



Voice over IP em Asterisk

Serviço de Redes II

Docente

Luís Eduardo Faria dos Santos

Discentes

Miguel Umbelino da Mota Roma

a2021138955(@isec.pt)

Pedro Miguel Neves Martins

a2021135054(@isec.pt)

Licenciatura em Engenharia Informática

Ramo de Redes e Administração de Sistemas

Instituto Politécnico de Coimbra

Instituto Superior de Engenharia de Coimbra

Coimbra, 25 de janeiro de 2024

Indíce

1	Introdução	1
1.1	Enquadramento	1
1.2	Objetivos	1
1.3	Estrutura do Relatório	2
2	Conceitos	3
2.1	<i>Voice over IP</i>	3
2.2	<i>Session Initiation Protocol</i>	3
2.3	<i>Real-time Transport Protocol</i>	4
2.4	<i>Inter-Asterisk eXchange v2</i>	4
2.5	Private Branch Exchange	4
3	Tecnologias usadas	5
3.1	<i>Asterisk</i>	5
3.2	<i>FreePBX</i>	5
3.3	<i>Zoiper</i>	6
3.4	<i>MicroSIP</i>	6
3.5	<i>Wireshark</i>	6
3.6	<i>WinSCP</i>	6
4	Planeamento	7
4.1	Ambientes laboratoriais	7
4.2	Experiências	8
5	Guião Laboratorial	10
5.1	Instalações necessárias	10
5.2	Chamadas através do protocolo SIP	10
5.2.1	<i>Ubuntu</i>	10
5.2.2	<i>Asterisk</i>	12
5.3	Chamadas internas	13
5.3.1	Configuração do ficheiro sip.conf	13

5.3.2	Configuração do ficheiro extensions.conf	14
5.3.3	<i>Zoiper</i>	15
5.3.4	<i>MicroSIP</i>	16
5.3.5	<i>Zoiper</i>	16
5.3.6	<i>MicroSIP</i>	17
5.3.7	Chamada entre dois utilizadores	18
5.4	Chamadas externas	19
5.5	Gravação de chamadas no <i>Ubuntu</i>	22
5.6	Chamadas através do protocolo IAX2	23
5.6.1	<i>Servidores FreePBX</i>	23
5.6.2	Configurações na interface <i>web</i>	25
5.6.3	Chamada de teste	30
6	Análise do tráfego <i>WireShark</i>	31
6.1	Captura do tráfego	31
6.2	Chamada com o protocolo <i>SIP</i>	31
6.2.1	Inicialização da chamada	32
6.2.2	Transmissão de áudio	33
6.2.3	Encerramento da chamada	34
6.3	Chamada com o protocolo <i>IAX2</i>	34
6.3.1	Inicialização da chamada	35
6.3.2	Transmissão de áudio	36
6.3.3	Encerramento da chamada	36
7	Conclusões	37
7.1	Objetivos alcançados	37
7.2	Opinião acerca da tecnologia estudada	37
	Referências	38

Listas de Figuras

4.1	Ambiente experimental com 1 servidor <i>Asterisk</i>	7
4.2	Ambiente experimental com 2 servidores <i>Asterisk</i>	8
4.3	Ambiente experimental com 2 servidores <i>FreePBX</i>	8
5.1	Escolha do ficheiro <i>International Organization for Standardization</i> (ISO)	10
5.2	Escolha da placa de rede	11
5.3	Tipo de instalação	11
5.4	Configuração do <i>user</i> da <i>Virtual Machine</i> (VM)	12
5.5	Configuração da placa <i>Bridge Adapter</i> no <i>Ubuntu</i>	12
5.6	Instalação dos pacotes do <i>Asterisk</i>	13
5.7	Consola do <i>Asterisk</i>	13
5.8	Configuração do ficheiro sip.conf	14
5.9	Configuração do ficheiro extensions.conf	15
5.10	Utilizadores criados	15
5.11	Instalação do <i>Zoiper</i>	15
5.12	Instalação do <i>MicroSIP</i>	16
5.13	Configuração do <i>Zoiper</i>	16
5.14	Configuração do <i>Zoiper</i>	17
5.15	Configuração do <i>Zoiper</i>	17
5.16	Configuração do <i>MicroSIP</i>	18
5.17	Chamada interna	18
5.18	Chamada interna	19
5.19	Configuração do ficheiro sip.conf	19
5.20	Configuração do ficheiro extensions.conf	20
5.21	Adição do utilizador 1000 ao <i>Zoiper</i>	21
5.22	Adição do utilizador 1001 ao <i>MicroSIP</i>	21
5.23	Chamada efetuada com o protocolo <i>Session Initiation Protocol</i> (SIP) . .	22
5.24	Configuração do ficheiro extensions	22
5.25	Chamada efetuada entre dois utilizadores	23
5.26	Ficheiro wav	23
5.27	Instalação da ferramenta <i>FreePBX</i>	24

