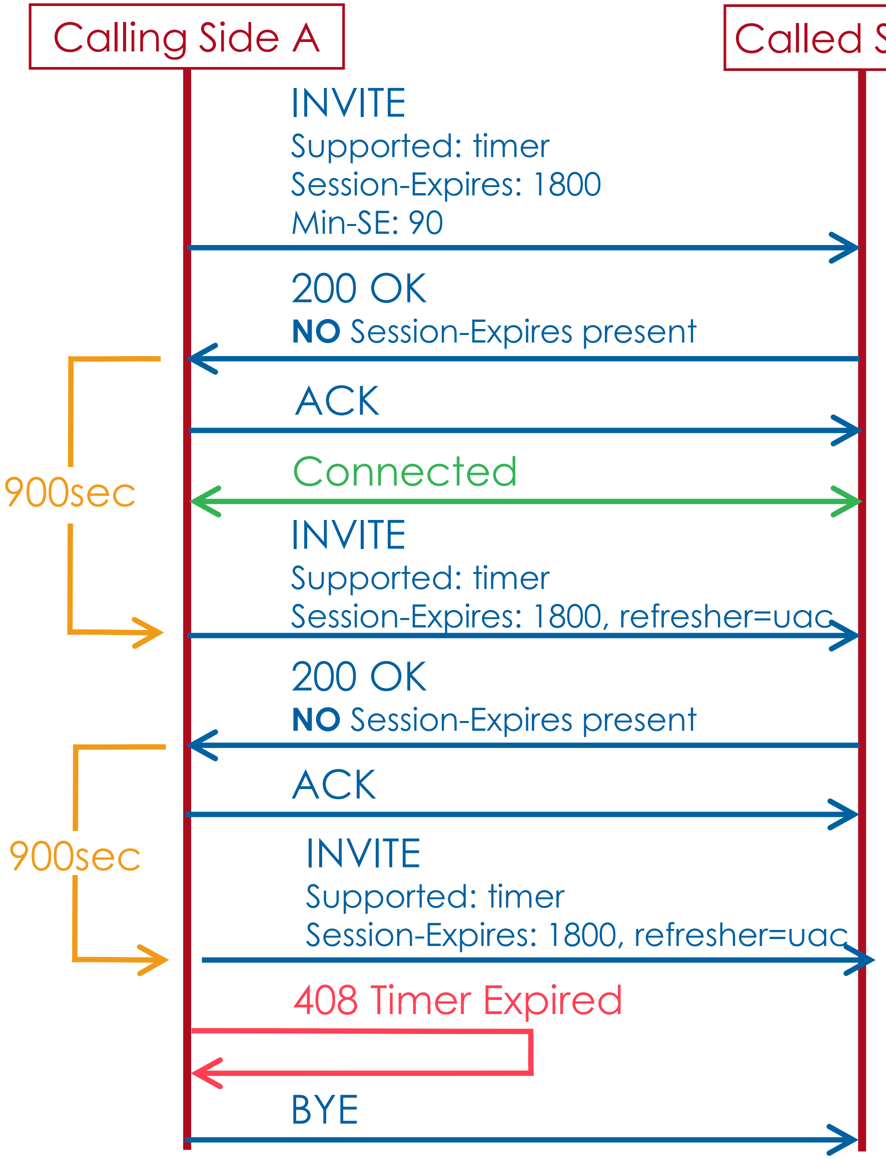
1. After 900sec side A sends the refreshing re-INVITE.

2. The side B crashed. The side A will receive a 408 Timer Expired and will send a BYE and the call is terminated.



SIDE A INITIATES SESSION TIMER BUT SIDE B DOESN'T SUPPORT IT

1. The calling side A requests a session timer.
2. The called side B acknowledges without a Session-Expires header.
3. Side A declares itself as refresher UAC and repeats the re-INVITE until the connection is released gracefully.
4. When side A receives a 408 Timer Expired it will send a BYE and the call is terminated.



SIDE B IS UAS AND ENFORCES SESSION TIMER

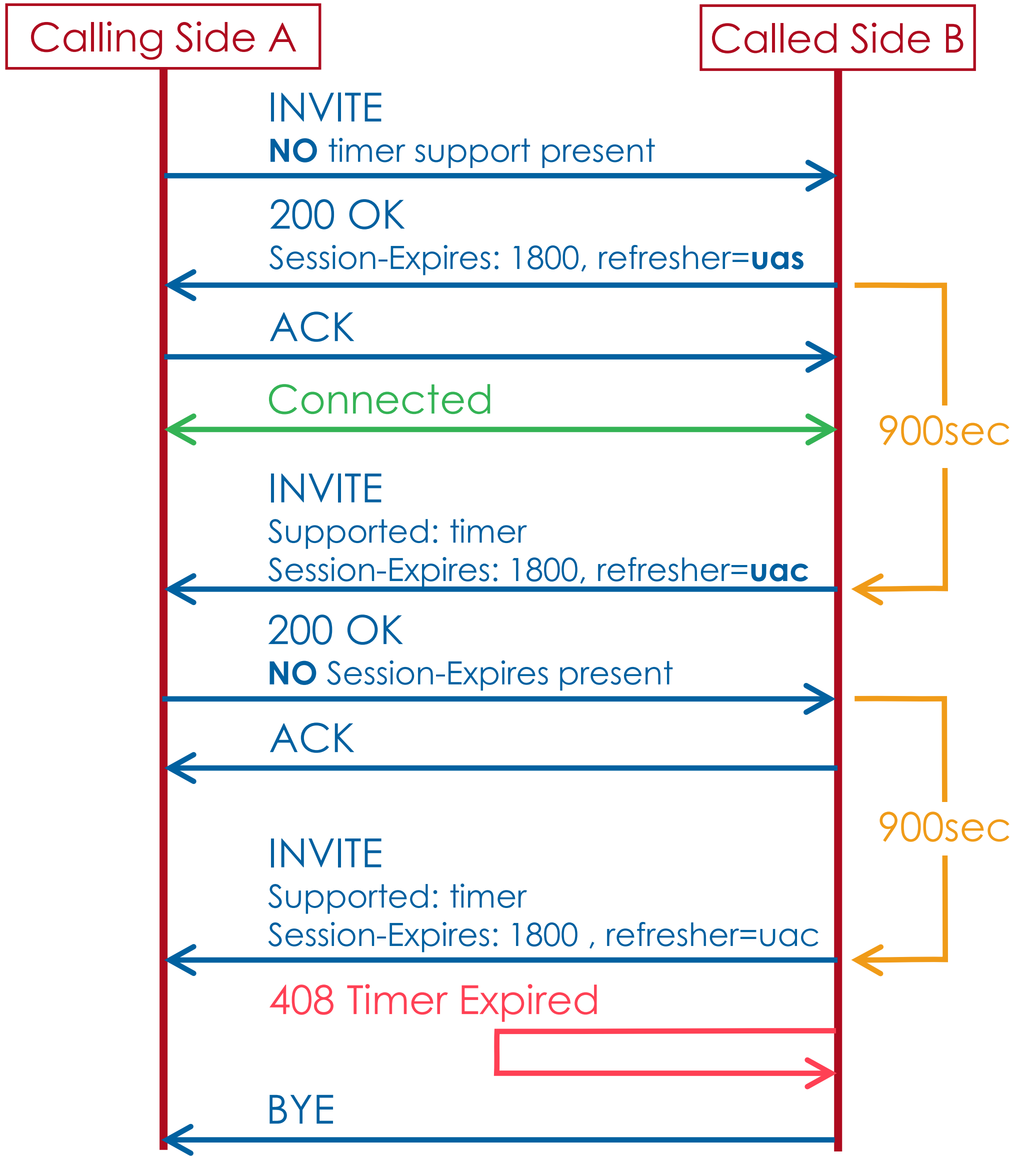
1. The calling side A initiates a session without session timer.

