EINTE Lab 4 - VoIP
Author: Sławomir Batruch

Setup

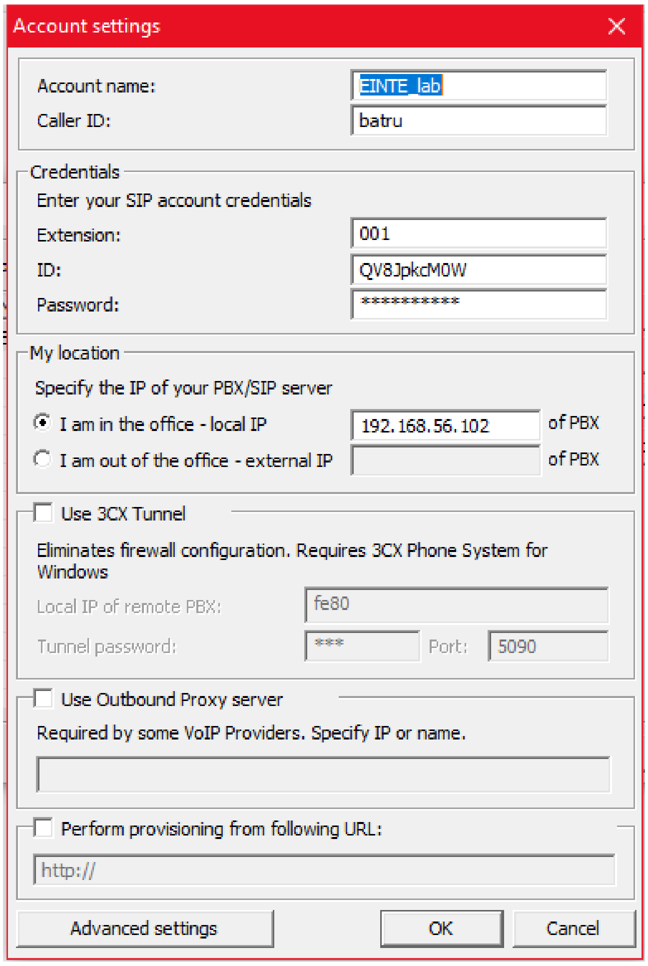
To perform the lab task, some setting up needed to be done.

After downloading all the necessary files, first we launched the debian VM to obtain the IP to connect to the VoIP server:

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1
000
    link/ether 08:00:27:35:34:6d brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.102/24 brd 192.168.56.255 scope global enp0s3
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe35:346d/64 scope link
       valid_lft forever preferred_lft forever
```

We see the address IPv4 enp0s3 being 192.168.56.102