5.28	Configuração da <i>password</i> da conta default <i>root</i>	24
5.29	Placa de rede configurada para os servidores <i>FreePBX</i>	24
5.30	Endereço <i>Internet Protocol</i> (IP) para acesso à interface <i>Web</i> no servidor <i>PBX01</i>	25
5.31	Endereço IP para acesso à interface <i>Web</i> no servidor <i>PBX01</i>	25
5.32	Teste de conectividade entre servidores	25
5.33	Configuração da conta de administração da interface <i>Web</i>	26
5.34	Adição de um <i>Inter-Asterisk eXchange v2 (IAX2) Trunk</i> no primeiro servidor	26
5.35	<i>Outgoing IAX2 Trunk</i> no primeiro servidor	26
5.36	<i>Incoming IAX2 Trunk</i> no primeiro servidor	27
5.37	Adição de um <i>IAX2 Trunk</i> no segundo servidor	27
5.38	<i>Outgoing IAX2 Trunk</i> no segundo servidor	27
5.39	<i>Incoming IAX2 Trunk</i> no segundo servidor	28
5.40	Configuração de <i>Outbound Routes</i> no primeiro servidor	28
5.41	<i>Dial patterns</i> que utilizam a rota do servidor <i>PBX01</i>	28
5.42	Configuração de <i>Outbound Routes</i> no segundo servidor	29
5.43	<i>Dial patterns</i> que utilizam a rota do servidor <i>PBX02</i>	29
5.44	Adição do utilizador com a extensão <i>100</i>	29
5.45	Adição do utilizador com a extensão <i>200</i>	30
5.46	Chamada efetuado com o protocolo IAX2	30
6.1	Exemplo de uma chamada usando o SIP	32
6.2	Inicialização da chamada com o protocolo SIP	33
6.3	Captura dos pacotes <i>Real-Time Transport Protocol</i> (RTP) do protocolo SIP	33
6.4	Informações dos pacotes RTP	34
6.5	Gravação do áudio da chamada	34
6.6	Fim da chamada usando o SIP	34
6.7	Exemplo de uma chamada usando o IAX2	35
6.8	Inicialização da chamada com o protocolo IAX2	36
6.9	Captura dos pacotes RTP no protocolo IAX2	36
6.10	Fim da chamada usando o IAX2	36

Acrónimos e Siglas

DHCP *Dynamic Host Configuration Protocol*

GUI *Graphical User Interface*

IAX2 *Inter-Asterisk eXchange v2*

IP *Internet Protocol*

ISO *International Organization for Standardization*

NAT *Network Address Translation*

PBX *Private Branch Exchange*

PSTN *Public Switched Telephone Network*

RTP *Real-Time Transport Protocol*

SIP *Session Initiation Protocol*

TCP *Transmission Control Protocol*

TLS *Transport Layer Security*

UDP *User Datagram Protocol*

VM *Virtual Machine*

VoIP *Voice over IP*

SFTP *SSH File Transfer Protocol*

FTP *File Transfer Protocol*

Capítulo 1

Introdução

Este relatório serve como base de documentação de todo o trabalho desenvolvido ao longo da realização do projeto B no âmbito da unidade curricular Serviço de Redes II da licenciatura em Engenharia Informática - Ramo de Redes e Administração de Sistemas do Instituto Superior de Engenharia de Coimbra no ano letivo de 2023/2024.

1.1 Enquadramento

Semelhante à introdução do telefone, que transformou radicalmente a comunicação humana, o *Voice over IP* (VoIP) redefiniu não apenas a forma como as empresas conduzem os negócios, mas também como as pessoas se comunicam diariamente.

Num cenário tecnológico em rápido avanço, uma forma de comunicação rápida e eficiente tornava-se essencial. O VoIP emergiu como uma tecnologia fundamental, ao permitir a transmissão de voz e multimídia pela Internet, substituindo as linhas telefónicas tradicionais, que eram bastante limitadas, quer por limitações geográficas ou devido aos altos custos.

Este serviço evoluiu de tal maneira que, atualmente, em todos os escritórios, a comunicação por via VoIP é de máxima importância, não apenas internamente, mas também externamente ao lidar com clientes, potenciais parceiros, investidores, entre outros. Fora do mundo dos negócios, no quotidiano, muitas pessoas utilizam esta tecnologia para comunicar de forma acessível ou até gratuitamente com familiares que vivem no estrangeiro ou mesmo apenas noutra parte do país.

1.2 Objetivos

O tópico central deste projeto passará pelo estudo e compreensão do funcionamento das comunicações por voz recorrendo ao VoIP e tecnologias a esta associada. Ao aprofundar nos detalhes do VoIP, pretendemos não apenas adquirir conhecimento teórico, mas

também obter experiência prática através da realização de experiências laboratoriais.