2. The called side B is configured to enforce session timer. It acknowledges with a session timer 1800 and declares itself as refresher UAS.

3. Side B starts the refreshing re-INVITE and declares itself as the actual refresher UAC.

   Note: The role of UAS and UAC changed because side B sent the re-INVITE. This ensures that Side B always performs the refresh.

4. When side B receives a 408 Timer Expired it will send a BYE and the call is terminated.


Calling Side A
Called Side B
INVITE
NO timer support present
200 OK
Session-Expires: 1800, refresher=**uas**
ACK
Connected
900sec
INVITE
Supported: timer
Session-Expires: 1800, refresher=**uac**
200 OK
NO Session-Expires present
ACK
900sec
INVITE
Supported: timer
Session-Expires: 1800 , refresher=uac
408 Timer Expired
BYE

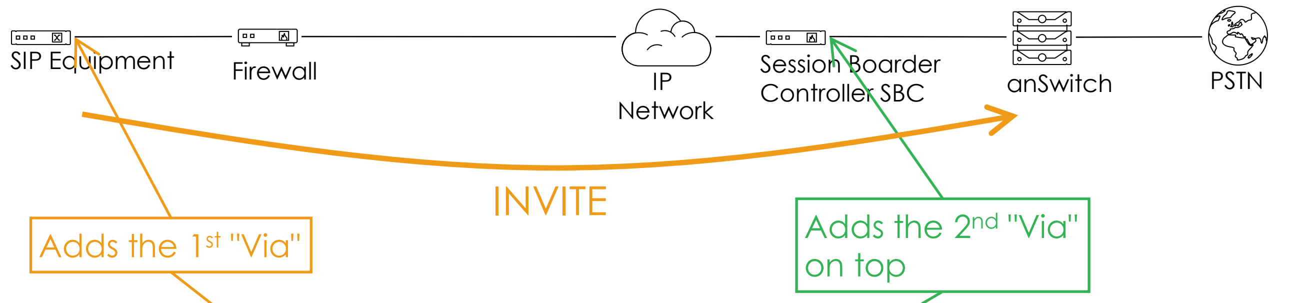
OTHER ALTERNATIVE SESSION TIMER HANDLING

- Check the RFC4028 for some other session timer scenarios with different initial UAS and UAC combinations.

3 SIP HEADER SPECIALS

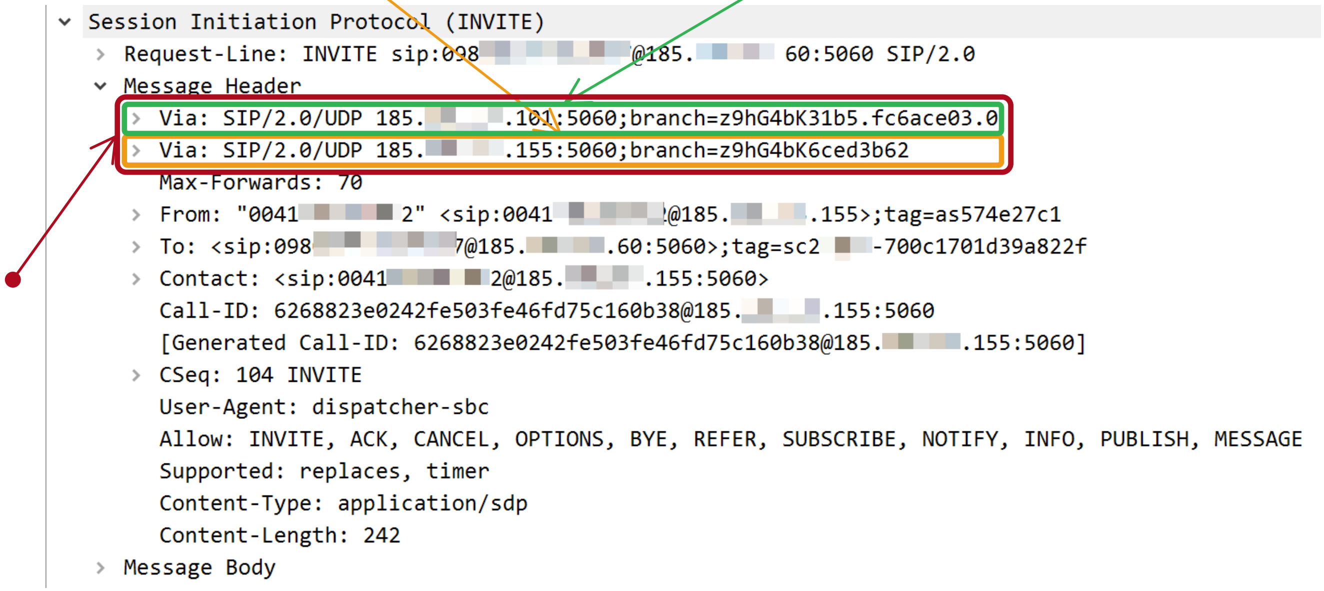
SIP-HEADER – UNDERSTANDING "VIA"

- Every proxy SIP device must add its own "Via" header before sending a SIP request.
- If there are already "Via" header in the message, the SIP device adds its new one at the top of the list before sending it to the next hop.
- The "Via" information allows the recipient of the request e.g., anSwitch, to return SIP responses to the correct device:
    - "Via":
      Via header identifies the protocol name (SIP), its version (2.0), transport type (e.g.: UDP or TCP), IP address of the SIP equipment, and the protocol port (typically 5060) used for the request.



Adds the 1st "Via"

