

Instituto Superior Técnico  
Mestrado Integrado em Engenharia Eletrotécnica e de  
Computadores

# Redes e Serviços Internet

Projeto Final

VoIP

Realizado por:

75573 – Bernardo Gomes

75876 – Tomás Costa

76090 – Diogo Ferreira

## Índice

Introdução .....	1
Arquitetura.....	1
Configuração do PBX.....	2
Protocolo SIP .....	4
Protocolo SDP .....	5
Protocolo RTP .....	6
Instruções de instalação do projeto.....	6
Ilustração das pilhas de protocolos utilizados .....	7
Observações no Wireshark .....	8
Diagramas temporais .....	12
Alterações no ficheiro sip.conf .....	17
Conclusões.....	19

# Introdução

Este projeto, tem como objetivo o desenvolvimento de um sistema de Interactive Video Voice Response (IVVR) baseado em voz sobre IP (VoIP), bem como a compreensão dos protocolos utilizados. Para a implementação do servidor de comunicações, baseou-se o sistema num PBX Asterisk e para a gestão das chamadas de voz, fez-se uso do protocolo SIP, enquanto que para a transmissão de som e vídeo recorreu-se a RTP/UDP.

## Arquitetura

A arquitetura do projeto realizado irá ser a seguinte:

- Plataforma PBX Asterisk, que irá correr numa máquina Linux acessível na rede, que irá permitir o registo de telefones SIP;
- Telefones SIP, que para os casos testados e que irão ser ilustrados correspondem a telefones X-Lite.

Para a inicialização do estudo efetuado, seguiram-se os seguintes passos:

1. Correu-se o comando *ifconfig* no servidor Linux, por forma a obter o IP ao qual os telefones se devem conectar (*192.168.56.101*);
2. Ligou-se o servidor através do comando *sudo asterisk -vvvv*;
3. Após a configuração dos telefones, estes conectam-se automaticamente ao servidor, sendo possível observar este facto através da linha de comandos por *sip show peers*, tal como pode ser verificado na Figura 1.



```
ubuntu-VirtualBox*CLI> sip show peers
Name/username      Host                               Dyn Forcerpor
t ACL Port        Status
3000                (Unspecified)                    D   N   6
    Unmonitored
3001/3001           192.168.56.1                     D   N   6
5500                (Unspecified)                    D   N   6
3002                (Unspecified)                    D   N   6
    Unmonitored
4000/4000           192.168.56.1                     D   N   6
0298                (Unspecified)                    D   N   6
4001                (Unspecified)                    D   N   6
    Unmonitored
5000                (Unspecified)                    D   N   6
    Unmonitored
5001                (Unspecified)                    D   N   6
    Unmonitored
7 sip peers [Monitored: 0 online, 0 offline Unmonitored: 2 online, 5 offline]
```

Figura 1-Telefones SIP ligados ao servidor

A partir destes telefones, irá assim ser possível realizar e aceitar chamadas telefónicas de outros telefones conectados à mesma rede, bem como realizar outras funções, a saber:

- Jogar ao jogo da tabuada;
- Saber as estatísticas do jogo;
- Aceder ao serviço de voicemail;
- Ligar ao operador;
- Desligar.

## Configuração do PBX

Para a implementação de um serviço como o pedido para o projeto, a parte essencial prende-se com a configuração do Private Branch Exchange (PBX). Este sistema terá como principal função reencaminhar as chamadas entre os utilizadores registados na rede, permitindo assim implementar a funcionalidade de ligar ao operador pedida no enunciado.

Sendo a plataforma Asterisk bastante personalizável, é possível editar os *dialplans*, por forma a implementar as funcionalidades pedidas. Estas configurações, são realizadas no ficheiro *extensions.conf*, responsável pela definição de contextos de números de extensões e as respetivas associações a telefones da rede. As alterações realizadas neste ficheiro foram as seguintes:

- Adição da aplicação IVVR ao PBX na extensão correspondente ao número de aluno algebricamente menor de entre os números dos elementos do grupo:

```
exten => _300X,1,Dial(sip/${EXTEN},,r)
exten => _400X,1,Dial(sip/${EXTEN},,r)
exten => _500X,1,Dial(sip/${EXTEN},,r)
exten => 9999,1,Goto(teste_IVVR,s,1)
;Alunos: acrescentar aqui
exten => 75573,1,Goto(proj,s,1)
```

Figura 2-Definição da extensão para a aplicação IVVR

- Criação das opções 9 e 0 do menu, correspondentes à chamada para o operador e a desligar a chamada respetivamente. Note-se que o operador, tal como definido no relatório, irá corresponder por defeito ao User ID 4000.

```
;CALL OPERATOR
exten => 9, 1, Dial(SIP/4000, 5, g)

    same => n,GotoIf($[${DIALSTATUS}=ANSWER]?atende:notatende)

; OPERADOR ATENDEU
    same => n(atende),Goto(s,menu)

; OPERADOR NAO ATENDEU
    same => n(notatende), Playback(${PATH}/after-the-tone)
    same => n, Record(/tmp/msg.gsm,,k)
    same => n,Goto(s,menu)

;HANGUP CALL
exten => 0, 1, Hangup()

;INVALID EXTENSION
exten => i,1,Playback(${PATH}/were-sorry)

    same => n,Goto(s,menu)
```

Figura 3-Funcionalidades 9 e 0

- Implementação do jogo da tabuada

```

; JOGO DA TABUADA
exten => 1,1,Playback(${PATH}/tabuada)
same => n,Wait(1)
same => n,Set(one=${RAND(1,9)});
same => n,SayDigits(${one})
same => n,Wait(1)
same => n,Set(two=${RAND(1,9)});
same => n,SayDigits(${two})
same => n,Read(input,,2,,10)
same => n,GotoIf(${READSTATUS} = TIMEOUT)?naoescreveu:escreveu)

; IF THE USER DOES NOT INPUT
same => n(naoescreveu),Goto(s,menu)

; IF THE USER INPUTS
same => n(escreveu),SayNumber(${input})
same => n,Set(mult=${one}*${two});
same => n,SayNumber(${mult})
same => n,GotoIf([ "${input}" = "${mult}" ]?equal:notequal)

; IF THEY ARE EQUAL
same => n(equal),Set(GLOBAL(WON)=${WON} + 1)
same => n,Set(WON_CALL=${WON_CALL}+1);
same => n,Playback(${PATH}/ganhou)

same => n,Goto(s,menu)

; IF THEY ARE DIFFERENT
same => n(notequal),Set(GLOBAL(LOST)=${LOST} + 1)
same => n,Set(LOST_CALL=${LOST_CALL}+1);
same => n,Playback(${PATH}/perdeu)

same => n,Goto(s,menu)

```

Figura 4 - Implementação do jogo da tabuada

No código apresentado na Figura 4, são gerados dois números aleatórios de 1 a 9, que são transmitidos ao utilizador através da função *SayDigits*. Posteriormente, é requerido ao utilizador do telefone que introduza o resultado do produto entre estes dois números num espaço de dez segundos. O resultado irá posteriormente ser sujeito a uma comparação com o resultado correto, contido na variável *mult*.

No caso de os resultados serem iguais, são incrementadas as variáveis *WON* e *WON\_CALL*, que contêm, respetivamente, o número de jogos totais ganhos pelo jogador e o número de jogos ganhos pelo jogador na chamada atual. No caso de as variáveis diferirem no valor armazenado, segue-se um processo semelhante para as variáveis *LOST* e *LOST\_CALL*, que têm significados semelhantes às variáveis explicadas anteriormente, mas para os jogos perdidos.

- Indicação das estatísticas do jogo.

```

; STATISTICS
exten => 2,1,Playback(${PATH}/estatisticas)

same => n,Wait(1)
same => n,Playback(${PATH}/woncall)
same => n,Wait(1)
same => n,SayNumber(${WON_CALL})
same => n,Wait(1)
same => n,Playback(${PATH}/lostcall)
same => n,SayNumber(${LOST_CALL})
same => n,Wait(1)
same => n,Playback(${PATH}/wontotal)
same => n,SayNumber(${WON})
same => n,Wait(1)
same => n,Playback(${PATH}/losttotal)
same => n,SayNumber(${LOST})
same => n,Wait(1)
same => n,Playback(${PATH}/goodbye)
same => n,Goto(s,menu)

```

Figura 5 - Implementação da indicação das estatísticas de jogo

Através da implementação apresentada, o utilizador irá ter indicação das estatísticas de jogo de acordo com a seguinte ordem: número de jogos ganhos na chamada atual, número de jogos perdidos na chamada atual, número de jogos ganhos no total de todas as chamadas e número de jogos perdidos no total de todas as chamadas.