1.3 Estrutura do Relatório

O presente relatório encontra-se organizado nos seguintes sete capítulos:

- *Capítulo 1* - Neste capítulo é feita uma introdução à envolvente geral do projeto e o enquadramento histórico do mesmo. São ainda apresentados os objetivos a cumprir durante o desenvolvimento do trabalho.
- *Capítulo 2* - Neste capítulo são apresentados conceitos essenciais para uma melhor compreensão do trabalho desenvolvido, essenciais para que seja possível entender alguns conteúdos abordados na parte experimental do projeto.
- *Capítulo 3* - Neste capítulo são apresentadas as tecnologias utilizadas para a realização da parte experimental do projeto.
- *Capítulo 4* - Neste capítulo é feita uma exposição das topologias desenvolvidas para efeitos de teste das ferramentas que se pretendem estudar. São ainda indicadas as experiências a serem alvo de teste em termos práticos.
- *Capítulo 5* - Neste capítulo encontra-se o guião laboratorial elaborado com as etapas seguidas, juntamente com auxiliares visuais, para a realização com sucesso dos testes propostos.
- *Capítulo 6* - Neste capítulo encontra-se o estudo feito relativamente aos pacotes capturados na realização das comunicações VoIP, onde podem ser observados os protocolos com o auxílio da ferramenta *Wireshark*.
- *Capítulo 7* - Neste capítulo são descritos os objetivos alcançados e é feita uma retrospectiva acerca do trabalho desenvolvido.

Para além destes capítulos, existem ainda as *Referências* que se encontram no final do documento e que servem de complemento ao relatório.

Capítulo 2

Conceitos

Neste capítulo serão apresentados alguns conceitos e terminologias essenciais para a compreensão do projeto desenvolvido.

2.1 *Voice over IP*

VoIP [9] é uma tecnologia que permite a comunicação em tempo real pela Internet ou por redes de protocolo IP, e que utiliza protocolos de rede, como o SIP e o RTP abordados posteriormente, para estabelecer chamadas e para garantir a qualidade da transmissão de áudio.

A tecnologia VoIP [4] converte sinais de áudio em dados digitais, que são transmitidos pela Internet. Os dados de voz são comprimidos antes da transmissão e descomprimidos no destino, usando *codecs* (algoritmos que comprimem e descomprimem a voz), permitindo a comunicação de voz por meio de uma conexão de Internet.

2.2 *Session Initiation Protocol*

O SIP [8] é um protocolo de controlo da *Application Layer* que suporta VoIP, especificando as mensagens enviadas entre terminais e lidando com os aspetos reais de uma conversa. Pode ser utilizado para criar, modificar e terminar sessões com um ou mais participantes. É frequentemente utilizado em sistemas de comunicação por VoIP em tempo real pela Internet. Os seus componentes principais incluem utilizadores e servidores SIP, responsáveis por registar utilizadores no sistema e coordenar a comunicação entre os mesmos.

Os clientes SIP geralmente utilizam *Transmission Control Protocol* (TCP) ou *User Datagram Protocol* (UDP) nas portas 5060 ou 5061 para se conectar a servidores SIP. A porta 5060 é usada para tráfego de sinalização não criptografada, enquanto a porta 5061 é tipicamente usada para tráfego criptografado com o *Transport Layer Security* (TLS).

2.3 *Real-time Transport Protocol*

Na sequência do protocolo anteriormente apresentado, surje o protocolo de transporte RTP [3] que atua na camada de aplicação sendo usado para transmitir dados em tempo real, como áudio e vídeo em fluxo contínuo através da Internet. Para transmissão de tal informação, é utilizado o protocolo UDP, porque permite uma entrega mais rápida dos dados.

Devido a este ser um protocolo assíncrono, não garante que os pacotes sejam entregues na ordem correta ou no tempo especificado, no entanto, ele fornece informações de tempo para permitir que os receptores reconstruam os dados corretamente.

2.4 *Inter-Asterisk eXchange v2*

É um protocolo de comunicação [5] de VoIP desenvolvido especificamente para o sistema de telefonia *Asterisk* que faz uso de um único porto UDP (4569) tanto para tráfego de controlo, como também para tráfego de dados, o que pode ser vantajoso em cenários de *Firewall* e ou *Network Address Translation* (NAT).

O tráfego de voz é transmitido *in-band*, oferecendo assim excelente desempenho e eficiência de largura de banda, tornando-o uma escolha popular para interligar servidores *Asterisk* ou conectar telefones SIP a sistemas *Asterisk*.

O IAX2 suporta *trunking* num único link, o que permite que dados de múltiplas chamadas sejam unidos num único conjunto de pacotes, o que significa que pode entregar informações para mais de uma chamada, reduzindo o gasto efetivo IP sem criar latência adicional.

2.5 *Private Branch Exchange*

O *Private Branch Exchange* (PBX) [7] é um sistema que atua como central telefónica numa organização fazendo reencaminhamento de chamadas entre os diferentes utilizadores em linhas locais, permitindo que todos os utilizadores partilhem um determinado número de linhas telefónicas externas.