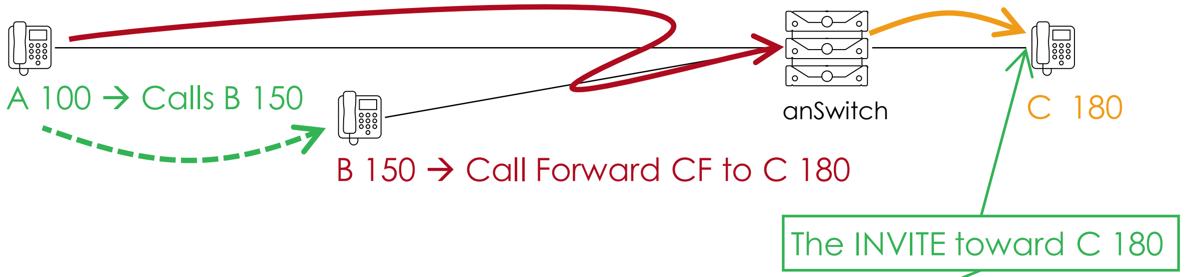
Adds the 2nd "Via" on top

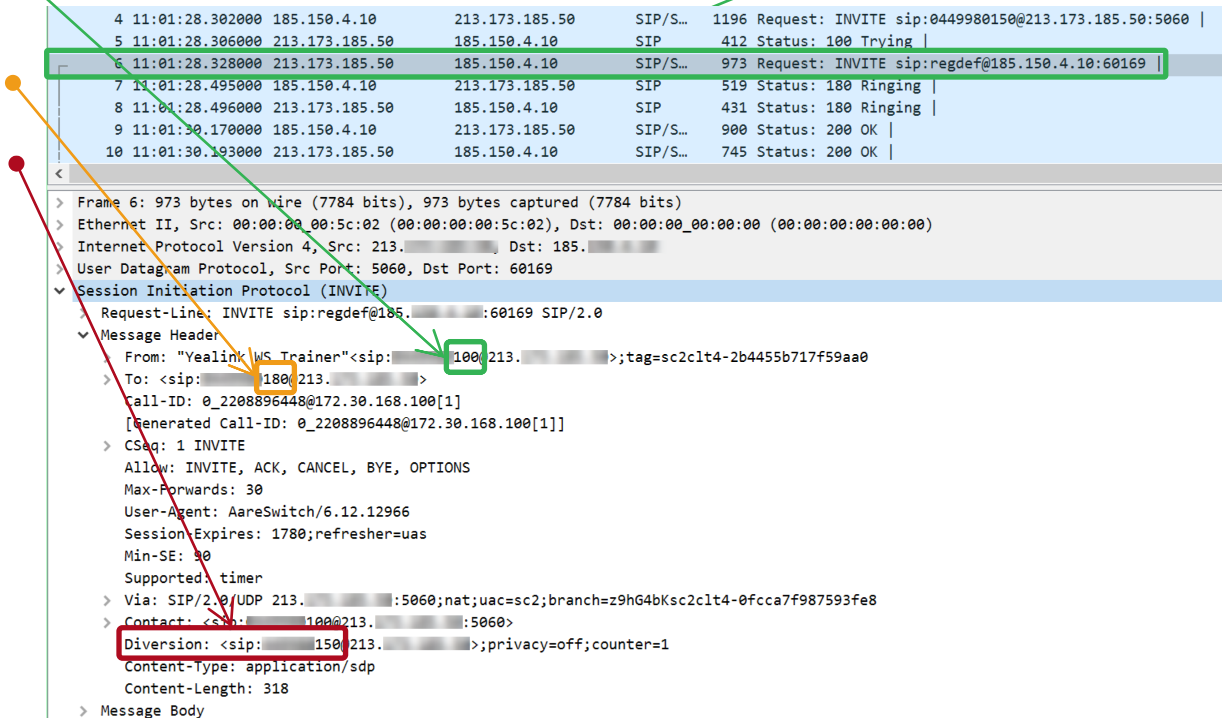


```
⌄ Session Initiation Protocol (INVITE)
  › Request-Line: INVITE sip:098[redacted]@185.[redacted]60:5060 SIP/2.0
  ⌄ Message Header
    › Via: SIP/2.0/UDP 185.[redacted].101:5060;branch=z9hG4bK31b5.fc6ace03.0
    › Via: SIP/2.0/UDP 185.[redacted].155:5060;branch=z9hG4bK6ced3b62
      Max-Forwards: 70
    › From: "0041[redacted]2" <sip:0041[redacted]@185.[redacted].155>;tag=as574e27c1
    › To: <sip:098[redacted]7@185.[redacted].60:5060>;tag=sc2[redacted]-700c1701d39a822f
    › Contact: <sip:0041[redacted]2@185.[redacted].155:5060>
      Call-ID: 6268823e0242fe503fe46fd75c160b38@185.[redacted].155:5060
      [Generated Call-ID: 6268823e0242fe503fe46fd75c160b38@185.[redacted].155:5060]
    › CSeq: 104 INVITE
      User-Agent: dispatcher-sbc
      Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, SUBSCRIBE, NOTIFY, INFO, PUBLISH, MESSAGE
      Supported: replaces, timer
      Content-Type: application/sdp
      Content-Length: 242
  › Message Body
```

SIP-HEADER – UNDERSTANDING "DIVERSION"

- When a call is forwarded by a SIP proxy then the identity which forwarded the call is noted in the SIP-header "Diversion":
  - "From":
    Contains the number who started the call
  - "To":
    Contains the number of the new destination
  - "Diversion":
    Contains the number of the entity that forwarded the call



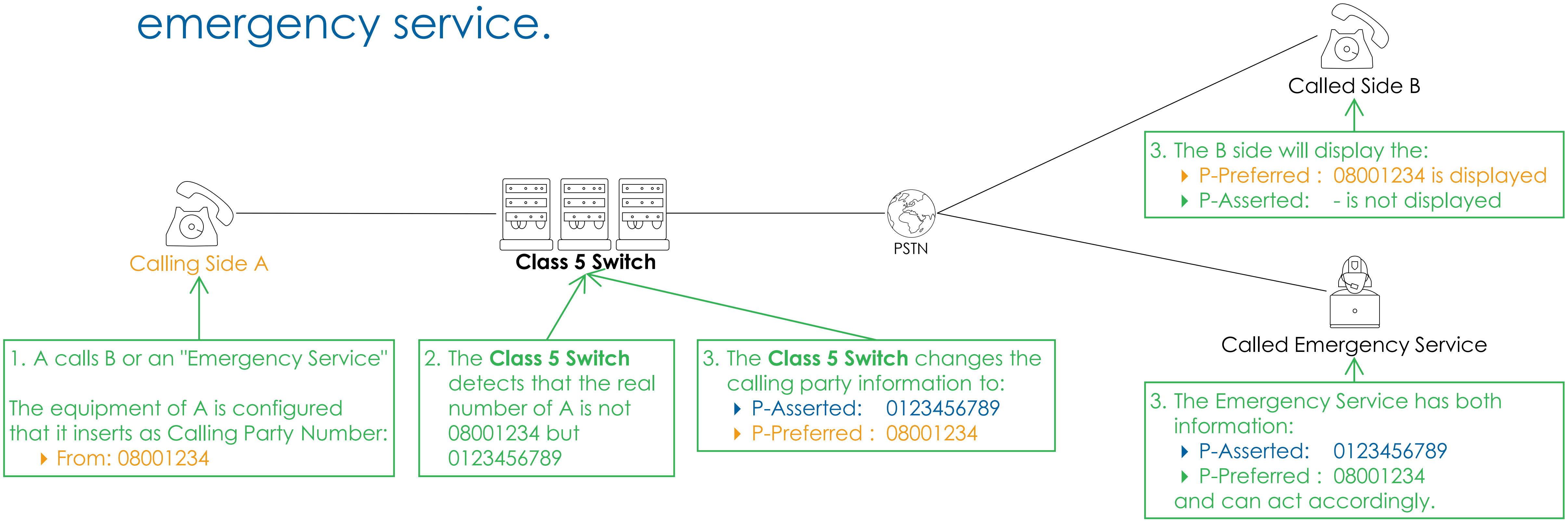


```
 4 11:01:28.302000 185.150.4.10    213.173.185.50  SIP/S…  1196 Request: INVITE sip:0449980150@213.173.185.50:5060 |
 5 11:01:28.306000 213.173.185.50  185.150.4.10    SIP      412 Status: 100 Trying |
 6 11:01:28.328000 213.173.185.50  185.150.4.10    SIP/S…   973 Request: INVITE sip:regdef@185.150.4.10:60169 |
 7 11:01:28.495000 185.150.4.10    213.173.185.50  SIP      519 Status: 180 Ringing |
 8 11:01:28.496000 213.173.185.50  185.150.4.10    SIP      431 Status: 180 Ringing |
 9 11:01:30.170000 185.150.4.10    213.173.185.50  SIP/S…   900 Status: 200 OK |
10 11:01:30.193000 213.173.185.50  185.150.4.10    SIP/S…   745 Status: 200 OK |

> Frame 6: 973 bytes on wire (7784 bits), 973 bytes captured (7784 bits)
> Ethernet II, Src: 00:00:00_00:5c:02 (00:00:00:00:5c:02), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
> Internet Protocol Version 4, Src: 213.[redacted], Dst: 185.[redacted]
> User Datagram Protocol, Src Port: 5060, Dst Port: 60169
v Session Initiation Protocol (INVITE)
  > Request-Line: INVITE sip:regdef@185.[redacted]:60169 SIP/2.0
  v Message Header
    > From: "Yealink WS Trainer"<sip:[redacted]100@213.[redacted]>;tag=sc2clt4-2b4455b717f59aa0
    > To: <sip:[redacted]180@213.[redacted]>
      Call-ID: 0_2208896448@172.30.168.100[1]
      [Generated Call-ID: 0_2208896448@172.30.168.100[1]]
    > CSeq: 1 INVITE
      Allow: INVITE, ACK, CANCEL, BYE, OPTIONS
      Max-Forwards: 30
      User-Agent: AareSwitch/6.12.12966
      Session-Expires: 1780;refresher=uas
      Min-SE: 90
      Supported: timer
    > Via: SIP/2.0/UDP 213.[redacted]:5060;nat;uac=sc2;branch=z9hG4bKsc2clt4-0fcca7f987593fe8
    > Contact: <sip:[redacted]100@213.[redacted]:5060>
      Diversion: <sip:[redacted]150@213.[redacted]>;privacy=off;counter=1
      Content-Type: application/sdp
      Content-Length: 318
  > Message Body
```

