

Instituto Superior Técnico

Projecto VoIP

Sistema IVVR

68239 – Rui Barradas

68477 – Helton Miranda

68626 – Ludijor Barros

72487 – Bruna Gondin

Introdução

O objectivo deste projecto é desenvolver um sistema *de Interactive Video Voice Response* (IVVR) baseado em voz sobre IP (Voice over IP – VoIP) e compreender os protocolos utilizados. Os protocolos utilizados no desenvolvimento do projecto foram: SIP (usado para a sinalização), SDP (faz a descrição dos media trocados entre as duas partes), RTP (usado para a transferência de dados de tempo real), RTCP (baseado no envio periódico de pacotes de controlo a todos os participantes na sessão, que nesse caso é a chamada), e UDP (tem a função de transportar os pacotes, mas, como sabemos, o UDP não garante qualidade de serviço). Houve implementação prática do projecto e, para descrição, no relatório utilizou-se diagramas temporais e pilhas de protocolos a fim de melhor demonstrar o funcionamento do projecto implementado.

Instalação do projecto

Inicialmente foi preciso instalar o asterisk (PBX), e para tal utilizamos o seguinte comando na linha de comandos do Linux: *sudo apt-get install asterisk*. No entanto, esse comando instalou a versão 1.6 do asterisk, que teve alguns problemas com o vídeo. Por isso, para efeitos de testes mais precisos utilizamos o computador do laboratório (que já tinha o PBX instalado, na versão 1.4 do asterisk). De seguida, substituímos os ficheiros *extensions.conf* e *sip.conf* pelos que tínhamos criado, utilizando os seguintes comandos:

- *mv extensions.conf '/etc/asterisk/extensions.conf'*.
- *mv sip.conf '/etc/asterisk/sip.conf'*.

Como podemos ver, os ficheiros *extensions.conf* e *sip.conf* foram os únicos ficheiros do asterisk que alteramos. Depois gravamos alguns ficheiros de som e de vídeo (utilizando o voicemail) e adicionamos os mesmos à pasta sounds do asterisk fazendo:

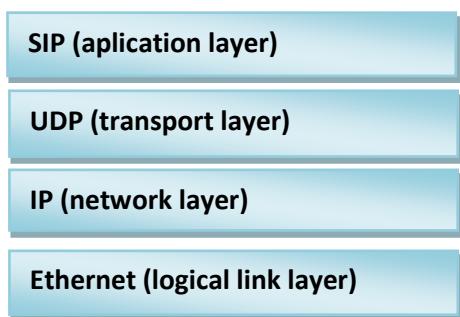
- *cd /var/spool/asterisk/voicemail/default/1234/INBOX*
- *mv * '/usr/share/asterisk/sounds/'*

Também instalamos as duas versões do X-Lite (3 e 4) e verificamos que para esta última o vídeo não funcionou.

Pilha de Protocolos

- **Sinalização**

Para a sinalização, como já foi dito, é utilizado o protocolo SIP (Session Initiation Protocol) que permite gerir sessões entre um conjunto de participantes. Como sabemos, o protocolo SIP tem quatro tipos de servidores. Neste projecto utilizamos um servidor proxy que actua de forma semelhante a um servidor proxy na Web, ou seja, gera pedidos ou envia pedidos a outros servidores. Pode ser usado para reenvio de chamadas. A vantagem de utilizar o SIP na sinalização é que oferece grande flexibilidade em relação aos outros protocolos, uma vez que não se preocupa, por exemplo, nem com o tipo de media trocado durante a sessão (a chamada), nem com a forma como os mesmos são transportados.



- **Transferências de fluxos de som e vídeo**

Para a transferência de fluxos de som e vídeo é utilizada o protocolo RTP e RTCP. O RTP providencia serviços de transporte na rede que são essenciais em aplicações de tempo real como tráfego de voz ou vídeo. O RTP contém campos como o “sequence number”, útil para as aplicações que utilizam RTP conseguirem detectar perdas de pacotes e também conseguirem verificar se os pacotes chegaram na ordem correcta, uma vez que o RTP normalmente corre sobre o UDP, e este último protocolo não tem garantia de qualidade de serviço. O RTCP providencia um conjunto de mensagens que são trocadas entre os utilizadores numa sessão que permitem obter um feedback em relação à qualidade da sessão. Sempre que é iniciada uma sessão RTP é também

iniciada uma sessão RTCP, mas como já dissemos as sessões RTCP apenas fornecem dados estatísticos.