Then, knowing the IP address and other information, we set up 3cx profile:



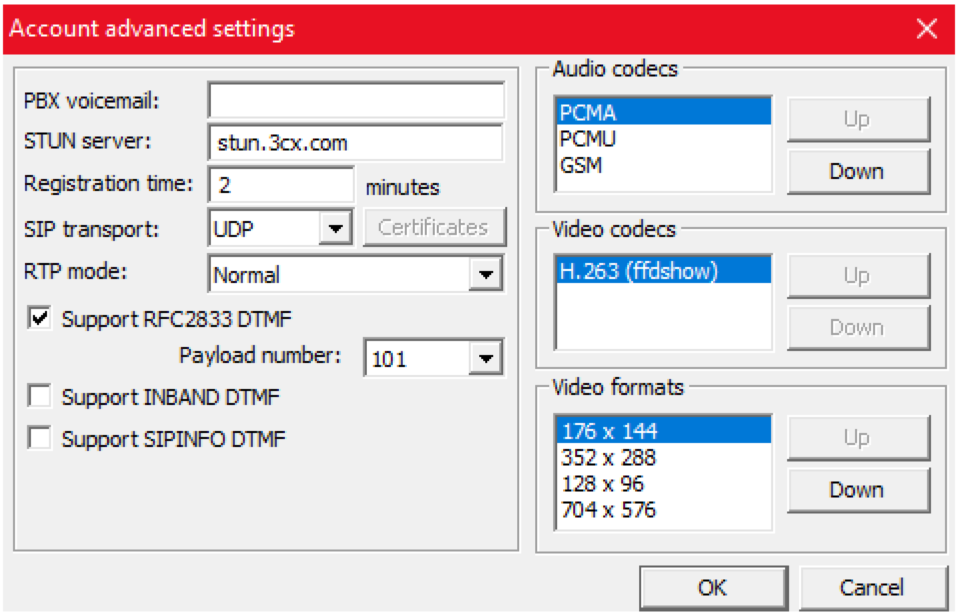
ID and pass was given in the instruction .pdf

Wireshark setup

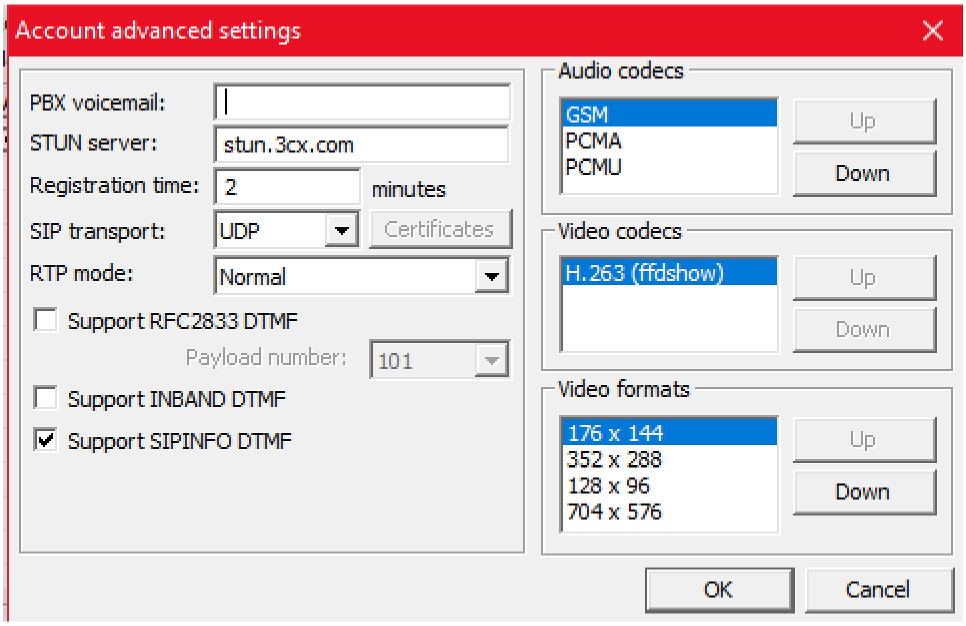
Wireshark was set to sniff traffic on the VirtualBox Host-Only Network. No capture filter was used, but later on a display filter was used to only display SIP and RTP traffic. Some other traffic, unrelated to the lab, was captured.
The obtained .pcapng was saved and analyzed after two calls were made.

3cx calls setup

Going into advanced settings, we can see more detailed call settings. Since the task required to make two calls with different settings, we show the used settings. 1st call:

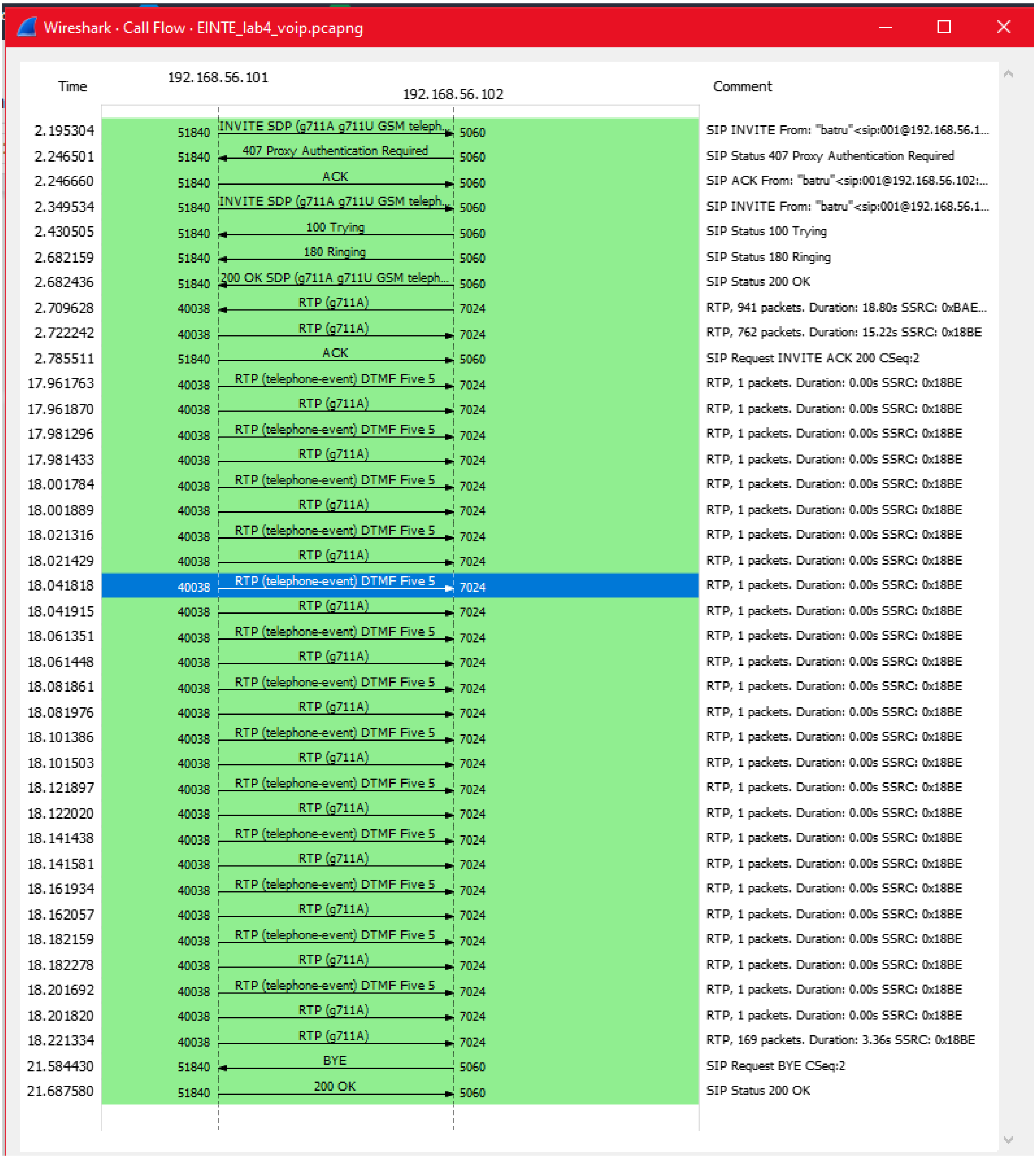


And for the 2nd call, GSM was the primary codec. Furthermore, RFC2833 DTMF support was turned off. Instead - SIPINFO DTMF support was turned on:

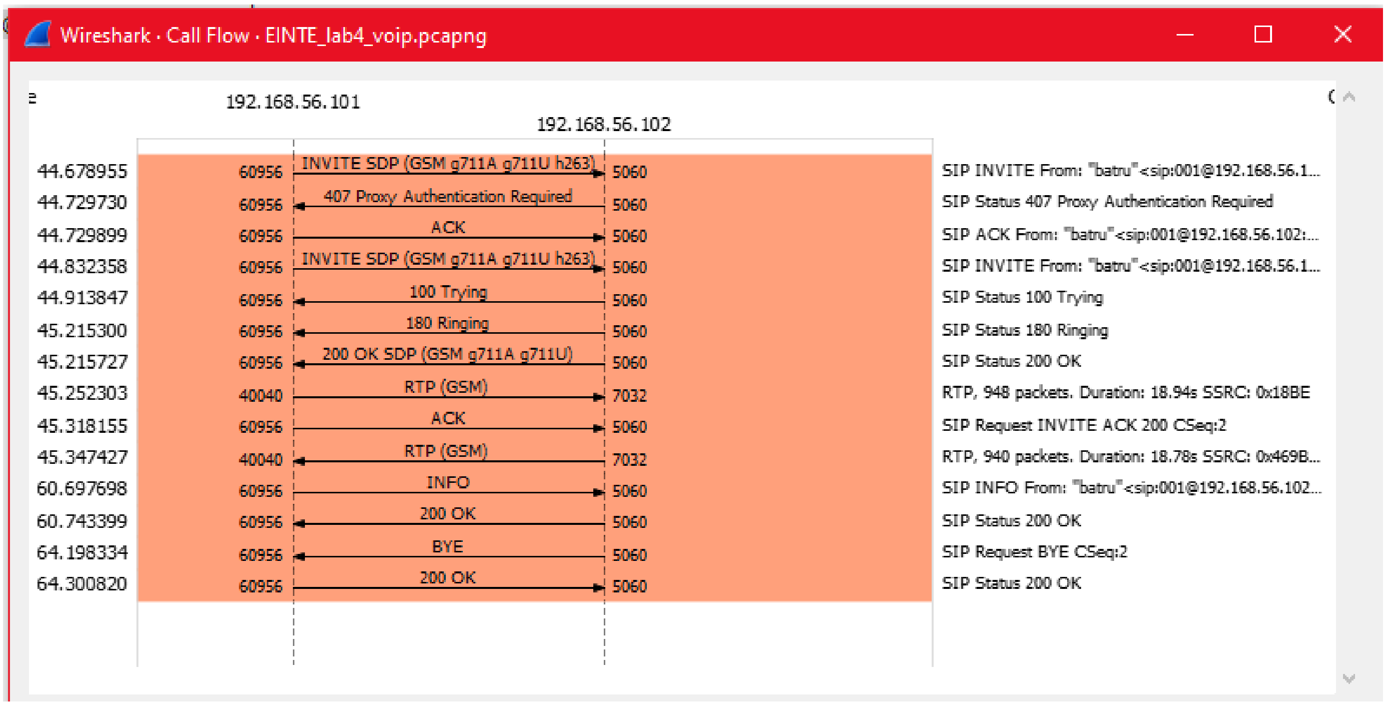


VoIP session analysis - Call Flow

Below we present the sequence of messages regarding the VoIP calls, as seen in the wireshark's telephony tab, flow sequence. For the first call (PCMA):



And below - 2nd call (GSM):



VoIP questions - first session (PCMA)

What is the meaning of 407 Proxy Authentication Required response received after the first INVITE?