SIP-HEADER – UNDERSTANDING "P-ASSERTED" & "P-PREFERRED"

- In the public PSTN network the telephone provider with a Class 5 Switch must prove the Caller ID of the calling side.
- This so that in the event of an emergency call, a valid and the calling party correctly identifying number is sent to the emergency service.



1. A calls B or an "Emergency Service"

The equipment of A is configured that it inserts as Calling Party Number:
- From: 08001234

2. The **Class 5 Switch** detects that the real number of A is not 08001234 but 0123456789

3. The **Class 5 Switch** changes the calling party information to:
- P-Asserted: 0123456789
- P-Preferred : 08001234

3. The B side will display the:
- P-Preferred : 08001234 is displayed
- P-Asserted: - is not displayed

3. The Emergency Service has both information:
- P-Asserted: 0123456789
- P-Preferred : 08001234

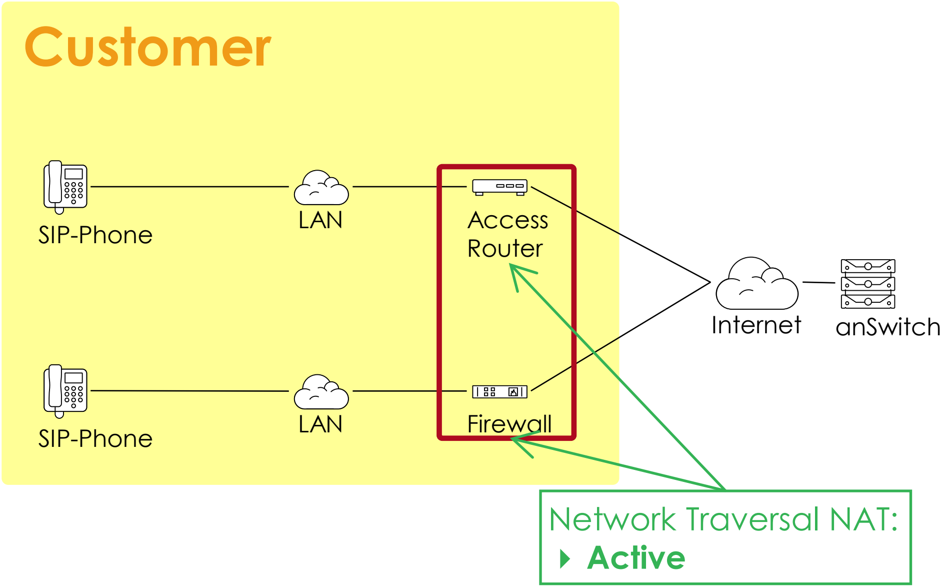
and can act accordingly.

4 IT "NAT TRAVERSAL"

OVERVIEW "NAT TRAVERSAL"

- The user's IP equipment can produce specific SIP problems we name:
  - "NAT Traversal Problems"

- Knowing about the underlying IP routing basics can prevent annoying support cases.

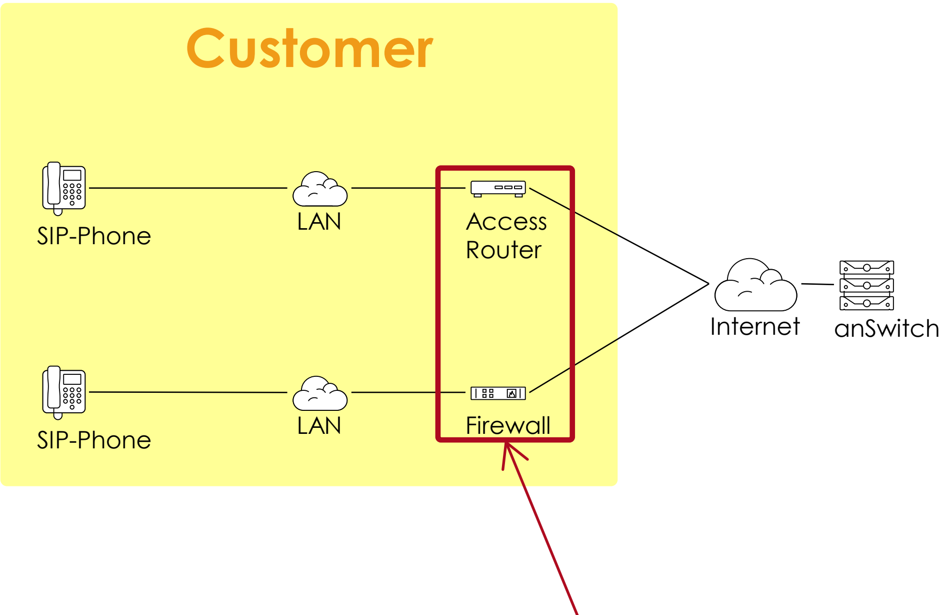


"NAT TRAVERSAL PROBLEMS" – SIP-ALG, SIP-HELPER

- Firewalls, router etc. often "support" SIP with SIP-ALG's and SIP-Helpers and so on.
- Often it is not clear what these "helpers" do and when they do something.
- They may manipulate the SIP messages in an unpredictable way and causes interoperation problems.