- Acesso ao serviço de *voicemail*

```
;VOICEMAIL
exten => 3, 1, Playback(${PATH}/password)

same => n,Read(pass,,,,,5)
same => n,GotoIf(${pass}=1234)?correct:incorrect)

same => n(correct),Playback(${PATH3}/msg)
same => n,Goto(s,menu)

same => n(incorrect),Goto(3,1)
same => n,Goto(3,1)
```

Figura 6 - Acesso ao serviço de voicemail

Após as configurações apresentadas, dever-se-á correr o comando *dialplan reload*. Após esta fase, cada utilizador que se liga ao servidor, passará assim a poder usufruir desta aplicação IVVR com todas as suas funcionalidades.

## Protocolo SIP

O protocolo SIP é um protocolo de sinalização multimédia *client-server*, utilizado para serviços de VoIP, e como tal, é um dos protocolos utilizados no nosso projeto. Este protocolo assume que a voz é transmitida por RTP, um protocolo que irá ser descrito na secção a seguir, e que realmente irá ser utilizado não só para a transmissão de voz, mas também para a transmissão de vídeo. Na utilização deste protocolo, o cliente realiza uma chamada telefónica e é o servidor quem a atende.

A necessidade de utilização deste protocolo, prende-se a necessidade de negociação dos *coding schemes* durante o *setup* da chamada. Este protocolo, apresenta, entre outras, as seguintes vantagens:

- Mantem a sinalização o mais simples possível;
- É um protocolo bastante flexível;
- Fornece a possibilidade de incluir informação não standard nas mensagens e de os utilizadores realizarem decisões inteligentes;
- O utilizador tem controlo da chamada, não havendo a necessidade de assinar serviços específicos com o operador.

A sintaxe das mensagens que permite a gestão das sessões entre participantes, é semelhante à sintaxe das mensagens HTTP.

Estas mensagens podem ter um de dois objetivos: pedidos de um cliente ao servidor, ou respostas (*status messages*) do servidor para o cliente. Em ambos os casos, as mensagens são constituídas por uma *start-line*, *headers* e *message body*. A *request-line* irá especificar qual o tipo de pedido que se pretende realizar, sendo especificado se este foi feito com ou sem sucesso. Os *headers*, irão especificar informação adicional sobre a mensagem, bem como os meios necessários para o transporte da mesma. O corpo da mensagem, irá descrever a sessão a estabelecer, bem como a descrição do tipo de mensagem multimédia.

Note-se que o objetivo deste protocolo não é a definição da estrutura nem do conteúdo da mensagem. Estas informações irão ser definidas pelo protocolo SDP.

Alguns dos métodos de *request* deste protocolo são os seguintes:

- INVITE – inicia a sessão e informa os intervenientes relativamente à sua participação, bem como do tipo de informação a transmitir;
- ACK – confirmação por parte do *client* da recepção da resposta do servidor;
- BYE – termina a sessão. Pode ser enviado pelas duas partes: cliente e servidor;
- OPTIONS – pedido por parte do cliente das opções que o servidor pode fornecer relativamente ao tipo de informação que pode ser enviada bem como ao tipo de resposta que será recebida após o envio de um INVITE;
- CANCEL – termina um *request* pendente.
- REGISTER – registo no servidor SIP e *login*;

Relativamente às respostas às mensagens de *request*, estas são compostas por um código de resposta, uma explicação textual e linhas de cabeçalho.

Os códigos de resposta são números de três dígitos, em que consoante o primeiro dígito, a informação de resposta tem o seguinte significado:

1. Informational;
2. Success;
3. Redirection;
4. Request failure;
5. Server Failure;
6. Global Failure.

Apenas a primeira resposta não é considerada como sendo final.

## Protocolo SDP

Este tipo de protocolo, à semelhança do que foi descrito anteriormente, irá auxiliar o protocolo SIP, descrito na secção anterior, sendo utilizado para a definição da *message body*. A estrutura do SDP pode ser observada na Figura 7.

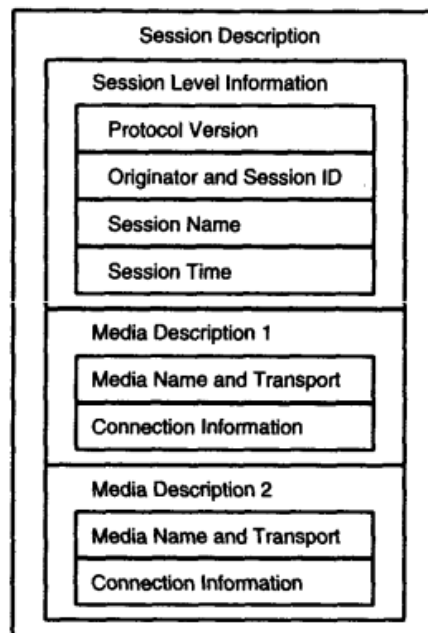


Figura 7 - Estrutura SDP

O protocolo SIP é assim utilizado para o estabelecimento de sessões multimédia, enquanto que o protocolo SDP será a linguagem estruturada que irá descrever estas sessões.

## Protocolo RTP

O protocolo RTP (Real-time Transport Protocol), é utilizado para a transferência de dados em tempo real. A utilização deste protocolo neste projeto, portanto bastante apropriada para a transmissão de áudio e vídeo. O protocolo, apesar de poder ser utilizado sobre qualquer protocolo de transporte, no nosso projeto corre em cima de UDP. O objetivo do protocolo, é de ajudar na deteção de pacotes perdidos e fora de ordem e conter o número de sequência no cabeçalho. Notar que este protocolo, apenas irá oferecer informação adicional, deixando o processamento da mesma a cargo da aplicação. Não oferece qualquer mecanismo que garanta a sequencialidade e a qualidade do serviço.

## Instruções de instalação do projeto

Por forma a instalar o projeto enviado no ficheiro *.zip*, deve-se seguir as seguintes instruções, assumindo que se está a executar o servidor numa máquina virtual Ubuntu e que os telefones X-Lite estão a ser executados no Windows:

1. Extrair o ficheiro para o Desktop do Ubuntu;
2. Abrir um terminal e entrar na pasta do Desktop que contém os ficheiros;
3. Copiar os ficheiros *extensions.conf*, *sip.conf* e *voicemail.conf* para as pastas do Asterisk (*/etc/asterisk*). Pode ser feito com recurso aos comandos *cp extensions.conf /etc/asterisk*, *cp sip.conf /etc/asterisk* e *cp voicemail.conf /etc/asterisk* no terminal. Outra maneira seria seleccionar os ficheiros com o file



manager (Files) em Home->Desktop e carregar com o botão do lado direito do rato escolher copy, entrar na diretoria que se pretende colar os ficheiros File System-> etc-> asterisk (/etc/asterisk) carregar com o botão do lado direito colar, deverá aparecer uma opção para substituir pelos ficheiros que já se encontram nesta pasta.

4. Alterar no ficheiro *extensions.conf* a variável global PATH que deverá conter o caminho para o programa ir buscar os ficheiros multimédia de voz e vídeo do projeto. Se for utilizada a máquina virtual sugerida pelo professor PATH deverá ter o seguinte aspecto: /home/ubuntu/Desktop/soundv1.
5. Por forma a correr o servidor, escrever o comando *sudo asterisk -vvvv*, seguido de *dialplan reload*, *sip reload* e *voicemail reload*, por forma a atualizar os planos de chamada e as configurações SIP de acordo com o nosso projeto;
6. Abrir as duas versões do X-Lite (User e operador) e selecionar *SIP account settings* e para o operador *softphone->account settings*;
7. No campo do user name, colocar 3001 para o telefone do user e 4000 para o telefone do operador;
8. No campo *domain*, colocar o endereço IP da máquina virtual, que pode ser verificado com o comando *ifconfig* no terminal do Ubuntu.

Observação: Os locais onde se encontram os ficheiros deverão ter as permissões necessárias para conseguir ler estes ficheiros. De modo a que isso seja possível é necessário executar, no terminal, o comando *sudo chmod 777 \** na diretoria onde se encontram estes ficheiros e nas pastas que os contêm. Se não conseguir copiar ou colar os ficheiros deverá também executar o comando anterior nas diretorias utilizadas.