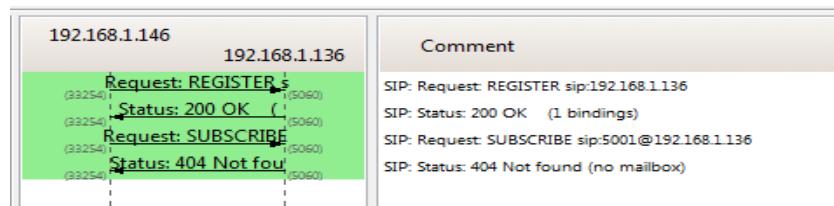


Diagrama Temporais

Neste ponto vamos explicar o funcionamento dos protocolos utilizados com base nos diagramas temporais mostrando nas mensagens os parâmetros mais relevantes.

SIP, RTP e RTCP

- Registo



Parâmetros da primeira mensagem:

REGISTER sip:192.168.1.136 SIP/2.0

Via: SIP/2.0/UDP 192.168.1.146:33254

From: "5001"<<sip:5001@192.168.1.136>>

To: "5001"<<sip:5001@192.168.1.136>>

O primeiro request enviado pelo cliente deve ser registrado (REGISTER), porque este request é que permite ao servidor saber o endereço do utilizador (usado pela sessão SIP). Como podemos ver na figura acima, o utilizador 192.168.1.146 envia um Request ao PBX (192.168.1.136) e, como foi o primeiro, esse utilizador deve ser registrado, por isso utiliza o método REGISTER. Para confirmar que o pedido do utilizador foi aceite,

o PBX envia uma mensagem de resposta ao utilizador, código de resposta (200) e uma explicação textual (OK) indicando o sucesso do registo do mesmo.

- Estabelecimento da chamada

| 192.168.1.146 | 192.168.1.136 | Comment |
|---|---------------|--|
| Request: INVITE sip:68239@192.168.1.136 (42346) | | SIP/SDP: Request: INVITE sip:68239@192.168.1.136, with session description |
| Status: 100 Trying (42346) | | SIP: Status: 100 Trying |
| Status: 200 OK, with session description (42346) | | SIP/SDP: Status: 200 OK, with session description |
| Request: ACK sip:68239@192.168.1.136:5060 (42346) | | SIP: Request: ACK sip:68239@192.168.1.136:5060 |

O estabelecimento da chamada começa com uma mensagem SIP INVITE, a “convidar” o destinatário à participar numa sessão, que, nesse caso, é uma chamada. Depois de receber essa mensagem o PBX envia uma mensagem de resposta ao utilizador com o código 100 e um campo textual (Trying) dizendo ao mesmo que está tentando estabelecer a ligação. Assim que a ligação é estabelecida com sucesso o PBX envia uma “mensagem” 200 OK ao utilizador. De seguida, o utilizador responde com um ACK e, a partir desse instante (só a partir desse instante) a ligação é dada como estabelecida.

Parâmetros da primeira mensagem:

```
INVITE sip:68239@192.168.1.136 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.146:43346
From: "5001"<sip:5001@192.168.1.136>
Contact: <sip:5001@192.168.1.146:43346>
```

SDP na mensagem INVITE:

```
v=0
o=- 2 2 IN IP4 192.168.1.146
s=CounterPath eyeBeam 1.5
c=IN IP4 192.168.1.146
t=0 0
m=audio 57148 RTP/AVP 0 8 3 101
a=alt:1 1 : JO1A34fm WmeSyMjz 192.168.1.146 57148
a=fmtp:101 0-15
a=rtpmap:101 telephone-event/8000
a=sendrecv
a=x-rtp-session id:581E95516ADF4D20A7BC1FCE7FE64766
m=video 15622 RTP/AVP 115 34
a=alt:1 1 : msd0JtgB /7OR7I4S 192.168.1.146 15622
a=fmtp:115 QCIF=1 MAXBR=1960
a=fmtp:34 QCIF=1 MAXBR=1960
a=rtpmap:115 H263-1998/90000
a=rtpmap:34 H263/90000
a=sendrecv
a=x-rtp-session-id:279AA745B4E64E2EABD31D62E42484D2
```

SDP na mensagem OK

```
v=0
o=root 65557230 65557230 IN IP4 192.168.1.136
s=Asterisk PBX 1.8.4.4~dfsg-2ubuntu1
c=IN IP4 192.168.1.136
b=CT:384
t=0 0
m=audio 16330 RTP/AVP 3 0 8 101
a=rtpmap:3 GSM/8000
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=ptime:20
a=sendrecv
m=video 14432 RTP/AVP 34
a=rtpmap:34 H263/90000
a=sendrecv
```

Os campos v,o,s,c,b,t,m,a são os campos das mensagens SDP. O campo m, por exemplo, contém informações sobre o media utilizado (o tipo, o formato, e o porto, que são os mais importantes).