- Prevent these problems by switching off permanently:
  - SIP-ALG
  - SIP-Helper

- Make sure that on the FW:
  - An own UDP port type is defined for the SIP protocol (default UDP 5060) and assign it to the policy that handles the incoming and outgoing IP traffic to the anSwitch.
  - The RTP port are defined correctly in the policies.



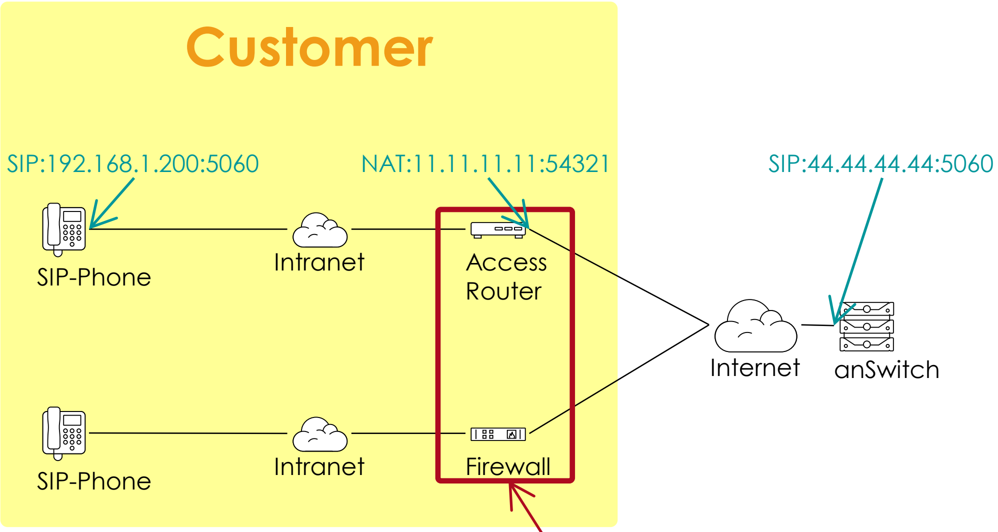
Switch off:
- SIP-ALG's
- SIP-Helpers
- etc.

Note

Often FWs predefine a SIP protocol type for using them in its policies. The experience shows that using predefined protocol types trigger the FW's usage of its built-in "helpers".

"NAT TRAVERSAL PROBLEMS" – NAT TRAVERSAL TIMEOUT

- During the registration process of a SIP device the anSwitch learns if a NAT is involved in the IP packet flow.
- The first REGISTRATION message of the SIP device opens a NAT port on the IP device e.g., access router:
  → Through this open NAT port all following SIP messages must pass
- In order that the NAT port isn't closed unexpectedly by the IP device e.g., NAT timeout, the anSwitch LoadBalancer sends every few seconds (default: 9sec, Data: 0d0a) a SIP OPTION message toward the NAT port.
- To prevent problems by closing NAT ports prematurely:
  - Check that the NAT'ing device keeps open the NAT port as long the SIP device is registered
    → Configure a long during connection > 30min



Make sure that the NAT is kept open as long the SIP device is registered.

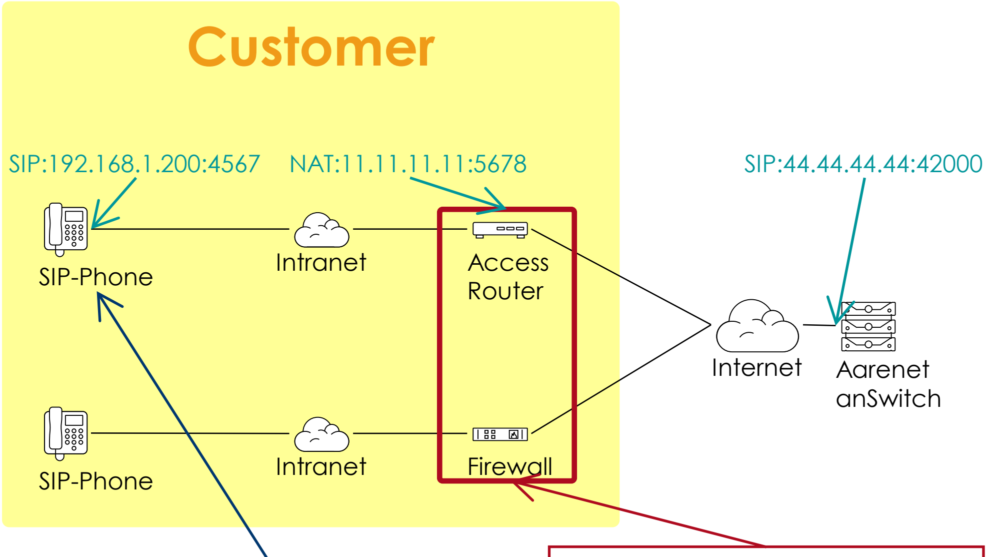
"NAT TRAVERSAL PROBLEMS" – RTP MEDIA STREAM

- The first RTP packet message of the SIP device opens a NAT port on the IP device e.g., access router, firewall.
  → Through this open NAT port all following RTP packets must pass.

- By receiving the first RTP packet the Aarenet anSwitch learns to which NAT port it must send its RTP packets.
- The NAT port is kept open as long RTP packets are exchanged.

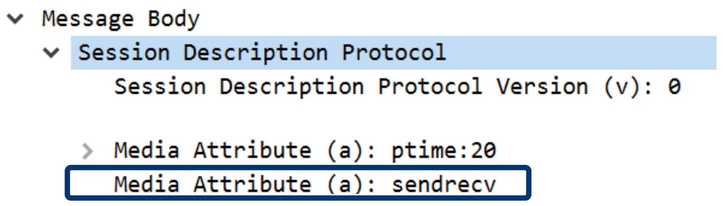
- To prevent problems by closing NAT ports prematurely:
  - Check that the NAT'ing device keeps open the NAT port as long the SIP device is registered
    → Configure a long during connection > 30min

Note It is paramount that the SIP device starts sending RTP packages immediately and keeps on sending them until the connection is closed.



Make sure that the NAT :
- Keep open then NAT port as long the SIP device is registered

When there are troubles check that the SIP phone sends in its SDP part the media attribute:
- sendrecv



```
∨ Message Body
  ∨ Session Description Protocol
      Session Description Protocol Version (v): 0

    > Media Attribute (a): ptime:20
      Media Attribute (a): sendrecv
```

5 AUDIO & MEDIA TRANSMISSION

OVERVIEW AUDIO & MEDIA TRANSMISSION

- Audio & media transmission and coding technology handles:
  - Audio
  - Fax
  - DTMF

- The SDP protocol is assigned for the negotiation of the transmission between the SIP peers
  (for details see the section "SIP & SDP: Protocol Basics").

- Planning the media transmission policies in a VoIP system is as important as the call routing planning.