Um servidor PBX permite também fazer ligações à rede telefónica pública, *Public Switched Telephone Network* (PSTN), possuindo uma ou várias linhas, que podem ser partilhadas pela organização poupando o custo de uma linha para cada utilizador até à central telefónica da empresa., sendo a sua gestão feita de forma transparente para os utilizadores.

Capítulo 3

Tecnologias usadas

Para ser possível realizar o conjunto de experiências que se sucedem nos seguintes capítulos foi necessária uma instalação prévia, de um conjunto de serviços e aplicações, com ferramentas a seu dispor que permitam explorar a implementação do serviço de VoIP e dos seus protocolos de comunicação.

3.1 *Asterisk*

Asterisk é uma plataforma usada para implementar a tecnologia VoIP. Funciona como um servidor de comunicações que permite a realização de chamadas telefónicas sobre redes IP.

O *Asterisk* oferece uma ampla gama de recursos avançados, incluindo chamadas em conferência, atendimento automático, encaminhamento de chamadas, etc. É usado também para implementar soluções de comunicações únicas, integrando voz, vídeo, mensagens e outras formas de comunicação numa única plataforma.

Além disso, a plataforma suporta diversos tipos de *hardware*, como placas de telefonia, adaptadores de VoIP, telefones IP e *gateways* de VoIP, o que torna mais fácil integrar os sistemas usados com a infraestrutura de VoIP.

3.2 *FreePBX*

Esta distribuição do *Asterisk* é uma solução popular para quando há a necessidade de implementar um sistema de telefonia escalável e econômico. O *FreePBX* [1] é uma *Graphical User Interface* (GUI) de código aberto baseada na *Web* que controla e gere o *Asterisk PBX*, um servidor de comunicações de código aberto.

Através da interface *Web* é possível gerir facilmente todas as funções de um sistema de VoIP, desde a criação e gestão de extensões de telefone até configuração de *SIP Trunks* que permitem o sistema VoIP se conectar à rede, entre outras.

3.3 *Zoiper*

O *Zoiper* [2] é um software de *softphone* VoIP gratuito que permite fazer chamadas telefónicas e videochamadas pela internet. É compatível com uma ampla gama de provedores de VoIP e PBX, pois suporta os principais protocolos que estes utilizam, entre os quais SIP e IAX2 abordados no capítulo seguinte.

3.4 *MicroSIP*

O *MicroSIP* [6] é projetado para ser uma solução simples e eficiente para comunicações VoIP. É utilizado para realizar chamadas de voz pela Internet, permitindo que os utilizadores se comuniquem entre si por meio de computadores ou outros dispositivos compatíveis com o SIP. É um recurso idêntico ao *Zoiper* sendo a sua utilização uma forma de ultrapassar a limitação imposta pela versão gratuita do *Zoiper* que apenas permite adicionar um utilizador.

3.5 *Wireshark*

O *Wireshark* [11] é uma ferramenta analisadora de pacotes de rede que apresenta os dados dos pacotes capturados e os organiza por protocolos com o máximo de detalhes possível. No âmbito do presente projeto foi utilizado para possibilitar uma compreensão mais aprofundado do que pressupõe, em termos de protocolos e tráfego, uma chamada entre dois utilizadores através do servidor *Asterisk* e do servidor *FreePBX*.

3.6 *WinSCP*

O *WinSCP* [10] é um cliente *SSH File Transfer Protocol* (SFTP), cliente *File Transfer Protocol* (FTP), de gestor de ficheiros de código aberto para *Windows*. A sua principal função é a transferência de ficheiros entre um computador local e um computador remoto. Para além disso, o *WinSCP* oferece funcionalidades de scripting e de gestor de ficheiros básico.

No contexto do presente projeto, a sua utilização passou pela extração dos ficheiros de captura *Wireshark* do servidor *FreePBX*, visto que este não tem uma GUI com a qual é possível interagir para arrastar ou copiar ficheiros para a máquina local.

Capítulo 4

Planeamento

Com este guia, pretende-se a simulação de alguns cenários que possam ocorrer num contexto real. Quer sejam estes situações de falhas e/ou avaria ou até situações de uso normal no dia-a-dia de modo a observar o comportamento do serviço de VoIP.

No fim de cada experiência serão apresentados os resultados (com auxiliares gráficos) e a conclusão retirada no final de cada teste. Abaixo estão algumas das experiências a desenvolver e uma breve descrição das mesmas

4.1 Ambientes laboratoriais

De maneira a possibilitar a experimentação das múltiplas funcionalidades oferecidas pelo *Asterisk* foram desenvolvidos três ambientes, onde cada um foi desenvolvido com o propósito da realização de uma determinada experiência.

A topologia da Figura 4.1, foi desenvolvida com o objetivo de testar uma simples ligação VoIP entre dois clientes que se encontram no mesmo servidor e portanto na mesma rede.