## Ilustração das pilhas de protocolos utilizados

As pilhas com os protocolos utilizados são as seguintes:

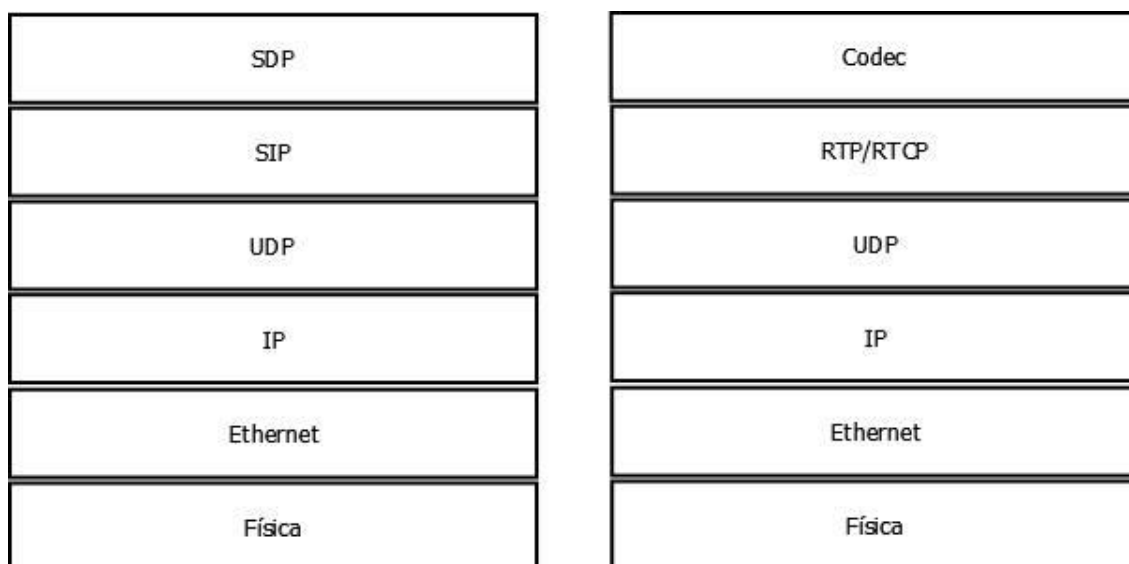


Figura 8 - Pilhas dos protocolos

No lado esquerdo da Figura 8 é apresentada a pilha de protocolos para a sinalização e no lado direito da figura é apresentada a pilha de protocolos para a transferência de fluxos de som e de vídeo.

## Observações no Wireshark

Com vista a verificar o funcionamento dos protocolos explicados nas secções anteriores, realiza-se uma série de capturas de pacotes através do *software Wireshark*. Assim, quando é inicializado o servidor, apenas estão disponíveis os pacotes relativos à troca de mensagens entre o computador e máquina virtual com informações relativas à internet.

Ao ligar o primeiro telefone X-Lite, com User ID 3001, é possível observar as seguintes trocas de pacotes:

No.	Time	Source	Destination	Protocol	Length	Info
43	22.554954	192.168.56.1	192.168.56.101	SIP	610	Request: REGISTER sip:192.168.56.101
44	22.654310	192.168.56.101	192.168.56.1	SIP	691	Status: 200 OK (1 bindings)
46	22.867440	192.168.56.1	192.168.56.101	SIP	613	Request: SUBSCRIBE sip:3001@192.168.56.101
47	22.869513	192.168.56.101	192.168.56.1	SIP	580	Status: 404 Not found (no mailbox)

Figura 9 – Arranque da aplicação do User 3001

Tal como se pode verificar pela Figura 9, para a ligação do telefone, é enviado um pacote SIP com método REGISTER, por forma a realizar o registo no servidor SIP, que neste caso é a versão 2.0. Como tal, verifica-se no request-line, o método, e o endereço IP do servidor (ver na Figura 10).

```
▼ Session Initiation Protocol
  ▼ Request-Line: REGISTER sip:192.168.56.101 SIP/2.0
    Method: REGISTER
  ▼ Request-URI: sip:192.168.56.101
    Request-URI Host Part: 192.168.56.101
    [Resent Packet: False]
```

Figura 10 - SIP request-line

Ao verificar o *message header*, verifica-se que a mensagem foi enviada pelo protocolo SIP sobre UDP, qual o endereço IP que enviou a mensagem e o respetivo porto. É também possível verificar a informação relativa ao contacto do utilizador que enviou o pedido. Esta informação é disposta na Figura 11.

```
▼ Contact: <sip:3001@192.168.56.1:1516;rinstance=d99711e23a4ba2f1>
  ▼ Contact-URI: sip:3001@192.168.56.1:1516;rinstance=d99711e23a4ba2f1
    Contactt-URI User Part: 3001
    Contact-URI Host Part: 192.168.56.1
    Contact-URI Host Port: 1516
    Contact parameter: rinstance=d99711e23a4ba2f1>
  ▼ To: "Halloween"<sip:3001@192.168.56.101>
    SIP Display info: "Halloween"
  ▶ SIP to address: sip:3001@192.168.56.101
  ▼ From: "Halloween"<sip:3001@192.168.56.101>;tag=ff2e4f1b
    SIP Display info: "Halloween"
  ▶ SIP from address: sip:3001@192.168.56.101
    SIP tag: ff2e4f1b
    Call-ID: 8c22631af8360432NDM5MGNhOWNjNTM4ZWZMTc3N2QxYzE4MmY3NjlkZDk.
  ▶ CSeq: 1 REGISTER
    Expires: 3600
    Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, NOTIFY, MESSAGE, SUBSCRIBE, INFO
    User-Agent: X-Lite release 1003l stamp 30942
    Content-Length: 0
```

Figura 11 - Informação adicional da mensagem de Request

Após o registo no servidor do novo utilizador, o servidor irá enviar uma mensagem de resposta com informação sobre o estado do pedido de registo. Esta mensagem

corresponde ao segundo pacote capturado na Figura 9. Sendo uma resposta de sucesso, tal como indicado na secção onde se descreve o protocolo SIP, o primeiro de três dígitos irá ser “2”. Tal como verificado, na Figura 12, o *status code* irá ser o 200, indicando desta forma que o pedido foi feito com sucesso. Para este tipo de pedidos, não é gerada nenhuma mensagem de *acknowledge*.

```
▼ Status-Line: SIP/2.0 200 OK
  Status-Code: 200
  [Resent Packet: False]
  [Request Frame: 8]
  [Response Time (ms): 2]
```

Figura 12 - Server response

Após o registo do novo user ter sido efetuado com sucesso, é possível verificar que este passa a estar monitored pelo servidor (ver Figura 13).

Name/username	Host
t ACL Port Status	
3000	(Unspecified)
Unmonitored	
3001/3001	192.168.56.1

Figura 13 - Início da aplicação telefónica do user 3001

Os pacotes seguintes verificados no *Wireshark*, são relativos a um pedido de subscrição de *voicemail*. Não estando este definido, seguem-se uma série de mensagens de *SUBSCRIBE* e de resposta de erro.

No.	Time	Source	Destination	Protocol	Length	Info
43	26.053460	192.168.56.1	192.168.56.101	SIP/SDP	1029	Request: INVITE sip:75573@192.168.56.101, with session description
44	26.055095	192.168.56.101	192.168.56.1	SIP	585	Status: 100 Trying
45	26.056359	192.168.56.101	192.168.56.1	SIP	601	Status: 180 Ringing
46	26.056750	192.168.56.101	192.168.56.1	SIP/SDP	929	Status: 200 OK, with session description
48	26.077637	192.168.56.1	192.168.56.101	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0xF213243C, Seq=6386, Time=28500, Mark
49	26.083565	192.168.56.101	192.168.56.1	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x155A63E6, Seq=2564, Time=160, Mark
50	26.096919	192.168.56.1	192.168.56.101	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0xF213243C, Seq=6387, Time=28660
...	...	...	...	...	...	...
174	27.363868	192.168.56.101	192.168.56.1	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x155A63E6, Seq=2628, Time=10400
175	27.384212	192.168.56.101	192.168.56.1	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x155A63E6, Seq=2629, Time=10560
176	27.394017	192.168.56.1	192.168.56.101	SIP	538	Request: BYE sip:75573@192.168.56.101:5060
177	27.395720	192.168.56.101	192.168.56.1	SIP	551	Status: 200 OK
189	32.106207	192.168.56.1	192.168.56.101	SIP	605	Request: REGISTER sip:192.168.56.101
190	32.107401	192.168.56.101	192.168.56.1	SIP	608	Status: 200 OK (0 bindings)

Figura 14 – Chamada do utilizador para o IVVR (com início e fim de chamada) e fim da aplicação telefónica

Ao ligar para a extensão correspondente ao número de aluno com número algebricamente inferior, é enviada uma mensagem de pedido ao servidor com o método de *INVITE*, onde no corpo da mensagem, é indicado o contacto que realizou o pedido, qual a extensão a que pretende aceder, entre outras (ver Figura 15).

```

▼ Contact: <sip:3001@192.168.56.1:49822>
  ▼ Contact-URI: sip:3001@192.168.56.1:49822
    Contactt-URI User Part: 3001
    Contact-URI Host Part: 192.168.56.1
    Contact-URI Host Port: 49822
  ▼ To: "75573"<sip:75573@192.168.56.101>
    SIP Display info: "75573"
  ▼ SIP to address: sip:75573@192.168.56.101
    SIP to address User Part: 75573
    SIP to address Host Part: 192.168.56.101
  ▼ From: "Halloween"<sip:3001@192.168.56.101>;tag=2904dc65
    SIP Display info: "Halloween"
  ▼ SIP from address: sip:3001@192.168.56.101
    SIP from address User Part: 3001
    SIP from address Host Part: 192.168.56.101
    SIP tag: 2904dc65
    Call-ID: 7a5e9834104eb457NDM5MGNhOWNjNTM4ZWM2MTc3N2QxYzE4MmY3NjlkZDk.
  ▼ CSeq: 1 INVITE

```

Figura 15 - Mensagem de INVITE

No *message body*, é apresentado o *Session Description Protocol*.