  ➔ Do not underestimate this topic as it may produce up to 50% of your daily support problems!

THE "CODEC CONCEPT"

- Do a "Codec Concept" :
  - Which is the standard audio codec in the VoIP system
  - Which other audio codec are supported
  - What is the standard Fax transmission technology in the VoIP system
  - Which DTMF transmission is supported
- Do interop testing with the VoIP system equipment and recommended customer SIP devices.
- Communicate the "Codec Concept" to your customers.

THE AUDIO TRANSMISSION

- A wide range of audio codecs exists that can be used for voice transmission, e.g.:
  - PCM based codec : G.711 → provide good voice quality
  - Different narrow-band codec : G.722, G.722, G.726 → Voice "ok"
  - Modern VoIP Codec : IBLC, Opus → provide good voice quality

- Opus the current state of the art audio codec:
  - Compressing codec
  - Adaptive codec, adapts to changing bandwidth availability
  - HD codec
  - Unaffected by packet-loss

- HD codec are only HD within the pure VoIP world.
  → The transition to the PSTN is usually associated with transcoding to a PCM or narrow-band codec.

THE FAX TRANSMISSION

- For the Fax transmission 2 options are available:
  - G.711: for in-band Fax transmission
    - Use G.711 in an IT environment of good quality with:
      - Low jitter
      - No packet loss
  - T.38: for packetized out-band Fax transmission
    - Use T.38 in a less reliable IT environment
    - There are different T.38 flavors that may cause interop problems between the peers
- Check which Fax transmission concept the PSTN carrier supports. Following its concept may reduce transcoding hassles.

Note

- Every transcoding hampers the Fax transmission!
- The experience shows that more than two transcoding points enhances the probability of erroneous Fax transmissions.

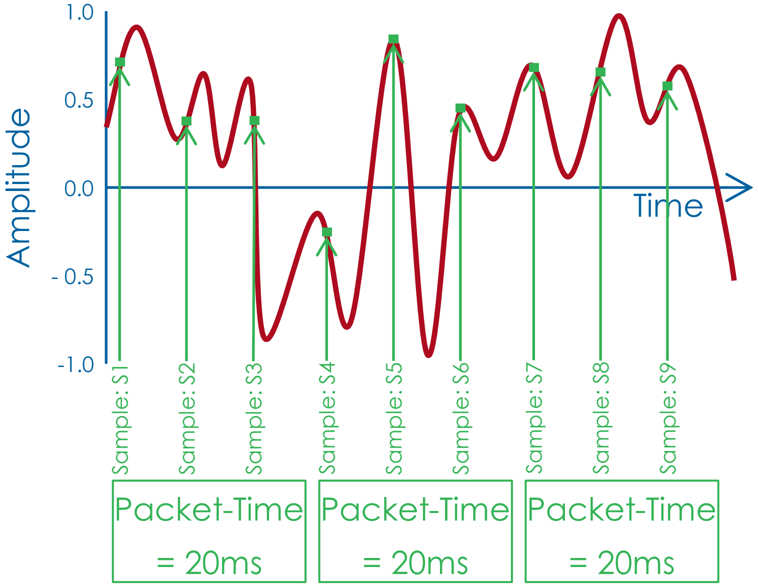
THE DTMF TRANSMISSION

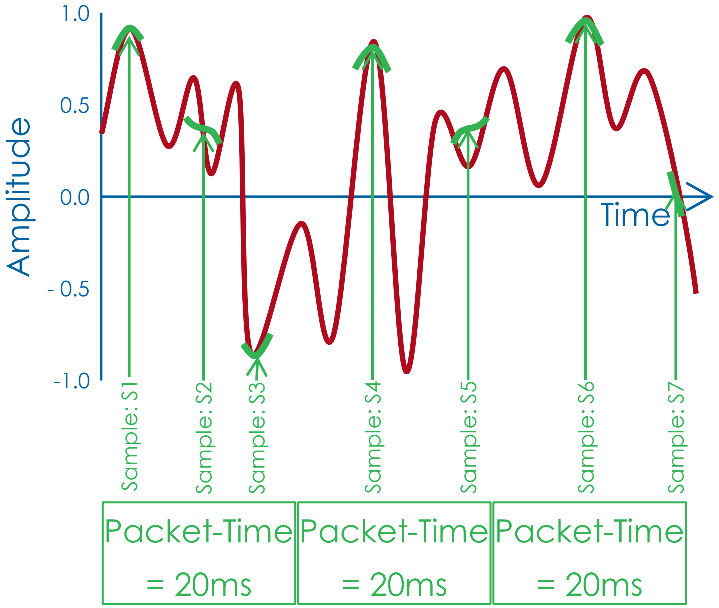
- There are different ways for transmitting DTMF:
  - In Band: Any audio codec can transmit DTMF
    - + Nothing special must be done
    - - If an audio transcoding occurs the DTMF may be unrecognizable at the destination peer
  - RFC2833: The DTMF digit is transmitted in an own RTP packet
    - + Works fine
    - - RFC2833 may be not supported by all SIP peers
  - SIP Info: The DTMF digit is transmitted in a SIP Info message
    - + DTMF transmission can be checked in the SIP message flow
    - - Audio duration is fix and not original

→ We experience good results by implementing RFC2833

THE AUDIO SAMPLING & PACKET-TIME

- There are different types of codecs available. They differentiate by:
  - Coding technology
  - Sampling rate
- For a successful audio transmission both peers have to use:
  - The same codec
  - The same Packet-Time