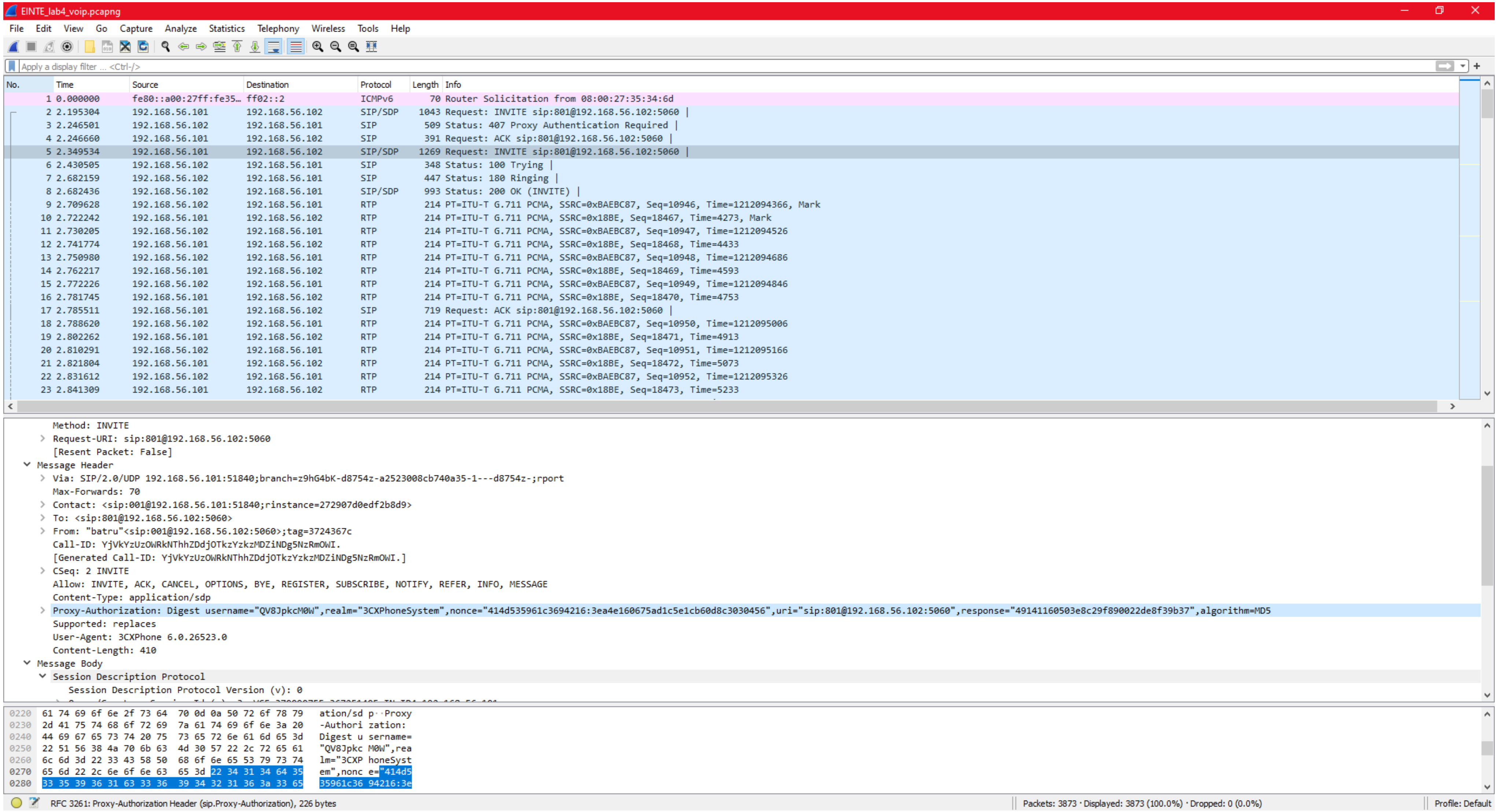
The first INVITE packet does not contain any authentication information.
After the 3cx client acknowledges that, it sends an INVITE message, but this time with credentials that are used for authentication.

That is why after the 2nd INVITE message, the SIP is ‘trying’ and then ‘ringing’. In general - after the 2nd INVITE message, the session is properly established.