Posteriormente à receção do pedido do *client*, o *server* irá enviar três mensagens de resposta:

1. 100 Trying – informa que o servidor está a realizar uma procura para o pedido do cliente. Como tal, a informação é dada ao cliente através de um *forking proxy*;
2. 180 Ringing – o agente recebe o *INVITE* e dá ao *user* o alerta de chamada;
3. 200 OK, com descritor da sessão – o servidor informa o utilizador que o seu pedido foi realizado com sucesso.

Neste caso, como o pedido feito pelo utilizador foi do tipo *INVITE*, este, ao receber a mensagem *200 OK*, deverá responder com uma mensagem de *acknowledge*, por forma a confirmar ao servidor que recebeu a resposta final ao seu pedido de *INVITE*.

Ao ser desligada a chamada, o cliente irá enviar uma mensagem com o método *BYE*, recebendo uma resposta do servidor em como que o seu pedido foi completado com sucesso (*200 OK*).

Notar que sempre que é enviada a *session description*, além do protocolo SIP, é ainda utilizado o SDP para a descrição da sessão e que, entre o início e o fim da chamada, a transmissão de áudio e vídeo é feita pelo protocolo RTP.

Ao terminar a aplicação telefónica, é enviado um novo pacote de método REGISTER, por forma a informar o servidor que o *user* se desconectou, como pode ser observado com o comando *sip show peers* (ver Figura 16).

```

3001/3001 (Unspecified)
Unmonitored

```

Figura 16 - Fim da aplicação telefónica do user 3001

No caso de durante a chamada, o *User 3001* decidir ligar para o operador (*User 4000*), é enviada uma mensagem com descritor da sessão para o servidor, e este irá responder com as mensagens de resposta descritas anteriormente. O PBX, irá posteriormente comunicar com o user do operador, e verificar se os codecs de ambos são compatíveis. A informação destes será descrita pelas mensagens de *200 OK* que irão conter as informações dos codecs pelo SDP.

327	3.162691	192.168.56.101	192.168.56.1	SIP/SDP	1067 Request: INVITE sip:4000@192.168.56.1:56008;rinstance=a370de55b54192e4, with
333	3.251140	192.168.56.1	192.168.56.101	SIP	358 Status: 100 Trying
341	3.409827	192.168.56.1	192.168.56.101	SIP	521 Status: 180 Ringing
519	5.187322	192.168.56.1	192.168.56.101	RTP	214 PT=ITU-T G.711 PCMU, SSRC=0x66350149, Seq=25243, Time=1025917561, Mark
520	5.187654	192.168.56.1	192.168.56.101	RTP	214 PT=ITU-T G.711 PCMU, SSRC=0x66350149, Seq=25244, Time=1025917721
523	5.208084	192.168.56.1	192.168.56.101	RTP	214 PT=ITU-T G.711 PCMU, SSRC=0x66350149, Seq=25245, Time=1025917881
526	5.225373	192.168.56.1	192.168.56.101	RTP	214 PT=ITU-T G.711 PCMU, SSRC=0x66350149, Seq=25246, Time=1025918041
530	5.242265	192.168.56.1	192.168.56.101	RTP	214 PT=ITU-T G.711 PCMU, SSRC=0x66350149, Seq=25247, Time=1025918201
531	5.247976	192.168.56.1	192.168.56.101	SIP/SDP	942 Status: 200 OK, with session description
532	5.247976	192.168.56.1	192.168.56.101	SIP/XML	763 Request: INFO sip:3001@192.168.56.1:5060
533	5.247976	192.168.56.101	192.168.56.1	SIP	519 Request: ACK sip:4000@192.168.56.1:56008;rinstance=a370de55b54192e4
534	5.248191	192.168.56.101	192.168.56.1	SIP	561 Status: 200 OK

Figura 17 – Estabelecimento de ligação com o operador na chamada user-operador

No caso de os codecs serem compatíveis, o PBX irá enviar mensagens de INVITE para ambos os Users, para que eles possam comunicar de forma independente, sem haver congestionamento. No caso de não serem, será necessário que as informações passem pelo PBX que irá funcionar como tradutor entre ambas as partes.

Para o caso do projeto, os codecs são compatíveis, sendo visível na Figura 18 as duas mensagens de INVITE, para cada utilizador.

614	5.953797	192.168.56.101	192.168.56.1	SIP	477 Request: ACK sip:3001@192.168.56.1:10686
615	5.953972	192.168.56.101	192.168.56.1	SIP/SDP	900 Request: INVITE sip:3001@192.168.56.1:10686, in-dialog, with session descr.
616	5.954214	192.168.56.101	192.168.56.1	SIP/SDP	944 Request: INVITE sip:4000@192.168.56.1:63933;rinstance=128bd6dae45a8bd0, in
617	6.020316	192.168.56.1	192.168.56.101	SIP/SDP	820 Status: 200 OK, with session description
618	6.020844	192.168.56.101	192.168.56.1	SIP	519 Request: ACK sip:4000@192.168.56.1:63933;rinstance=128bd6dae45a8bd0
619	6.062918	192.168.56.1	192.168.56.101	SIP/SDP	792 Status: 200 OK, with session description
620	6.063283	192.168.56.101	192.168.56.1	SIP	477 Request: ACK sip:3001@192.168.56.1:10686

Figura 18 - Invites do PBX para users da chamada user-operador

Após o envio destas mensagens, cada um deles irá ter a informação do porto para o qual deverão enviar os conteúdos multimédia. No final da chamada, observa-se o envio dos pacotes BYE e 200 OK, indicando que o interveniente 4000 desligou a chamada anterior.

574	7.639155	192.168.56.1	192.168.56.101	SIP	533 Request: BYE sip:3001@192.168.56.1:5060
575	7.639657	192.168.56.101	192.168.56.1	SIP	560 Status: 200 OK
576	7.639944	192.168.56.101	192.168.56.1	SIP/SDP	906 Request: INVITE sip:3001@192.168.56.1:45512, in-dialog, with
577	7.652998	192.168.56.1	192.168.56.101	RTP	214 PT=ITU-T G.711 PCMU, SSRC=0xBBC97B21, Seq=7923, Time=1621880
578	7.672899	192.168.56.1	192.168.56.101	RTP	214 PT=ITU-T G.711 PCMU, SSRC=0xBBC97B21, Seq=7924, Time=1622040
579	7.692996	192.168.56.1	192.168.56.101	RTP	214 PT=ITU-T G.711 PCMU, SSRC=0xBBC97B21, Seq=7925, Time=1622200
580	7.712518	192.168.56.1	192.168.56.101	RTP	214 PT=ITU-T G.711 PCMU, SSRC=0xBBC97B21, Seq=7926, Time=1622360
581	7.733154	192.168.56.1	192.168.56.101	RTP	214 PT=ITU-T G.711 PCMU, SSRC=0xBBC97B21, Seq=7927, Time=1622520
582	7.747361	192.168.56.1	192.168.56.101	SIP/SDP	792 Status: 200 OK, with session description
583	7.747793	192.168.56.101	192.168.56.1	SIP	477 Request: ACK sip:3001@192.168.56.1:45512
584	7.748859	192.168.56.101	192.168.56.1	SIP	508 Request: BYE sip:3001@192.168.56.1:45512
586	7.855437	192.168.56.1	192.168.56.101	SIP	434 Status: 200 OK

Figura 19 - Fim da chamada user-operador

Após o fim desta chamada, é feito um INVITE por forma a manter a primeira chamada feita entre o User 3001 e o PBX, sendo que no final se observa a mesma troca de pacotes descrita anteriormente para o fim da chamada.