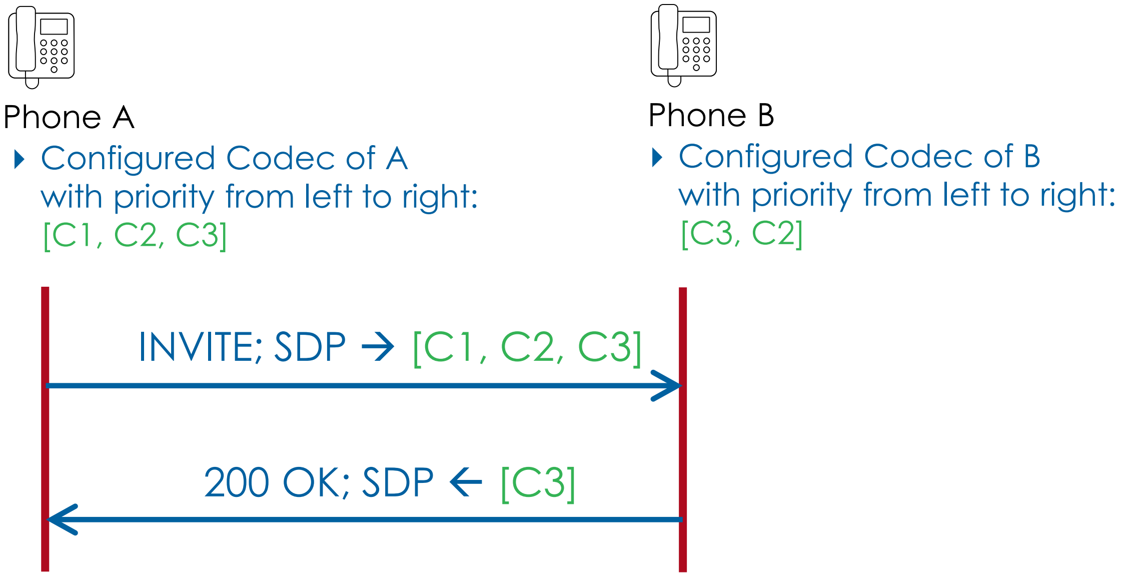
OVERVIEW OF SUPPORTED AUDIO CODECS

Codec	Media Attributes		Remark
G.711µlaw, PCMU	0 PCMU/8000		Very good quality VoIP codec
GSM	3 GSM/8000		Standard mobile codec
G.723-53 / G.723-63	4 G723/8000	Packet size = 30ms	Quality VoIP codec
G.711alaw, PCMA	8 PCMA/8000		Very good quality VoIP codec, ISDN
G.722	9 G722/8000		Quality VoIP codec
G.729	18 G729/8000	fmtp: 18 annexb=no	Low quality VoIP codec
G.726-16 / aa12-g726-16	97 G726-16/8000		Good quality VoIP codec
G.726-24 / aa12-g726-24	96 G726-24/8000		Good quality VoIP codec
G.726-32 / aa12-g726-32	99 G726-32/8000		Good quality VoIP codec
G.726-40 / aa12-g726-40	98 G726-40/8000		Good quality VoIP codec
101	101 telephone-event/8000	fmtp: 101 0-16	DTMF, RFC 2833 0-15 : DTMF character 0-9, *,#, A,B,C,D 0-16 : DTMF character 0-9, *,#, A,B,C,D, Flash
IBLC	102 ILBC/8000		Modern VoIP codec
Opus	103 OPUS/48000/2		Modern adaptive VoIP codec
Speex	110 SPEEX/8000		Encryption codec
125	125 X-CLEAR-CHANNEL/8000		Data service: Echo canceling will be switched off and the data bit by bit transferred.

- More details see: https://en.wikipedia.org/wiki/RTP_audio_video_profile

THE AUDIO CODEC NEGOTIATION

- Standard codec negotiation:
  - A offers a list of supported codecs
  - B selects the best matching codec
- If no codec matches, then the call will be rejected.



Phone A

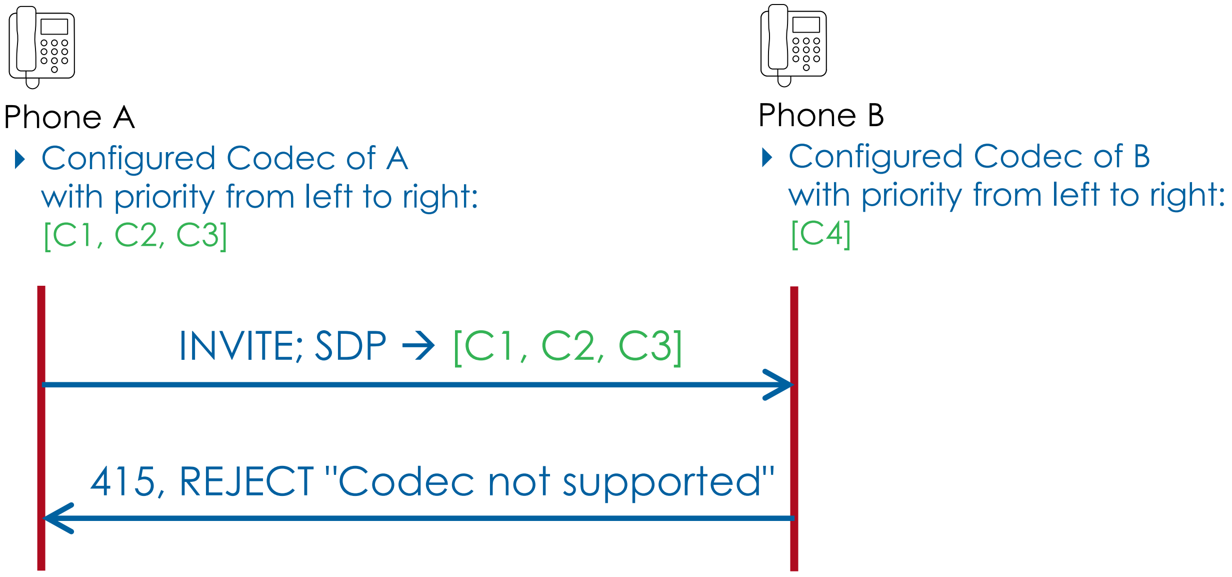
- Configured Codec of A with priority from left to right: [C1, C2, C3]

Phone B

- Configured Codec of B with priority from left to right: [C3, C2]

INVITE; SDP → [C1, C2, C3]

200 OK; SDP ← [C3]



Phone A

- Configured Codec of A with priority from left to right: [C1, C2, C3]

Phone B

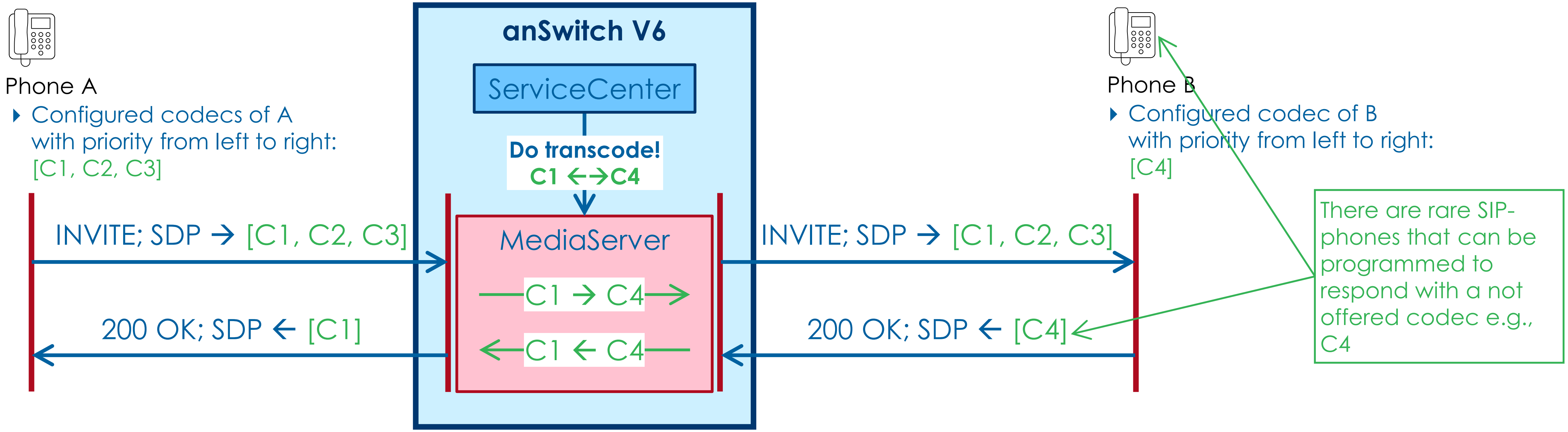
- Configured Codec of B with priority from left to right: [C4]

INVITE; SDP → [C1, C2, C3]

415, REJECT "Codec not supported"

ANSWITCH V6: AUDIO NEGOTIATION & TRANSCODING

- If the called peer selects a not offered codec e.g., C4, then the anSwitch jumps in and does automatically transcode the audio stream.