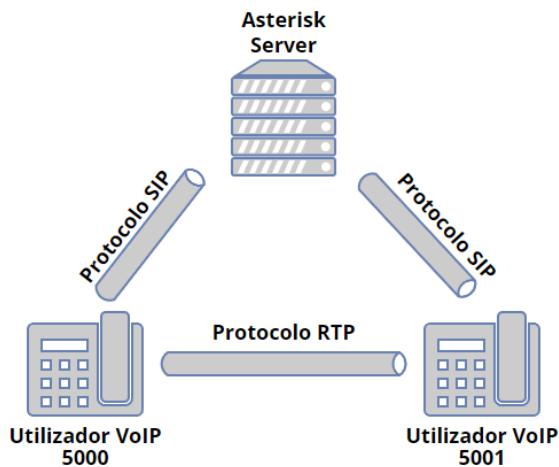


Figura 4.1: Ambiente experimental com 1 servidor *Asterisk*

A topologia da Figura 4.2, foi desenvolvida com o objetivo de testar o método de ligação entre vários servidores *Asterisk*, através da implementação do *SIP Trunking*, e como as chamadas eram encaminhadas entre diferentes redes.

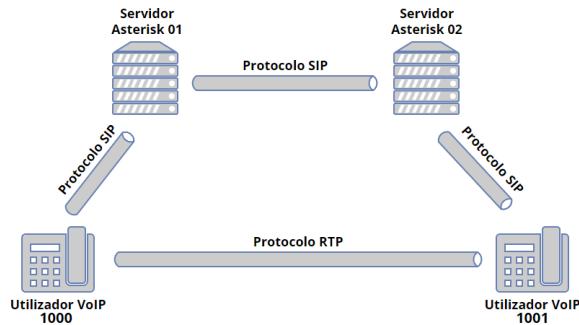


Figura 4.2: Ambiente experimental com 2 servidores *Asterisk*

A topologia da Figura 4.3, foi desenvolvida com o objetivo de testar o método de ligação entre vários servidores *FreePBX*, através da implementação do *IAX2 Trunking*, e como as chamadas eram encaminhadas entre diferentes redes.

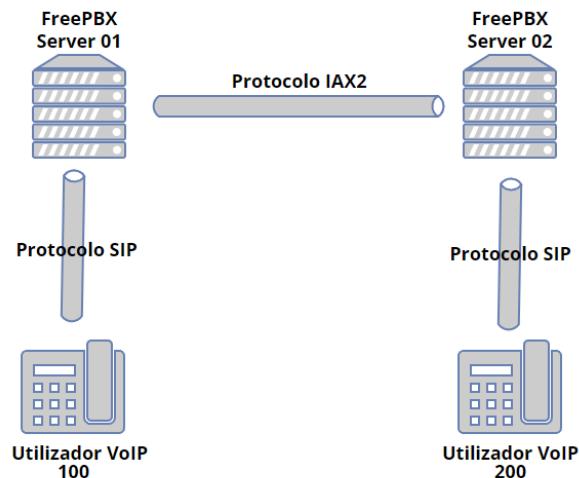


Figura 4.3: Ambiente experimental com 2 servidores *FreePBX*

4.2 Experiências

Alguns dos testes que se pretendem realizar, nos servidores criados, para compreensão do funcionamento do VoIP e quais as suas limitações, bem como os seus pontos fortes:

- **Chamadas internas:** O objetivo é testar a funcionalidade de chamadas internas no *Asterisk* e no *Zoiper*, onde os dois dispositivos se comunicam dentro da mesma rede. No *Asterisk*, são configurados dois utilizadores SIP e no *Zoiper* criados dois

perfis (um para cada utilizador) e conectados ao *Asterisk*. Através do *Zoiper*, é efetuada uma chamada de um perfil para outro dentro da rede.

- **Encaminhamento de chamadas:** Configuração de regras de encaminhamento de chamadas no *Asterisk* e através do *Zoiper* fazer chamadas que correspondam às regras de encaminhamento configuradas.
- **Gravação de chamadas:** São definidas as configurações de gravação, como a diretoria onde as gravações serão armazenadas e as condições para iniciar a gravação. Utilizando o *Zoiper* e o *MicroSIP*, mais uma vez, para fazer chamadas e verificar no *Asterisk* se elas realmente estão a ser gravadas na diretoria correta.

Capítulo 5

Guião Laboratorial

Neste capítulo será documentado todo o processo de configuração do ambiente sob o qual iremos trabalhar, bem como todas as experiências realizadas ao longo do projeto e as diversas conclusões a retirar sobre cada uma.

5.1 Instalações necessárias

Para ser possível realizar o conjunto de experiências que se sucedem nas seguintes secções foi necessária uma instalação prévia, de um conjunto de serviços e aplicações, com ferramentas a seu dispor que permitam explorar as chamadas VoIP e os seus protocolos.

5.2 Chamadas através do protocolo SIP

5.2.1 *Ubuntu*

Primeiramente, foi feita a instalação de um ambiente *Ubuntu* no *VirtualBox*. A versão escolhida foi a 22.04.3, como pode ser visto na Figura 5.1.