A captura ilustrada na Figura 20 corresponde agora a uma situação em que o operador não atende.

407	3.954562	192.168.56.101	192.168.56.1	SIP/SDP	1067 Request: INVITE sip:4000@192.168.56.1:56008;rinstance=a370de55b54192e4,
412	4.042841	192.168.56.1	192.168.56.101	SIP	358 Status: 100 Trying
424	4.259205	192.168.56.1	192.168.56.101	SIP	521 Status: 180 Ringing
930	9.312967	192.168.56.101	192.168.56.1	SIP	471 Request: CANCEL sip:4000@192.168.56.1:56008;rinstance=a370de55b54192e4
946	9.457267	192.168.56.1	192.168.56.101	SIP	480 Status: 200 OK
947	9.457302	192.168.56.1	192.168.56.101	SIP	429 Status: 487 Request Terminated
948	9.457768	192.168.56.101	192.168.56.1	SIP	519 Request: ACK sip:4000@192.168.56.1:56008;rinstance=a370de55b54192e4

Figura 20 - Ligação ao operador que não atende

Nesta situação, não tendo o operador atendido a chamada na janela temporal pedida, é enviada uma mensagem CANCEL por forma a cancelar o pedido pendente. É ainda

visível o pacote com *flag* 487 que indica que o pedido foi terminado, neste caso devido à ação de CANCEL (poderia ser também por BYE).

Finalmente, testa-se o caso em que o *user* liga para a extensão e coloca a chamada em espera e de seguida retoma a chamada. O caso descrito é apresentada na Figura 21.

No.	Time	Source	Destination	Protocol	Length	Info
489	4.784908	192.168.56.1	192.168.56.101	SIP/SDP	856	Request: INVITE sip:75573@192.168.56.101:5060, in-dialog, with session desc
490	4.785910	192.168.56.101	192.168.56.1	SIP	600	Status: 100 Trying
491	4.786016	192.168.56.101	192.168.56.1	SIP/SDP	931	Status: 200 OK, with session description
492	4.996266	192.168.56.1	192.168.56.101	SIP	498	Request: ACK sip:75573@192.168.56.101:5060
560	38.113281	192.168.56.1	192.168.56.101	SIP/SDP	861	Request: INVITE sip:75573@192.168.56.101:5060, in-dialog, with session desc
561	38.114280	192.168.56.101	192.168.56.1	SIP	600	Status: 100 Trying
562	38.114779	192.168.56.101	192.168.56.1	SIP/SDP	931	Status: 200 OK, with session description
585	38.330220	192.168.56.1	192.168.56.101	SIP	498	Request: ACK sip:75573@192.168.56.101:5060

Figura 21 - Colocação da chamada em espera e retoma da chamada

Como pode ser observado, é enviada uma mensagem de INVITE, seguida de TRYING, 200 OK e ACK. A razão da utilização destes métodos tendo já sido descrita anteriormente, não será repetida novamente. No entanto, ao analisar o conteúdo do *message body* das mensagens de INVITE, é possível observar que é mudado o endereço da ligação da chamada. Assim, quando a chamada é colocada em espera, o endereço de ligação passa a ser o 0.0.0.0, e quando a chamada é retomada, volta a ser o endereço anterior, 192.168.56.1. Ver Figura 22.

▼ Connection Information (c): IN IP4 0.0.0.0	▼ Connection Information (c): IN IP4 192.168.56.1
Connection Network Type: IN	Connection Network Type: IN
Connection Address Type: IP4	Connection Address Type: IP4
Connection Address: 0.0.0.0	Connection Address: 192.168.56.1

Figura 22 - Mudança de endereços ao colocar a chamada em espera e ao retomá-la

# Diagramas temporais

Nesta secção, irá ser sintetizada a informação da secção anterior, com recurso a diagramas temporais. O funcionamento do programa será então o seguinte:

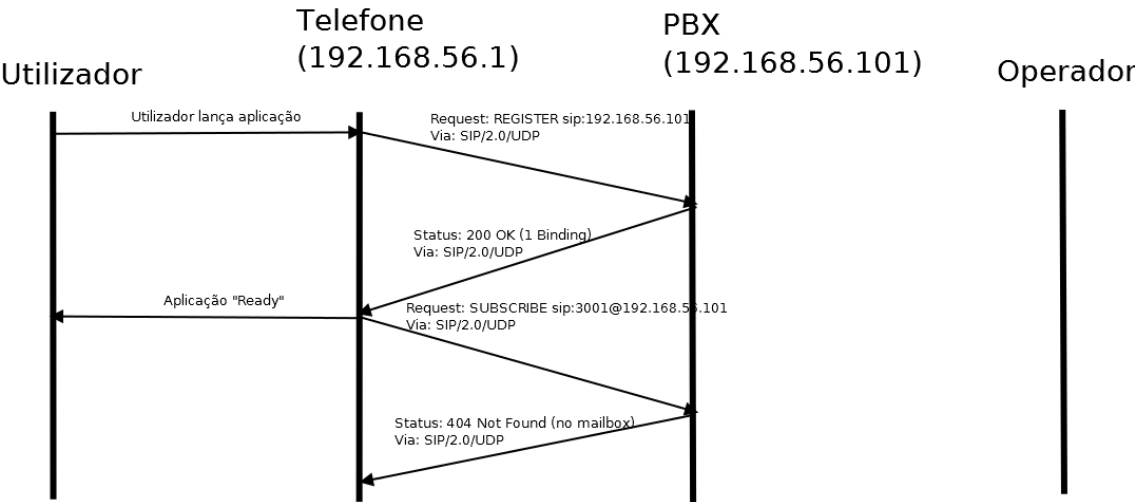


Figura 23 - Arranque da aplicação telefónica



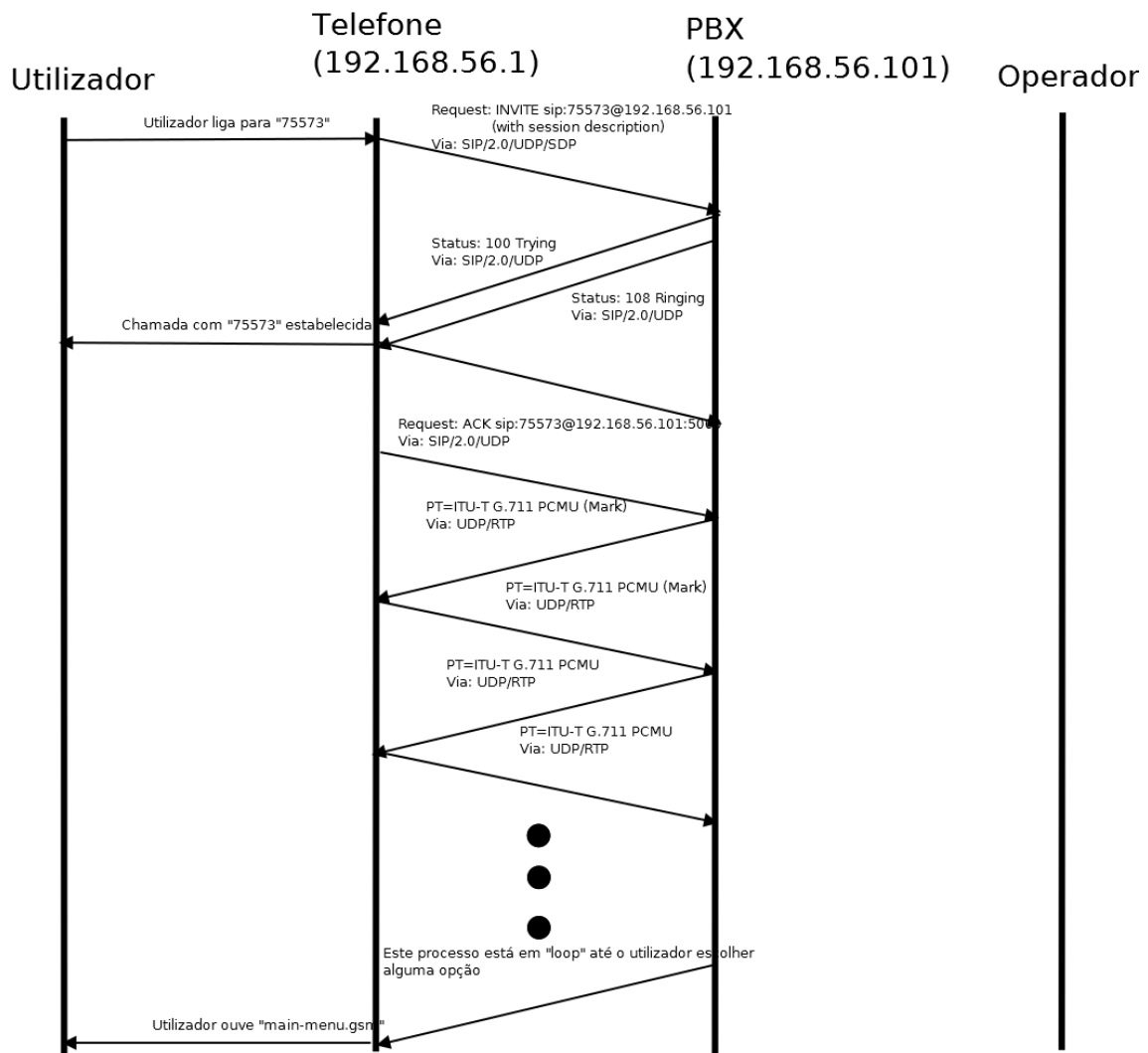


Figura 24 - Telefonema do utilizador para o sistema

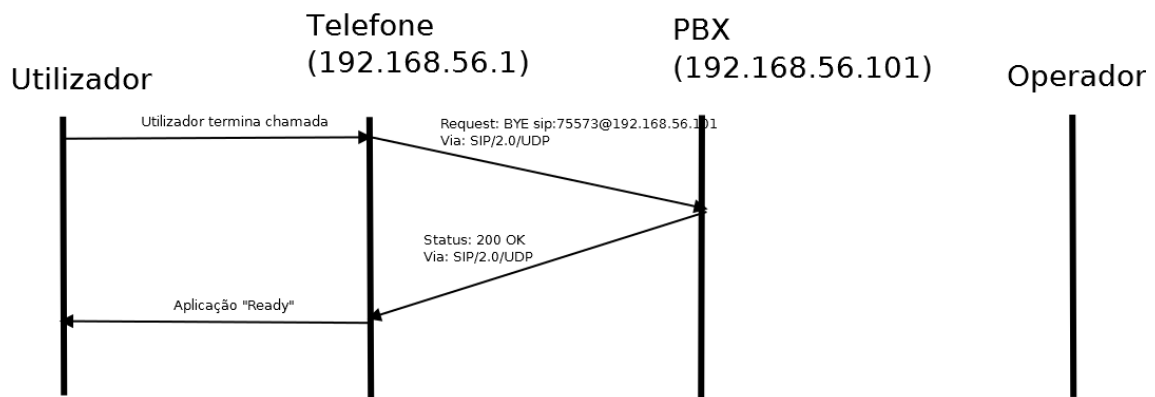


Figura 25 - Fim do telefonema do utilizador para o sistema

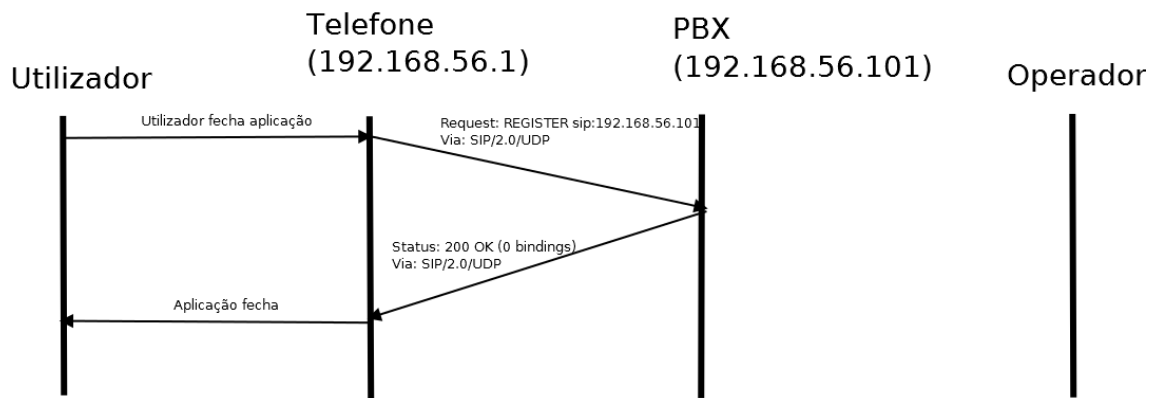


Figura 26 - Fecho da aplicação telefónica

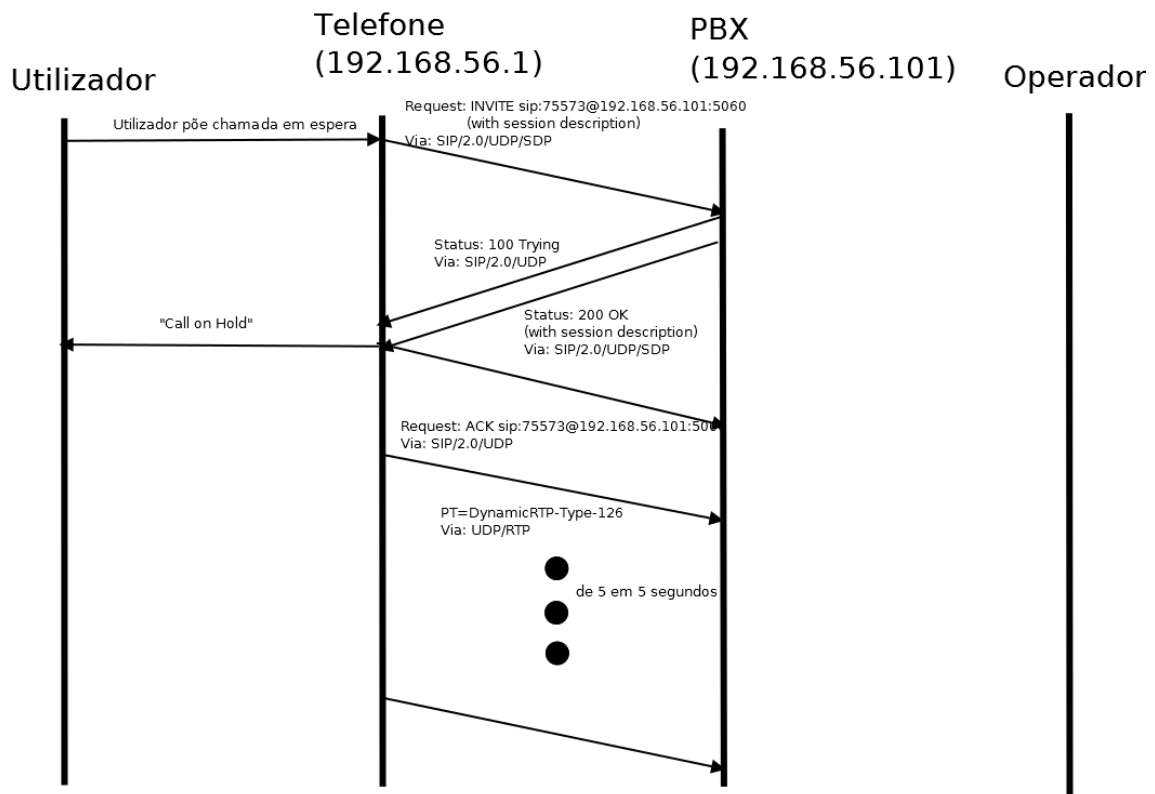


Figura 27 - Colocação da chamada em espera