- **Término da Sessão**

| 192.168.1.146 | 192.168.1.136 | 192.168.1.133 | 192.168.1.144 | Comment |
|---|---------------|---------------|---------------|---|
| Request: BYE sip:68 (43346) (5060) Status: 200 OK (43346) (5060) | | | | SIP: Request: BYE sip:68239@192.168.1.136:5060 SIP: Status: 200 OK |

Estas são as mensagens trocadas quando sempre que um utilizador pretende terminar uma sessão. Como podemos ver, para terminar a sessão o utilizador envia uma mensagem BYE, indicando ao PBX que pretende terminar a sessão. Depois o PBX responde com um 200 OK, indicando que a sessão foi terminada com sucesso.

- **Ligaçao entre dois utilizadores (o utilizador quer falar com o operador que está na extensão 4000)**

No PBX – IP 192.168.2.99: O IP dos utilizadores são diferentes pois a captura destes pacotes não ocorreu na mesma altura, nem nos mesmos PC's.

- **Captura no PBX**

| 192.168.2.167 | 192.168.2.169 | Comment |
|--|---------------|--|
| Request: INVITE sip: (5151) (5060) Status: 100 Trying (5151) (5060) Status: 200 OK, wit (5151) (5060) PT=ITU-T G.711 PCMU (23408) (51422) PT=ITU-T G.711 PCMU (23408) (51422) Request: ACK sip:68 (5151) (5060) PT=DynamicRTP-Type- (6024) (12958) Payload type=RTP Pv (23408) (51422) PT=ITU-T G.711 PCMU (23408) (51422) Request: INVITE sip: (5151) (5060) Status: 180 Ringing (5151) (5060) Status: 200 OK, wit (5151) (5060) Request: ACK sip:40 (5151) (5060) Request: INVITE sip: (5151) (5060) Request: INVITE sip: (5151) (5060) PT=ITU-T G.711 PCMU (19728) (51302) PT=ITU-T G.711 PCMU (19728) (51302) PT=ITU-T G.711 PCMU (23408) (51422) Status: 200 OK, wit (5151) (5060) Request: ACK sip:50 (5151) (5060) Status: 200 OK, wit (5151) (5060) Request: ACK sip:40 (5151) (5060) Request: INVITE sip: (5151) (5060) Status: 200 OK, wit (5151) (5060) Request: ACK sip:50 (5151) (5060) Request: BYE sip:68 (5151) (5060) Status: 200 OK (5151) (5060) Request: INVITE sip: (5151) (5060) PT=ITU-T G.711 PCMU (19728) (51302) Status: 200 OK, wit (5151) (5060) Request: ACK sip:40 (5151) (5060) Request: BYE sip:40 (5151) (5060) Status: 200 OK (5151) (5060) | | SIP/SDP: Request: INVITE sip:68239@192.168.2.99, with session description SIP: Status: 100 Trying SIP/SDP: Status: 200 OK, with session description RTP: PT=ITU-T G.711 PCMU, SSRC=0x14609DEC, Seq=39105, Time=160 RTP: PT=ITU-T G.711 PCMU, SSRC=0x8C5525D9, Seq=3029, Time=2869100, Mark SIP: Request: ACK sip:68239@192.168.2.99 RTP: PT=DynamicRTP-Type=126, SSRC=0x17CF8C70, Seq=5987, Time=0 RTP: PT=DynamicRTP-Type=126, SSRC=0x17CF8C70, Seq=5987, Time=0 RTP EVENT: Payload type=RTP Event, DTMF Nine 9 RTP: PT=ITU-T G.711 PCMU, SSRC=0x8C5525D9, Seq=3319, Time=2915500 SIP/SDP: Request: INVITE sip:4000@192.168.2.169:21336;instance=25da9aeda3a619a2, with session desc SIP: Status: 180 Ringing SIP/SDP: Status: 200 OK, with session description SIP: Request: ACK sip:4000@192.168.2.169:21336;instance=25da9aeda3a619a2 SIP/SDP: Request: INVITE sip:5001@192.168.2.167:51513, with session description SIP/SDP: Request: INVITE sip:4000@192.168.2.169:21336;instance=25da9aeda3a619a2, with session desc RTP: PT=ITU-T G.711 PCMU, SSRC=0x3528427F, Seq=35932, Time=2940776 RTP: PT=ITU-T G.711 PCMU, SSRC=0x1FD4B127, Seq=3247, Time=3267900, Mark RTP: PT=ITU-T G.711 PCMU, SSRC=0x14609DEC, Seq=39554, Time=3267896 SIP/SDP: Status: 200 OK, with session description SIP: Request: ACK sip:5001@192.168.2.167:51513 SIP/SDP: Status: 200 OK, with session description SIP: Request: ACK sip:4000@192.168.2.169:21336;instance=25da9aeda3a619a2 SIP/SDP: Request: INVITE sip:5001@192.168.2.167:51513, with session description SIP/SDP: Status: 200 OK, with session description SIP: Request: ACK sip:5001@192.168.2.167:51513 SIP: Request: BYE sip:68239@192.168.2.99 SIP: Status: 200 OK SIP/SDP: Request: INVITE sip:4000@192.168.2.169:21336;instance=25da9aeda3a619a2, with session desc RTP: PT=ITU-T G.711 PCMU, SSRC=0x1FD4B127, Seq=3202, Time=3420700 SIP/SDP: Status: 200 OK, with session description SIP: Request: ACK sip:4000@192.168.2.169:21336;instance=25da9aeda3a619a2 SIP: Request: BYE sip:4000@192.168.2.169:21336;instance=25da9aeda3a619a2 SIP: Status: 200 OK |