To see this actually being the case, we look into the following packet info:

```
∨ Message Header
  > Via: SIP/2.0/UDP 192.168.56.101:51840;branch=z9hG4bK-d8754z-b20d8068893a9c77-1---d8754z-;rport
    Max-Forwards: 70
  > Contact: <sip:001@192.168.56.101:51840;rinstance=272907d0edf2b8d9>
  > To: <sip:801@192.168.56.102:5060>
  > From: "batru"<sip:001@192.168.56.102:5060>;tag=3724367c
    Call-ID: YjVkYzUzOWRkNThhZDdjOTkzYzkzMDZiNDg5NzRmOWI.
    [Generated Call-ID: YjVkYzUzOWRkNThhZDdjOTkzYzkzMDZiNDg5NzRmOWI.]
  > CSeq: 1 INVITE
    Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REGISTER, SUBSCRIBE, NOTIFY, REFER, INFO, MESSAGE
    Content-Type: application/sdp
    Supported: replaces
    User-Agent: 3CXPhone 6.0.26523.0
    Content-Length: 410
```

Is the header of the first INVITE.



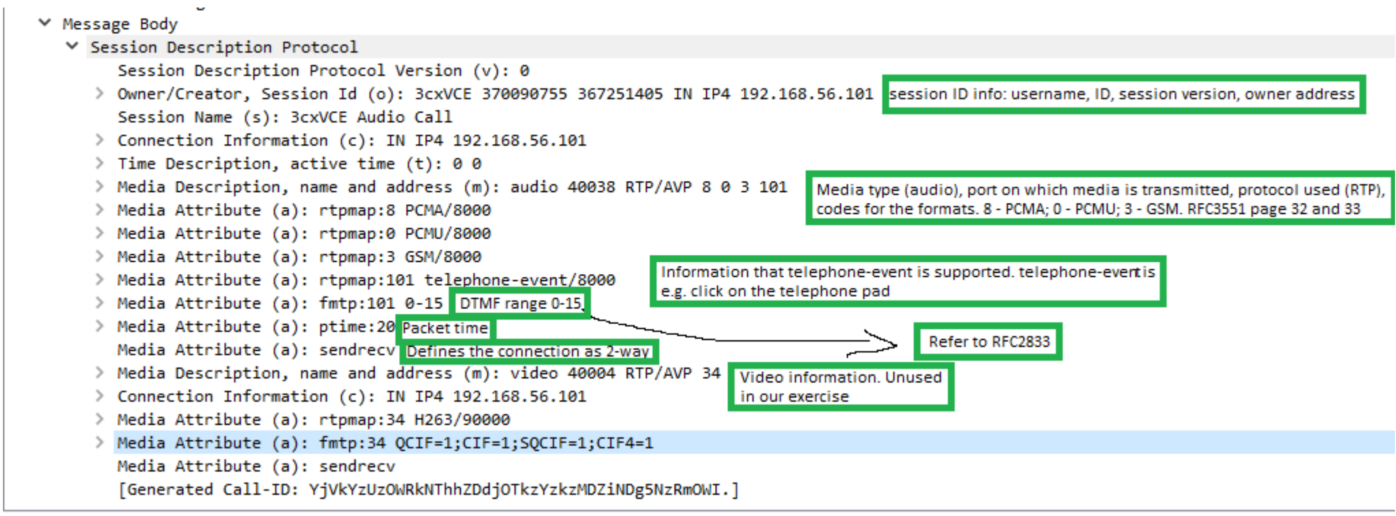
Is the header of the 2nd invite message. We see the proxy-authorization message.
We can't see the exact password and username due to md5 encryption.

What is the content of the Media Description line in SDP descriptions sent within the SIP INVITE and 200 OK messages? Attach the relevant screenshots and explain the meaning of the most important fields

The following sources were used to understand the body of SDP:
Page 32 and 33 from: https://datatracker.ietf.org/doc/html/rfc3551
Codes for the formats

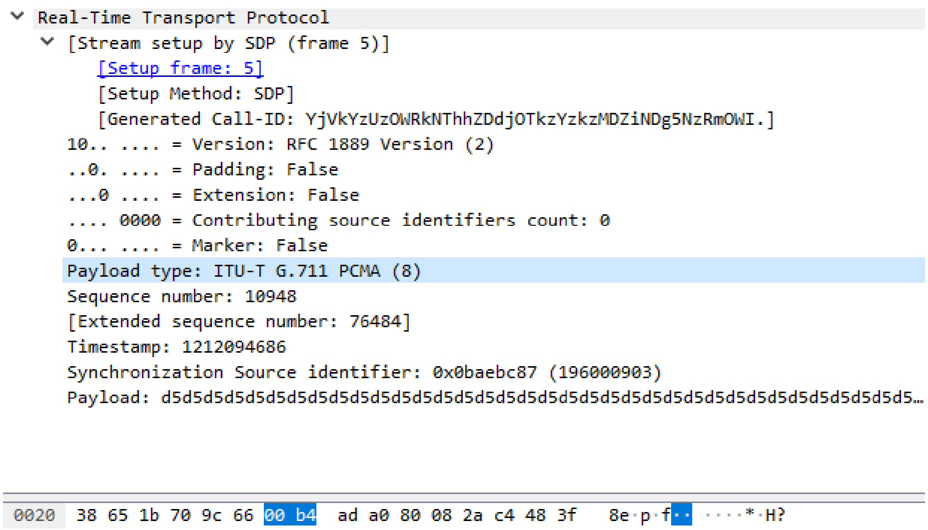
Table 1, page 10 from: https://datatracker.ietf.org/doc/html/rfc2833
DTMF (Dual-tone multi-frequency signaling) range explained

The explanations are on the screenshot. Text in green borders are our comments on the information to the left:



What is the voice codec used in the session?