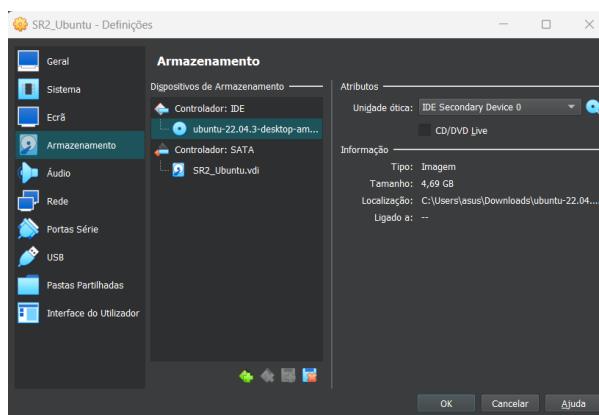


Figura 5.1: Escolha do ficheiro ISO

De seguida, uma placa de rede foi configurada como *Bridge Adapter* (Figura 5.2) de modo a ter conexão com outra máquina virtual, que vai ser necessário posteriormente e com o próprio sistema hospedeiro.

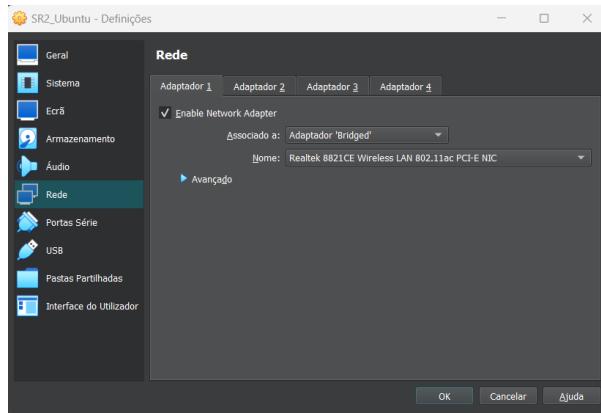


Figura 5.2: Escolha da placa de rede

Depois de criada a VM, foi feita a configuração do ambiente *Ubuntu* sob o qual vão ser realizadas as configurações a trabalhar. O tipo de instalação escolhido foi *minimal installation*, como pode ser visto na Figura 5.3, que não é mais do que a instalação do sistema operativo sem pacotes adicionais, contendo apenas os componentes essenciais necessários para o funcionamento básico do sistema.

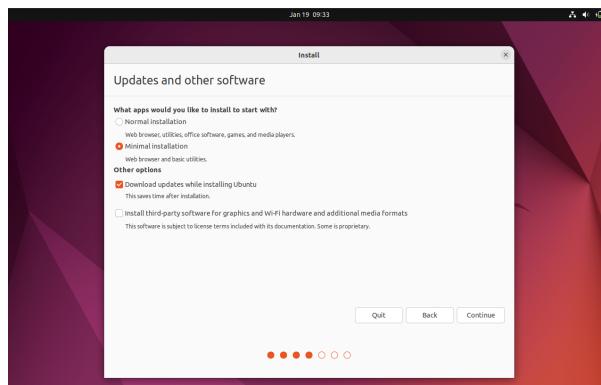


Figura 5.3: Tipo de instalação

Para concluir a instalação foi pedida a criação de um utilizador que possa ser utilizado para fazer *login* no ambiente e com que se possa proceder às alterações necessárias a serem feitas, como é possível observar na Figura 5.4.

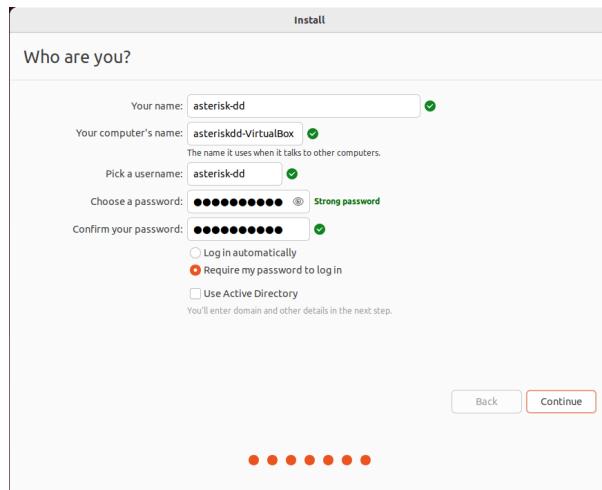


Figura 5.4: Configuração do *user* da VM

Depois de configurado o ambiente *Ubuntu*, foi adicionado um IP estático à VM, como podemos ver na Figura 5.5. Esta configuração foi implementada como medida de precaução para que não haja problemas com a mudança inesperada de endereço IP, caso este estivesse a ser atribuído por *Dynamic Host Configuration Protocol* (DHCP), visto que poderia causar problemas na implementação do serviço de VoIP.