- **No Utilizador 5001 – IP 192.168.1.146 (Utilizador 1)**

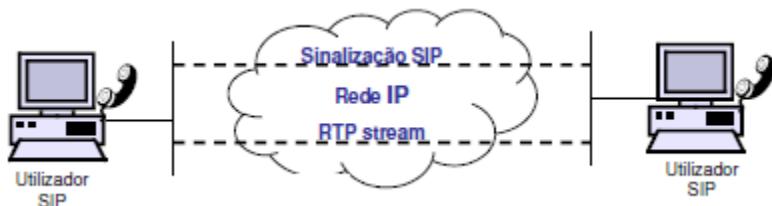
| 192.168.1.146 | 192.168.1.151 | Comment |
|---|---------------|---|
| Request: INVITE sip: PT=ITU-T G.711 PCMU | | SIP/SDP: Request: INVITE sip:5001@192.168.1.146:33254, with session description |
| (46612) [5060] (17096) [5060] | | RTP: PT=ITU-T G.711 PCMU, SSRC=0x3B7364BE, Seq=36, Time=344580, Mark |
| PT=ITU-T G.711 PCMU | | RTCP: PT=ITU-T G.711 PCMU, SSRC=0xB21988C, Seq=11372, Time=4700540 |
| (46612) [5060] (17096) [5060] | | SIP/SDP: Status: 200 OK, with session description |
| Status: 200 OK, with session description | | SIP: Request ACK sip:5001@192.168.1.146:33254 |
| Request: ACK sip:50 | | RTP: PT=ITU-T G.711 PCMU, SSRC=0x3B7364BE, Seq=42, Time=345540 |
| (46612) [5060] (17096) [5060] | | RTCP: Receiver Report: Source description: Goodbye |
| PT=ITU-T G.711 PCMU | | SIP/SDP: Request: INVITE sip:5001@192.168.1.146:33254, with session description |
| (46612) [5060] (17096) [5060] | | RTP: PT=ITU-T G.711 PCMU, SSRC=0xB21988C, Seq=11383, Time=4702300 |
| Receiver Report: S | | RTCP: Sender Report: Source description |
| (46612) [5060] (17096) [5060] | | SIP/SDP: Status: 200 OK, with session description |
| Request: INVITE sip: PT=ITU-T G.711 PCMU | | SIP: Request ACK sip:5001@192.168.1.146:33254 |
| (46612) [5060] (17096) [5060] | | RTP: PT=ITU-T G.711 PCMU, SSRC=0x3B7364BE, Seq=53, Time=347300 |
| Sender Report: Sou | | RTCP: Sender Report: Source description |
| (46612) [5060] (17096) [5060] | | RTCP: Receiver Report: Source description |
| Status: 200 OK, with session description | | SIP/SDP: Status: 200 OK, with session description |
| Request: ACK sip:50 | | SIP: Request ACK sip:5001@192.168.1.146:33254 |
| (46612) [5060] (17096) [5060] | | RTP: PT=ITU-T G.711 PCMU, SSRC=0x3B7364BE, Seq=53, Time=347300 |
| PT=ITU-T G.711 PCMU | | RTCP: Sender Report: Source description |
| (46612) [5060] (17096) [5060] | | RTCP: Receiver Report: Source description |
| Sender Report: Sou | | SIP/SDP: Status: 200 OK, with session description |
| (46612) [5060] (17096) [5060] | | RTCP: Sender Report: Source description |
| Receiver Report: S | | RTCP: Receiver Report: Source description |
| (46612) [5060] (17096) [5060] | | SIP/SDP: Status: 200 OK, with session description |
| Sender Report: Sou | | RTCP: Sender Report: Source description |
| (46612) [5060] (17096) [5060] | | RTCP: Receiver Report: Source description |
| Receiver Report: S | | RTP: PT=DynamicRTP-Type-126, SSRC=0x74612B38, Seq=2792, Time= |
| (46612) [5060] (17096) [5060] | | RTCP: PT=DynamicRTP-Type-126, SSRC=0x74612B38, Seq=2792, Time= |
| PT=DynamicRTP-Type- | | |
| (46612) [5060] (17096) [5060] | | |