Looking at arbitrary payload that was sent during the first call:



We see the payload type being ITU-T G.711 PCMA (G.711 with A-law algorithm)

In some RTP packets the Mark field in the packet header has the value of 1 – why?

From RFC3550, https://datatracker.ietf.org/doc/html/rfc3550 page 13:
marker (M): 1 bit

> The interpretation of the marker is defined by a profile. It is intended to allow significant events such as frame boundaries to be marked in the packet stream.

So the point of the marker bit is to separate bitstreams, frame boundaries.
For the first call, there are two marker bits set to 1 at the beginning.

How the information about the pressed key is sent by the SIP client to the VoIP server?

With the use of RTP EVENT packet type. Example snippet showing these packets:

1569	17.961763	192.168.56.101	192.168.56.102	RTP EVENT	58	Payload type=RTP Event, DTMF Five 5
1570	17.961870	192.168.56.101	192.168.56.102	RTP	214	PT=ITU-T G.711 PCMA, SSRC=0x18BE, Seq=19230, Time=126193
1571	17.970735	192.168.56.102	192.168.56.101	RTP	214	PT=ITU-T G.711 PCMA, SSRC=0xBAEBC87, Seq=11709, Time=1212216446
1572	17.981296	192.168.56.101	192.168.56.102	RTP EVENT	58	Payload type=RTP Event, DTMF Five 5
1573	17.981433	192.168.56.101	192.168.56.102	RTP	214	PT=ITU-T G.711 PCMA, SSRC=0x18BE, Seq=19232, Time=126353
1574	17.992355	192.168.56.102	192.168.56.101	RTP	214	PT=ITU-T G.711 PCMA, SSRC=0xBAEBC87, Seq=11710, Time=1212216606
1575	18.001784	192.168.56.101	192.168.56.102	RTP EVENT	58	Payload type=RTP Event, DTMF Five 5
1576	18.001889	192.168.56.101	192.168.56.102	RTP	214	PT=ITU-T G.711 PCMA, SSRC=0x18BE, Seq=19234, Time=126513
1577	18.012905	192.168.56.102	192.168.56.101	RTP	214	PT=ITU-T G.711 PCMA, SSRC=0xBAEBC87, Seq=11711, Time=1212216766
1578	18.021316	192.168.56.101	192.168.56.102	RTP EVENT	58	Payload type=RTP Event, DTMF Five 5
1579	18.021429	192.168.56.101	192.168.56.102	RTP	214	PT=ITU-T G.711 PCMA, SSRC=0x18BE, Seq=19236, Time=126673
1580	18.028733	192.168.56.102	192.168.56.101	RTP	214	PT=ITU-T G.711 PCMA, SSRC=0xBAEBC87, Seq=11712, Time=1212216926
1581	18.041818	192.168.56.101	192.168.56.102	RTP EVENT	58	Payload type=RTP Event, DTMF Five 5
1582	18.041915	192.168.56.101	192.168.56.102	RTP	214	PT=ITU-T G.711 PCMA, SSRC=0x18BE, Seq=19238, Time=126833
1583	18.049204	192.168.56.102	192.168.56.101	RTP	214	PT=ITU-T G.711 PCMA, SSRC=0xBAEBC87, Seq=11713, Time=1212217086
1584	18.061351	192.168.56.101	192.168.56.102	RTP EVENT	58	Payload type=RTP Event, DTMF Five 5

We see that the event is a ‘DTMF Five 5’

```
Real-Time Transport Protocol
  [Stream setup by SDP (frame 5)]
      [Setup frame: 5]
      [Setup Method: SDP]
      [Generated Call-ID: YjVkYzUzOWRkNThhZDdjOTkzYzkzMDZiNDg5NzRmOWI.]