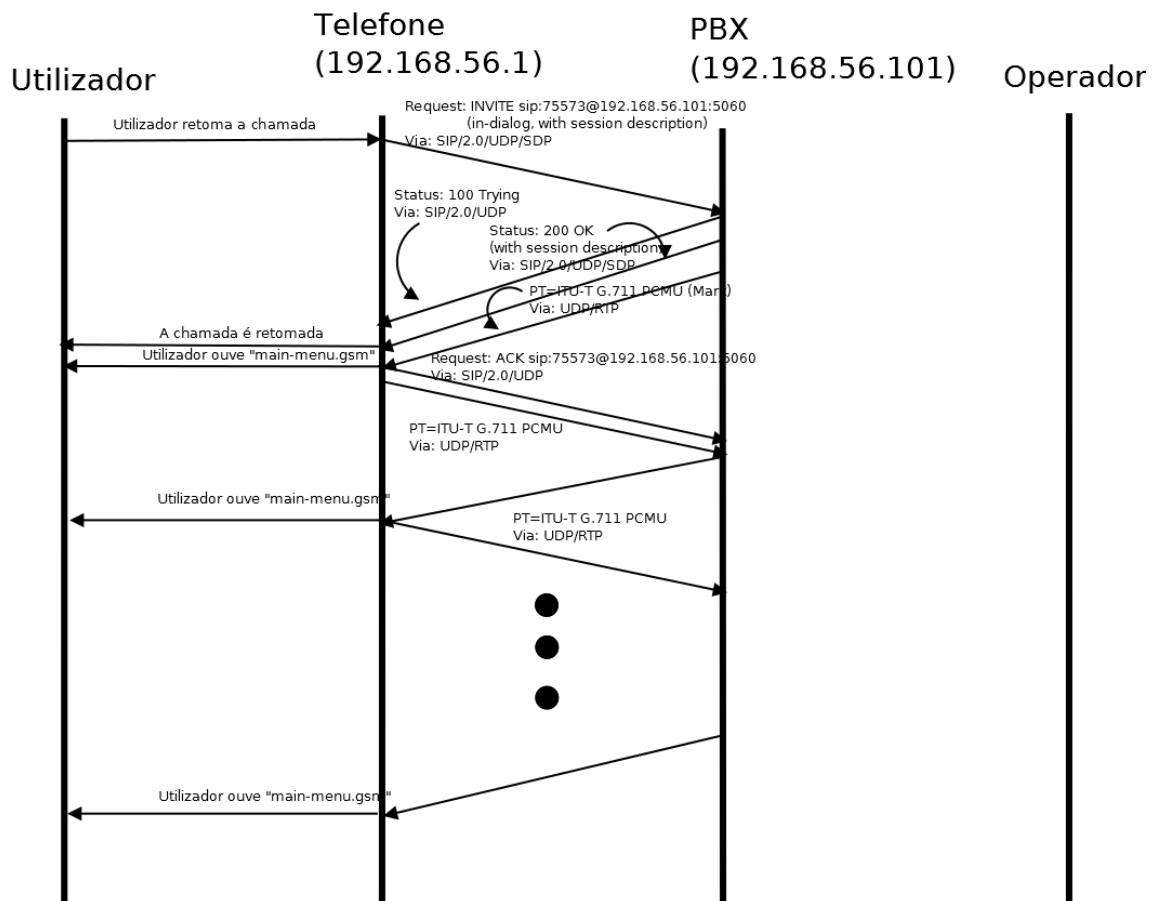


Figura 28 - Retoma da chamada em espera

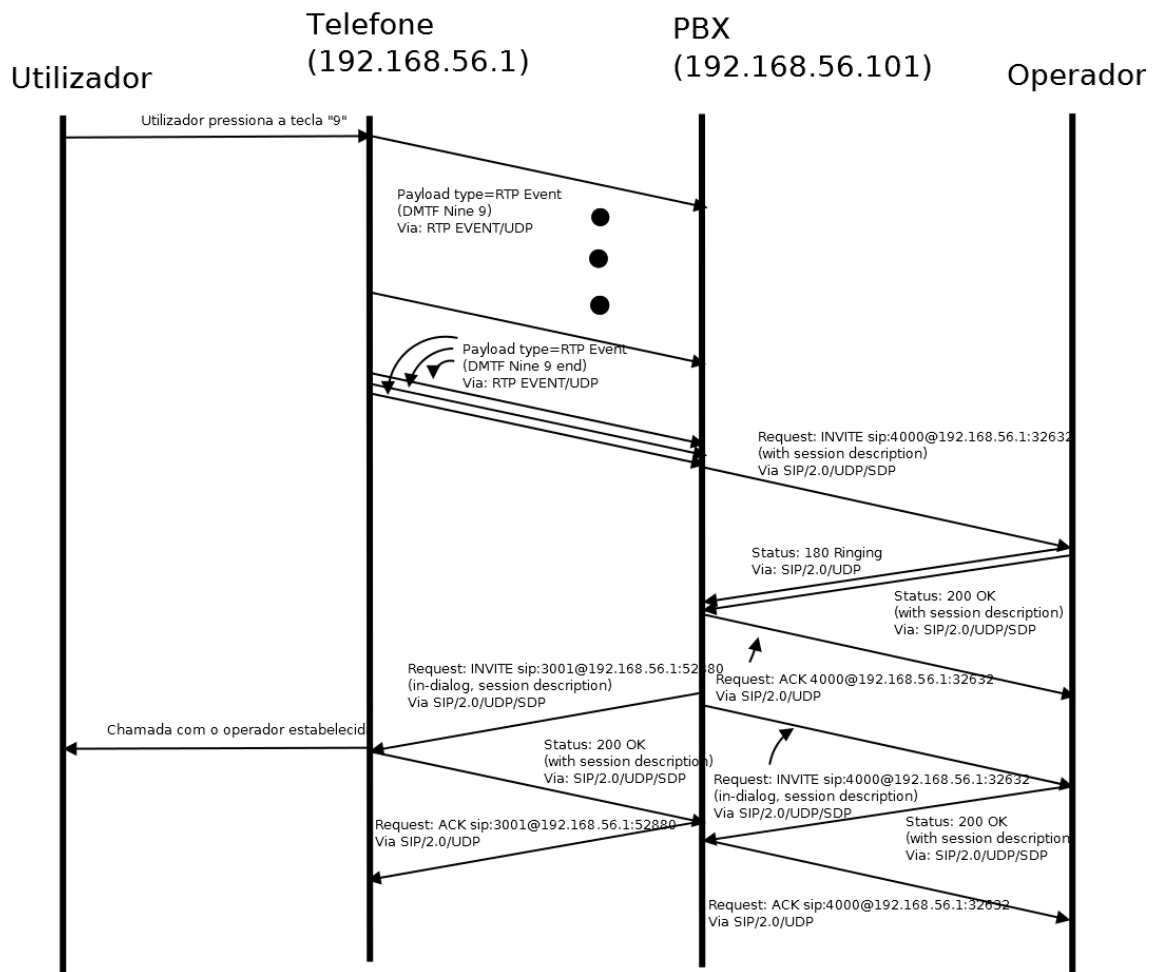


Figura 29 - User liga ao operador e este atende

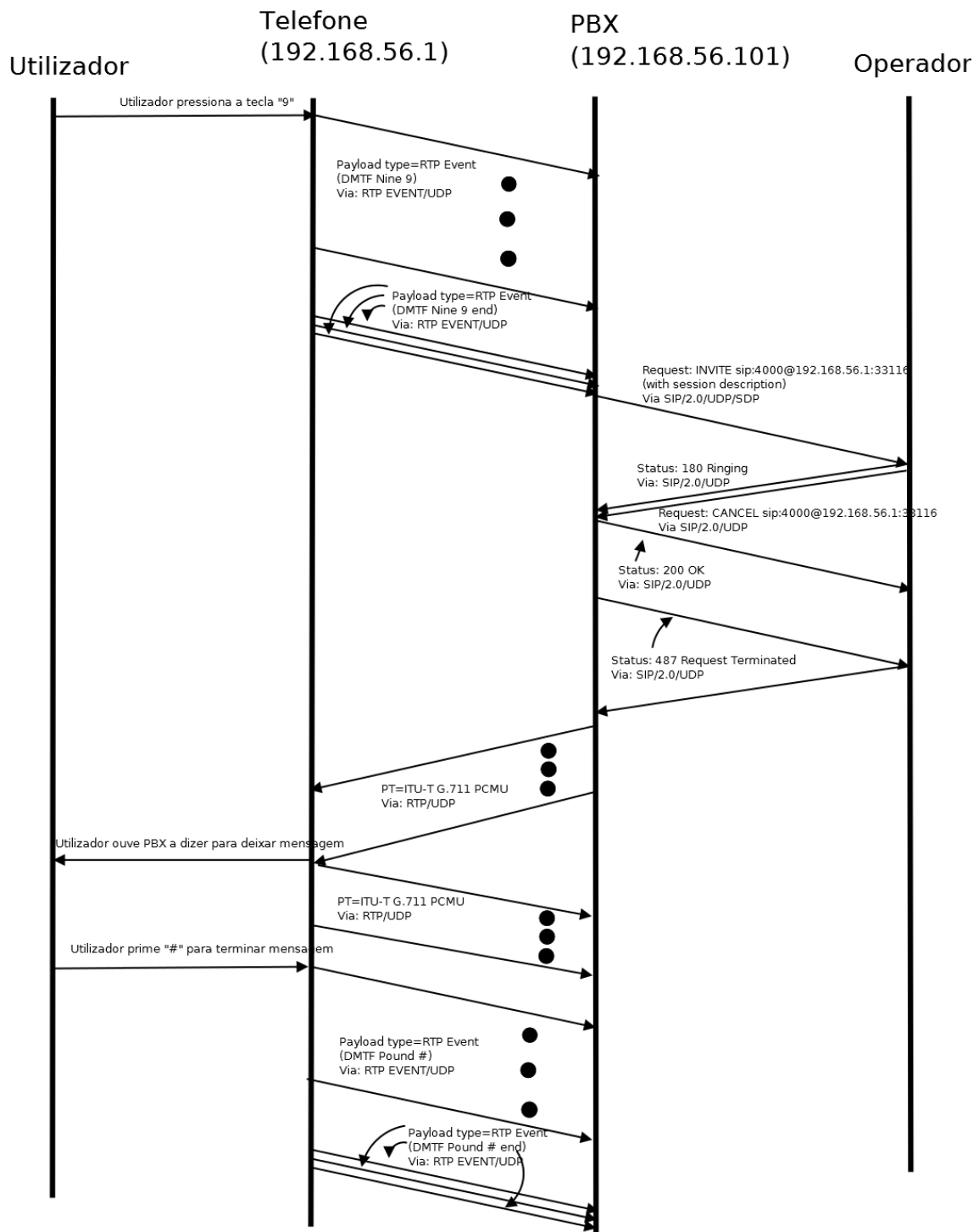


Figura 30 - User liga ao operador e este não atende

## Alterações no ficheiro sip.conf

Ao alterar as configurações SIP para *directmedia=no*, será de esperar que a sequência de operações executadas pelo PBX aquando da realização de uma chamada seja diferente.