- **No Operador 4000 – IP 192.168.1.151 (Utilizador 2)**

| 192.168.1.136 | 192.168.1.146 | Comment |
|---|---------------|--|
| Request: INVITE sip: Status: 180 Ringing | | SIP/SDP: Request: INVITE sip:4000@192.168.1.151:38100;instance=45348497a0ea7a62, with session desc |
| (5060) (38100) [5060] (38100) | | SIP: Status: 180 Ringing |
| Status: 200 OK, with session description | | SIP/SDP: Status: 200 OK, with session description |
| (5060) (38100) [5060] (38100) | | SIP: Request: ACK sip:40 |
| Request: ACK sip:40 | | SIP/SDP: Request: INVITE sip:4000@192.168.1.151:38100;instance=45348497a0ea7a62, with session desc |
| (5060) (38100) [5060] (38100) | | RTP: PT=ITU-T G.711 PCMU, SSRC=0x4504251E, Seq=1736, Time=2735700, Mark |
| PT=ITU-T G.711 PCMU | | RTP: PT=ITU-T G.711 PCMU, SSRC=0x1EA5087, Seq=13763, Time=4378440 |
| (5060) (38100) [5060] (38100) | | SIP: Status: 100 Trying |
| Status: 100 Trying | | RTP: PT=ITU-T G.711 PCMU, SSRC=0x4504251E, Seq=1752, Time=2738260 |
| (5060) (38100) [5060] (38100) | | SIP/SDP: Status: 200 OK, with session description |
| PT=ITU-T G.711 PCMU | | SIP: Request: ACK sip:4000@192.168.1.151:38100;instance=45348497a0ea7a62 |
| (5060) (38100) [5060] (38100) | | SIP/SDP: Request: INVITE sip:4000@192.168.1.151:38100;instance=45348497a0ea7a62, with session desc |
| Status: 200 OK, with session description | | RTP: PT=ITU-T G.711 PCMU, SSRC=0x4504251E, Seq=1757, Time=2739060 |
| (5060) (38100) [5060] (38100) | | RTP: PT=ITU-T G.711 PCMU, SSRC=0x1EA5087, Seq=13784, Time=4381800 |
| Request: ACK sip:40 | | SIP/SDP: Status: 200 OK, with session description |
| (5060) (38100) [5060] (38100) | | SIP/SDP: Status: 200 OK, with session description |
| Request: INVITE sip: PT=ITU-T G.711 PCMU | | SIP: Request: ACK sip:4000@192.168.1.151:38100;instance=45348497a0ea7a62 |
| (5060) (38100) [5060] (38100) | | SIP/SDP: Request: INVITE sip:4000@192.168.1.151:38100;instance=45348497a0ea7a62, with session desc |
| Status: 200 OK, with session description | | RTP: PT=ITU-T G.711 PCMU, SSRC=0x4504251E, Seq=1768, Time=2740820 |
| (5060) (38100) [5060] (38100) | | SIP/SDP: Status: 200 OK, with session description |
| Request: ACK sip:40 | | SIP: Request: ACK sip:4000@192.168.1.151:38100;instance=45348497a0ea7a62 |
| (5060) (38100) [5060] (38100) | | SIP/SDP: Request: ACK sip:4000@192.168.1.151:38100;instance=45348497a0ea7a62 |
| PT=ITU-T G.711 PCMU | | RTP: PT=ITU-T G.711 PCMU, SSRC=0x1EA5087, Seq=13789, Time=4382600 |
| (5060) (38100) [5060] (38100) | | SIP/SDP: Request: INVITE sip:4000@192.168.1.151:38100;instance=45348497a0ea7a62, with session desc |
| Request: INVITE sip: PT=ITU-T G.711 PCMU | | RTP: PT=ITU-T G.711 PCMU, SSRC=0x4504251E, Seq=1768, Time=2740820 |
| (5060) (38100) [5060] (38100) | | SIP/SDP: Status: 200 OK, with session description |
| Status: 200 OK, with session description | | SIP: Request: ACK sip:4000@192.168.1.151:38100;instance=45348497a0ea7a62 |
| (5060) (38100) [5060] (38100) | | SIP/SDP: Request: ACK sip:4000@192.168.1.151:38100;instance=45348497a0ea7a62 |
| Request: ACK sip:40 | | RTP: PT=ITU-T G.711 PCMU, SSRC=0x1EA5087, Seq=13800, Time=4384360 |
| (5060) (38100) [5060] (38100) | | RTP: PT=DynamicRTP-Type-126, SSRC=0xC5349F7F, Seq=3962, Time= |
| PT=DynamicRTP-Type- | | |
| (5060) (38100) [5060] (38100) | | |