  10.. .... = Version: RFC 1889 Version (2)
  ..0. .... = Padding: False
  ...0 .... = Extension: False
  .... 0000 = Contributing source identifiers count: 0
  1... .... = Marker: True
  Payload type: telephone-event (101)
  Sequence number: 19229
  [Extended sequence number: 84765]
  Timestamp: 126193
  Synchronization Source identifier: 0x000018be (6334)
RFC 2833 RTP Event
  Event ID: DTMF Five 5 (5)
  0... .... = End of Event: False
  .0.. .... = Reserved: False
  ..00 1010 = Volume: 10
  Event Duration: 160
```

This is a telephone-event that was mentioned before.

Then at the end of the button press, RTP EVENTS showing the press has ended are sent. Note the ‘end of event: true’:

```
1608 18.221217   192.168.56.101   192.168.56.102   RTP EVENT    58 Payload type=RTP Event, DTMF Five 5 (end)
1609 18.221231   192.168.56.101   192.168.56.102   RTP EVENT    58 Payload type=RTP Event, DTMF Five 5 (end)
1610 18.221237   192.168.56.101   192.168.56.102   RTP EVENT    58 Payload type=RTP Event, DTMF Five 5 (end)
1611 18.221334   192.168.56.101   192.168.56.102   RTP         214 PT=ITU-T G.711 PCMA, SSRC=0x18BE, Seq=192
1612 18.231136   192.168.56.102   192.168.56.101   RTP         214 PT=ITU-T G.711 PCMA, SSRC=0xBAEBC87, Seq=
1613 18.241820   192.168.56.101   192.168.56.102   RTP         214 PT=ITU-T G.711 PCMA, SSRC=0x18BE, Seq=192
1614 18.251843   192.168.56.102   192.168.56.101   RTP         214 PT=ITU-T G.711 PCMA, SSRC=0xBAEBC87, Seq=

> Internet Protocol Version 4, Src: 192.168.56.101, Dst: 192.168.56.102
> User Datagram Protocol, Src Port: 40038, Dst Port: 7024
v Real-Time Transport Protocol
  v [Stream setup by SDP (frame 5)]
      [Setup frame: 5]
      [Setup Method: SDP]
      [Generated Call-ID: YjVkYzUzOWRkNThhZDdjOTkzYzkzMDZiNDg5NzRmOWI.]
    10.. .... = Version: RFC 1889 Version (2)
    ..0. .... = Padding: False
    ...0 .... = Extension: False
    .... 0000 = Contributing source identifiers count: 0
    0... .... = Marker: False
    Payload type: telephone-event (101)
    Sequence number: 19255
    [Extended sequence number: 84791]
    Timestamp: 126193
    Synchronization Source identifier: 0x000018be (6334)
v RFC 2833 RTP Event
    Event ID: DTMF Five 5 (5)
    1... .... = End of Event: True
    .0.. .... = Reserved: False
    ..00 1010 = Volume: 10
    Event Duration: 2080

0000  08 00 27 35 34 6d 0a 00  27 00 00 05 08 00 45 00   ··'54m·· '·····E·
0010  00 2c b6 ae 00 00 80 11  91 f6 c0 a8 38 65 c0 a8   ·,······ ····8e··
0020  38 66 9c 66 1b 70 00 18  76 d3 80 65 4b 37 00 01   8f·f·p·· v··eK7··
0030  ec f1 00 00 18 be 05 8a  08 20                     ········ ·
```

What is the sequence number and timestamp of the first RTP packet sent by the server to the client, and why are they not 0 or 1?

If the sequence number would be 0 or 1 in every case, then simultaneous streams would interfere with each other. Or it wouldn’t even be possible to have many different streams at the same time.
Randomisation of that prevents the issue of collision with other streams.

For the first RTP packet received FROM SIP server in the 1st call:

- Sequence number is 10946
- Time is 1212094366

2nd RTP packet, which is 1st sent TO SIP server is:

- Sequence number 18467
- Time 4273

VoIP questions - second session (GSM)

Which voice codec was used [in the second session]?

Looking at an arbitrary RTP packet:

```
Real-Time Transport Protocol
  [Stream setup by SDP (frame 1975)]
  10.. .... = Version: RFC 1889 Version (2)
  ..0. .... = Padding: False
  ...0 .... = Extension: False
  .... 0000 = Contributing source identifiers count: 0
  0... .... = Marker: False
  Payload type: GSM 06.10 (3)
  Sequence number: 18472
  [Extended sequence number: 84008]
  Timestamp: 5801
  Synchronization Source identifier: 0x000018be (6334)
  Payload: d820a2e15a50004924924924500049249249245000492492492450004924924924
```

We see the payload type being GSM 06.10. This codec is also known as Full Rate. It is a rather old codec, with low bitrate for modern standards.