No caso exposto na Figura 18, o PBX obteve os *codecs* de ambos os intervenientes da chamada, e pelo facto de serem compatíveis pôs os intervenientes a comunicar entre si, sem ser necessária a sua interferência.

Ao alterar as linhas referentes à transmissão de *media*, o comportamento do PBX será sempre independente dos *codecs* de cada *user*, optando pelo comportamento alternativo ao caso anterior em que toda a informação necessita de passar pelo servidor para ser transmitida. Simplemente no primeiro caso este caso apenas sucedia para tradução da informação enviada, mas neste segundo caso o comportamento adoptado será sempre o mesmo, levando eventualmente a um congestionamento de tráfego no caso de existirem vários utilizadores.

A captura realizada com o Wireshark para este caso pode ser observada na Figura 31.

No.	Time	Source	Destination	Protocol	Length	Info
427	4.090880	192.168.56.101	192.168.56.1	SIP/SDP	1065	Request: INVITE sip:4000@192.168.56.1:54425;rinstance=d232825486582770
432	4.174598	192.168.56.1	192.168.56.101	SIP	358	Status: 100 Trying
444	4.370002	192.168.56.1	192.168.56.101	SIP	521	Status: 180 Ringing
779	7.214801	192.168.56.1	192.168.56.101	SIP/SDP	942	Status: 200 OK, with session description
780	7.215154	192.168.56.101	192.168.56.1	SIP	519	Request: ACK sip:4000@192.168.56.1:54425;rinstance=d232825486582770
785	7.219737	192.168.56.1	192.168.56.101	SIP/XML	763	Request: INFO sip:3001@192.168.56.101:5060
786	7.219748	192.168.56.1	192.168.56.101	SIP/SDP	942	Status: 200 OK, with session description
787	7.219922	192.168.56.101	192.168.56.1	SIP	561	Status: 200 OK
788	7.220111	192.168.56.101	192.168.56.1	SIP	519	Request: ACK sip:4000@192.168.56.1:54425;rinstance=d232825486582770
5298	27.691233	192.168.56.1	192.168.56.101	SIP	533	Request: BYE sip:3001@192.168.56.101:5060
5299	27.691525	192.168.56.101	192.168.56.1	SIP	560	Status: 200 OK
6457	38.975174	192.168.56.1	192.168.56.101	SIP	538	Request: BYE sip:75573@192.168.56.101:5060
6458	38.975456	192.168.56.101	192.168.56.1	SIP	551	Status: 200 OK

Figura 31 - Captura de uma chamada com *directmedia=no*

Como mostrado na figura, o contacto inicial com o servidor é feito de forma semelhante, no entanto, após o ACK da ligação do primeiro interveniente, é enviado um pacote INFO, relativo a *mid session information*, que não altera o estado da mesma. Após a confirmação por parte dos *users*, é dado início à chamada, sendo o PBX a “ponte” entre os intervenientes da chamada. Estando a opção de chamada directa inviabilizada pelas configurações SIP, torna-se desnecessário o envio das duas mensagens de INVITE verificadas no caso anterior.

Os diagramas temporais para a situação apresentada mantêm-se iguais à excepção do caso em que o user liga para o operador e este atende, sendo o diagrama o apresentado na Figura 32.

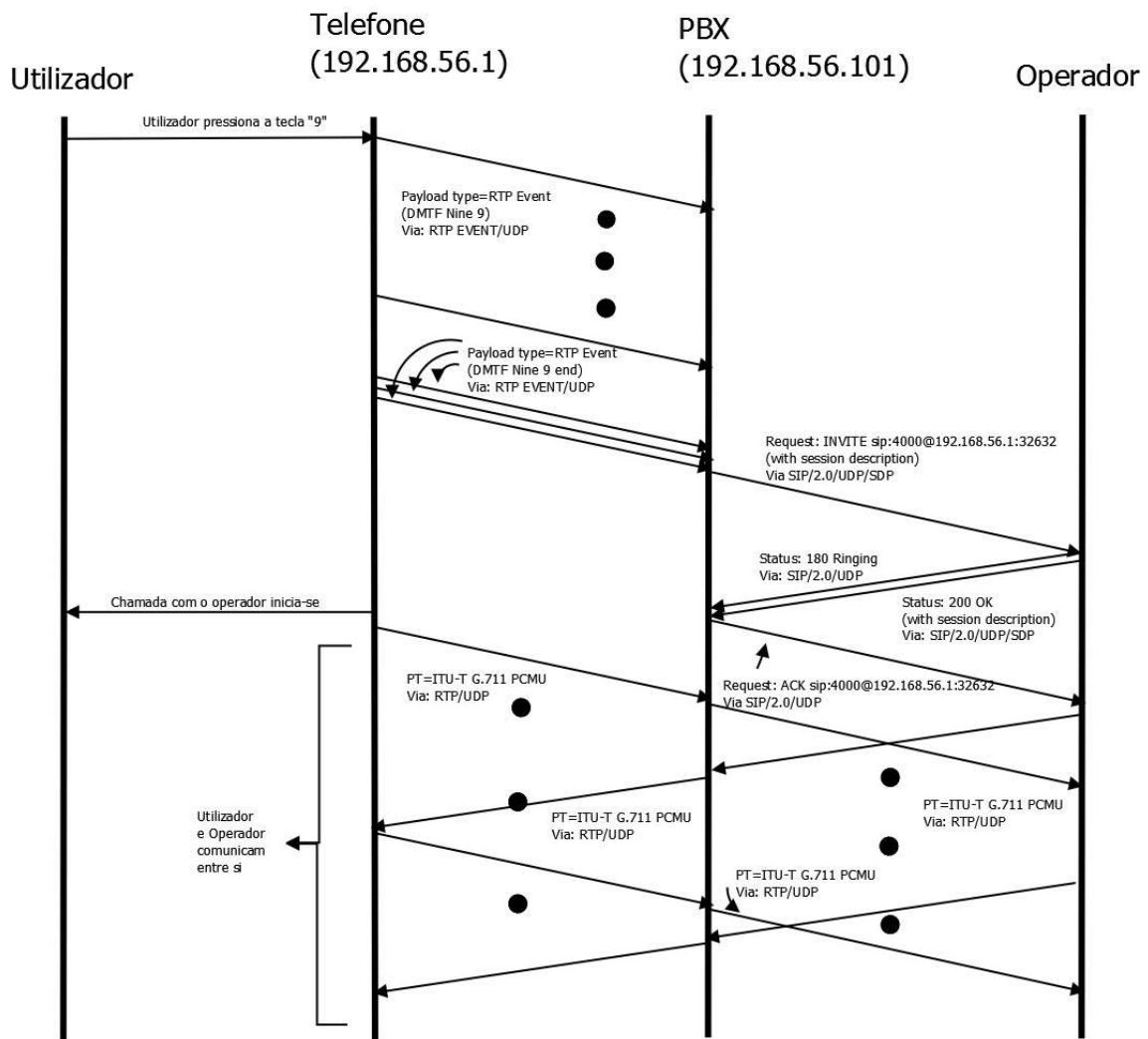


Figura 32 - Chamada entre user e operador, quando o operador atende, com *directmedia=no*

## Conclusões

Com o projecto realizado, foi possível compreender melhor os protocolos SIP, RTP, SDP, bem como o funcionamento e configuração de um PBX baseado na plataforma Asterisk. Foi ainda possível observar as alterações de comportamento do PBX consoante as configurações SIP de transmissão de *media*, bem como as alterações daí provenientes no envio de *request methods*.