Estas foram as mensagens trocadas quando fizemos uma chamada entre dois utilizadores. Como podemos ver (na captura do PBX), quando o Utilizador 1 carrega na tecla 9 (zona a vermelho), o PBX inicia o processo de estabelecimento da chamada entre o Utilizador 1 e o 2. Neste instante são tocadas um conjunto de mensagens entre os mesmos, que passam sempre pelo PBX, uma vez que este funciona como um servidor proxy. Inicialmente (após o utilizador 1 carregar a tecla 9), o PBX envia um INVITE ao utilizador 2, indicando a intenção do utilizador de estabelecer a chamada com o mesmo. Depois o utilizador 2 envia uma mensagem

com o código de resposta 180 e um campo textual (Ringing). O utilizador 2 envia um 200 OK indicando o sucesso da ligação, que depois é confirmado pelo PBX (através do envio de um ACK). De seguida, o PBX faz um INVITE para o utilizador 1, de modo a enviar os dados referentes ao utilizador 2 para o utilizador 1. Como podemos ver os pacotes SIP e os pacotes RTP passam por canais diferentes.



O RTP entra em acção nesta fase, para a transmissão de fluxos de dados em tempo real entre os dois utilizadores. É possível verificar que as mensagens RTP têm o Sequence Number (que identifica a ordem dos pacotes) e o Time (que ajuda na sincronização). As mensagens RTCP contêm um conjunto de campos que permitem fazer o controlo de fluxo de dados, como já dissemos, e como também se pode ver nas figuras acima.

Conclusão

Como podemos ver, para o desenvolvimento do projecto utilizamos vários protocolos, de várias camadas. Para a sinalização utilizamos o SIP, para o transporte de fluxo de dados em tempo real foi utilizado o RTP e para o controlo do fluxo de dados foi utilizado o RTCP. O RTP normalmente corre sobre o UDP, por isso é necessário que as aplicações contenham mecanismos de controlo e correcção da ordem de pacotes, por isso temos o RTCP que fornece as informações necessárias a estas aplicações. Ainda temos o SDP dentro do corpo do SIP, que contém informações sobre o utilizador, e os tipos de ficheiros que o mesmo utiliza.

A sintaxe do SIP é semelhante a do HTTP, o que é uma grande vantagem para este protocolo, visto que os programas desenvolvidos para usar o HTTP podem, facilmente, serem adaptados para usar o SIP, mas isso também fará com que as mesmas consumam mais largura de banda.

As mensagens com código 1xx são de informação e as mensagens com código 2xx são de sucesso, tal com já tínhamos visto.

O projecto ajudou-nos bastante a perceber o funcionamento dos protocolos utilizados.