The difference in the sound quality was heard (comparing to 1st call)

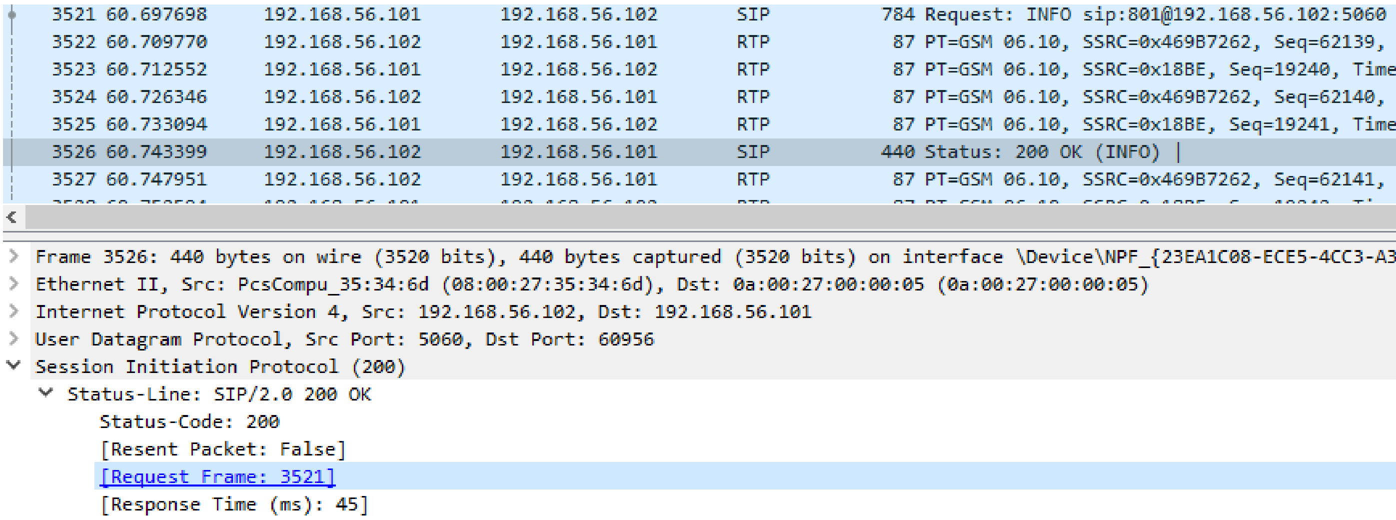
How is the information about the pressed key sent to the VoIP server?

With the use of SIP INFO:

```
3521 60.697698   192.168.56.101   192.168.56.102   SIP    784 Request: INFO sip:801@192.168.56.102:5060 |
3522 60.709770   192.168.56.102   192.168.56.101   RTP     87 PT=GSM 06.10, SSRC=0x469B7262, Seq=62139, T
3523 60.712552   192.168.56.101   192.168.56.102   RTP     87 PT=GSM 06.10, SSRC=0x18BE, Seq=19240, Time=
3524 60.726346   192.168.56.102   192.168.56.101   RTP     87 PT=GSM 06.10, SSRC=0x469B7262, Seq=62140, T
3525 60.733094   192.168.56.101   192.168.56.102   RTP     87 PT=GSM 06.10, SSRC=0x18BE, Seq=19241, Time=
3526 60.743399   192.168.56.102   192.168.56.101   SIP    440 Status: 200 OK (INFO) |
3527 60.747951   192.168.56.102   192.168.56.101   RTP     87 PT=GSM 06.10, SSRC=0x469B7262, Seq=62141, T

Frame 3521: 784 bytes on wire (6272 bits), 784 bytes captured (6272 bits) on interface \Device\NPF_{23EA1C08-ECE5-4CC3-A34
Ethernet II, Src: 0a:00:27:00:00:05 (0a:00:27:00:00:05), Dst: PcsCompu_35:34:6d (08:00:27:35:34:6d)
Internet Protocol Version 4, Src: 192.168.56.101, Dst: 192.168.56.102
User Datagram Protocol, Src Port: 60956, Dst Port: 5060
Session Initiation Protocol (INFO)
  Request-Line: INFO sip:801@192.168.56.102:5060 SIP/2.0
    Method: INFO
    Request-URI: sip:801@192.168.56.102:5060
    [Resent Packet: False]
  Message Header
  Message Body
    Signal=5\r\n
    Duration=250\r\n
```

Then an OK response is received, which means the signal '5' was sent properly.



Closing remarks

Below we provide a list of sources and what info was taken from them:

- https://datatracker.ietf.org/doc/html/rfc3551 - rtpmap format codes
- https://datatracker.ietf.org/doc/html/rfc2833 - DTMF code ranges
- https://datatracker.ietf.org/doc/html/rfc3550 - marker information

Naturally, the display filter in wireshark should be `sip || rtp`. That way, packets unrelated to the lab are not seen.

.pdf files of sequence flow are also attached, exported from Wireshark.

End