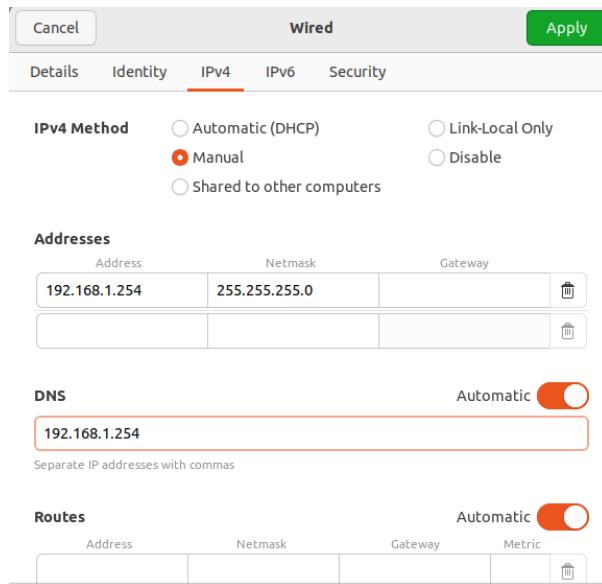


Figura 5.5: Configuração da placa *Bridge Adapter* no Ubuntu

5.2.2 Asterisk

A instalação da tecnologia *Asterisk* e todos os pacotes que lhe estão associados, no ambiente *Ubuntu*, apenas foi preciso colocar o comando ilustrado na Figura 5.6.

Depois de instalada a ferramenta, temos acesso a uma consola, que pode ser vista na Figura 5.7, onde podemos interagir com o sistema, visualizar determinadas informações

Figura 5.6: Instalação dos pacotes do *Asterisk*

de utilizadores e de troncos e realizar outras tarefas relacionadas ao *Asterisk*.

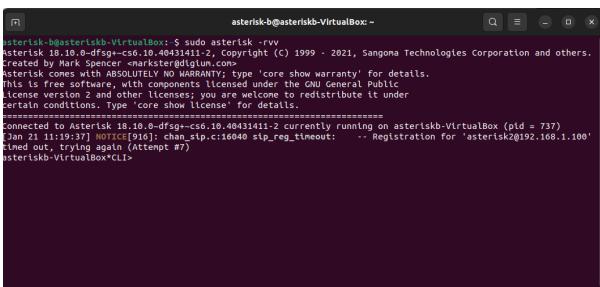


Figura 5.7: Consola do Asterisk

5.3 Chamadas internas

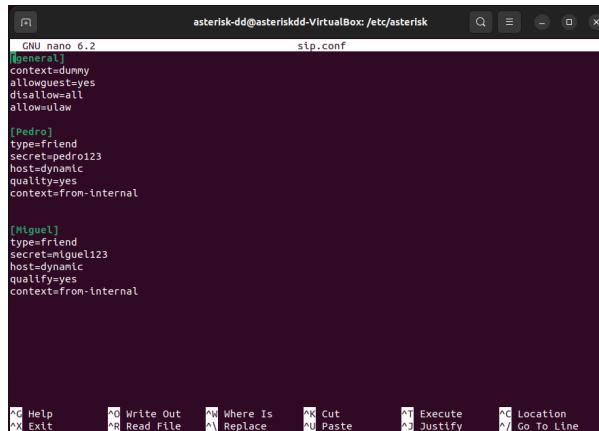
O *Asterisk* permite a criação de uma rede de telefonia interna, onde os utilizadores podem ligar de uns para os outros, usando números de extensão designados. Esta experiência envolve a criação de 2 utilizadores num servidor *Asterisk*, bem como a configuração de contextos que determinam o que acontece numa chamada telefónica.

5.3.1 Configuração do ficheiro sip.conf

O ficheiro **sip.conf** é específico para o protocolo SIP, que é amplamente utilizado para estabelecer, modificar e encerrar sessões de comunicação, como chamadas de voz e vídeo, pela Internet. Contém as configurações relacionadas aos dispositivos SIP que se conectam posteriormente ao servidor *Asterisk*. Neste ficheiro, são especificados os parâmetros necessários para configurar telefones IP, *softphones*, *gateways* e outros dispositivos que usam o protocolo SIP para comunicarem com o servidor *Asterisk*.

Para esta experiência, foi configurado o ficheiro **sip** para aceitar chamadas de qualquer utilizador sem fornecer as credenciais de autenticação. Além disso, o comando **disallow=all** desabilita todos os *codecs* inicialmente e o **allow=ulaw** ativa apenas o *codec*

ulaw para comunicação de áudio. Na criação dos utilizadores, criamos uma senha e definimos o seu contexto específico que vai ser configurado no passo seguinte. Todo este processo é possível ser visto na Figura 5.8.



```

GNU nano 6.2
[sip.conf]
[general]
context=dummy
allowguest=yes
disallow=all
allowulaw

[Pedro]
type=friend
secret=pedro123
host=dynamic
quality=yes
context=from-internal

[Miguel]
type=friend
secret=miguel123
host=dynamic
quality=yes
context=from-internal

```

Figura 5.8: Configuração do ficheiro **sip.conf**

5.3.2 Configuração do ficheiro **extensions.conf**

O ficheiro **extensions.conf** organiza as configurações em contextos. Cada contexto contém uma série de regras e ações que especificam como as chamadas devem ser manipuladas dentro desse contexto específico. Dentro de um contexto, é comum definir regras que especificam como o *Asterisk* deve interpretar os números atribuídos aos utilizadores. As regras geralmente incluem padrões para combinar números e ações associadas a esses padrões. Nesta experiência, foi feita uma configuração simples e prática deste ficheiro, como mostra a Figura 5.9, que se provou suficiente para o que havia sido planeado como experiência.