Phone A

- Configured codecs of A with priority from left to right: [C1, C2, C3]

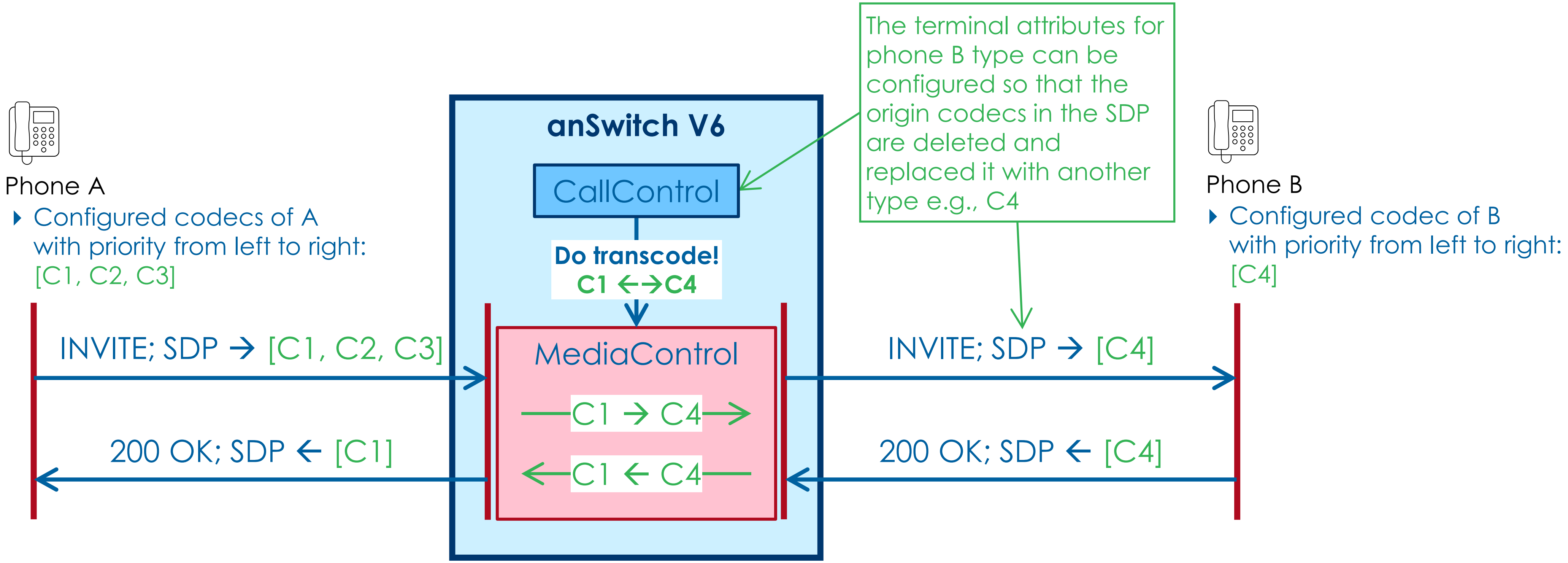
Phone B

- Configured codec of B with priority from left to right: [C4]

There are rare SIP-phones that can be programmed to respond with a not offered codec e.g., C4

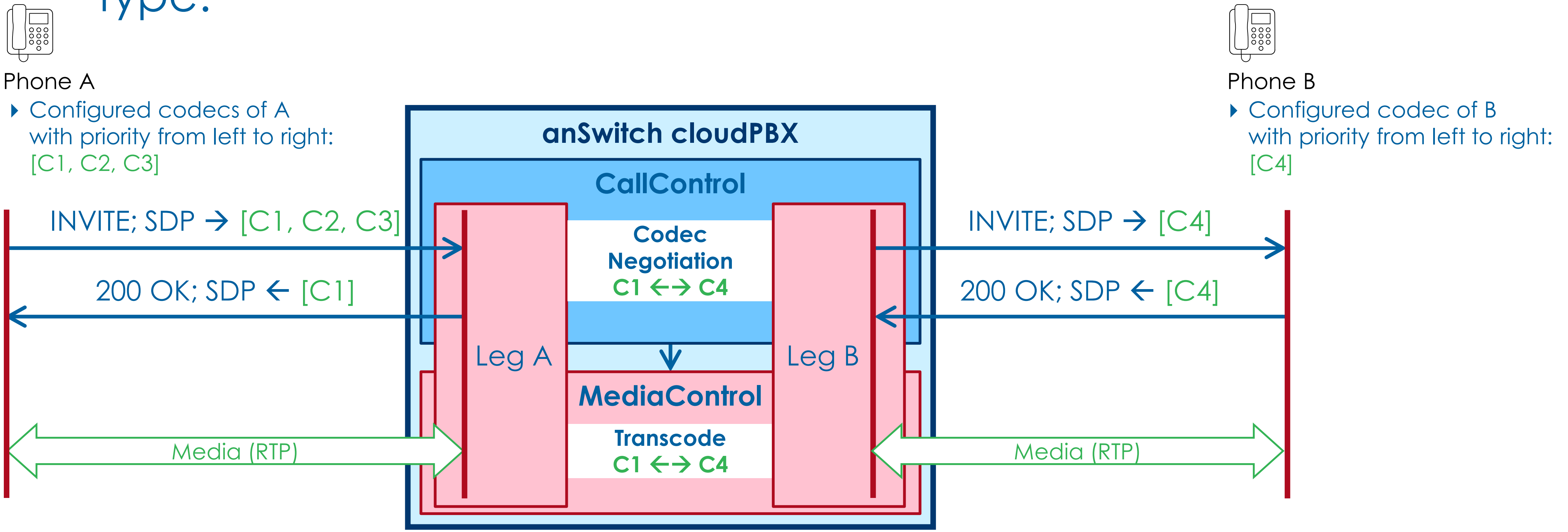
ANSWITCH V6: AUDIO CODEC FORCING TRANSCODING

- The anSwitch V6 can force the audio transcoding by sending the called side a preferred codec.
- The SDP protocol will be manipulated according a terminal attribute that is defined for the phone B type.


Phone A
Configured codecs of A with priority from left to right: [C1, C2, C3]
anSwitch V6
CallControl
Do transcode! C1 ←→C4
MediaControl
C1 → C4
C1 ← C4
The terminal attributes for phone B type can be configured so that the origin codecs in the SDP are deleted and replaced it with another type e.g., C4
Phone B
Configured codec of B with priority from left to right: [C4]
INVITE; SDP → [C1, C2, C3]
200 OK; SDP ← [C1]
INVITE; SDP → [C4]
200 OK; SDP ← [C4]

ANSWITCH V7: TRANSCODING IS STANDARD

- The anSwitch V7 terminates the media streams of each call leg and transcodes it if needed.
- The SDP protocol will be manipulated by the CallControl according the terminal attribute that is defined for the phone B type.



Phone A

- Configured codecs of A with priority from left to right: [C1, C2, C3]

Phone B

- Configured codec of B with priority from left to right: [C4]

LAST PAGE

Empty Page