No contexto *from-internal*, os utilizadores podem iniciar uma chamada telefónica com os números 5000 ou 5001 (Pedro e Miguel, respetivamente) para alcançar os utilizadores SIP. Essas regras direcionam as chamadas para o servidor, que tenta estabelecer a chamada por até 10 segundos.

```
GNU nano 6.2                               extensions.conf
[from-internal]
exten=>5000,1,dial(SIP/Pedro,10)
exten=>5001,1,dial(SIP/Miguel,10)

[ Wrote 3 lines ]
^X Help      ^Q Write Out   ^M Where Is   ^K Cut       ^T Execute   ^C Location
^R Read File ^A Replace    ^U Paste     ^J Justify   ^G Go To Line
```

Figura 5.9: Configuração do ficheiro `extensions.conf`

Como podemos ver na Figura 5.10, os dois utilizadores foram criados com sucesso no ambiente *Asterisk*.

```
asterisk-dd@asteriskdd-VirtualBox:~$ sudo asterisk -rvv
Asterisk 18.10.0-0-dfsg+-cs6.10.40431411-2, Copyright (C) 1999 - 2021, Sangoma Technologies Corporation and others
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for details.
This is free software, with components licensed under the GNU General Public
License version 2 and other licenses; you are welcome to redistribute it under
certain conditions. Type 'core show license' for details.
=====
Connected to Asterisk 18.10.0-0-dfsg+-cs6.10.40431411-2 currently running on asteriskdd-VirtualBox
(pto=2488)
asteriskdd-VirtualBox*CLI> sip show users
Username          Secret          Accountcode  Def.Context   ACL  Forceroport
Miguel           migue123        from-internal  No   No
Pedro            pedro123        from-internal  No   No
asteriskdd-VirtualBox*CLI>
```

Figura 5.10: Utilizadores criados

5.3.3 Zoiper

Após a instalação do *Asterisk* no *Ubuntu*, foi instalado o *Zoiper* que como, referido anteriormente, é uma ferramenta que permite efetuar e receber chamadas telefónicas através da *Internet*. A Figura 5.11 mostra a instalação da ferramenta.

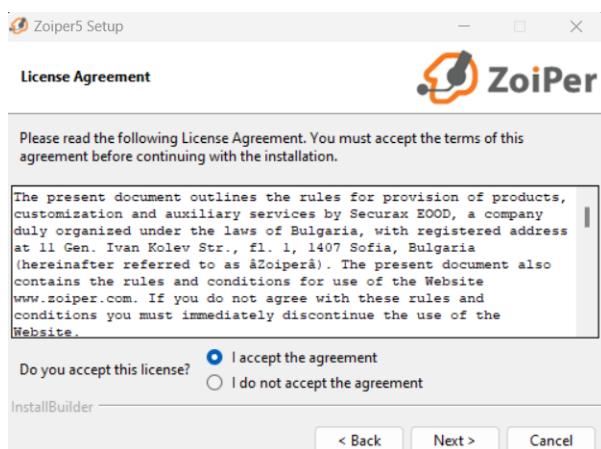


Figura 5.11: Instalação do *Zoiper*

5.3.4 MicroSIP

O *MicroSIP* é uma ferramenta semelhante ao *Zoiper*, também bastante usada com a mesma finalidade. Podemos ver a sua instalação na Figura 5.12.

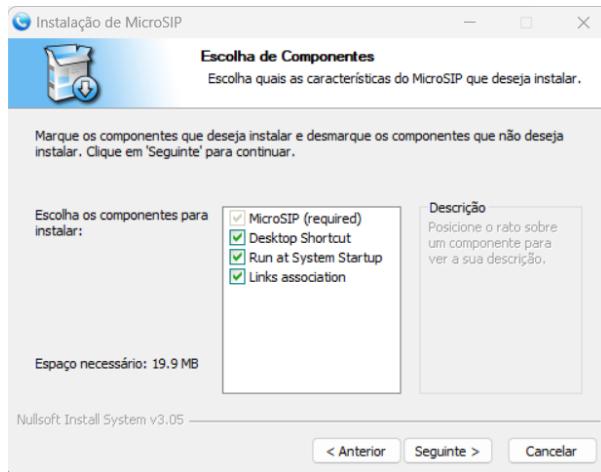


Figura 5.12: Instalação do *MicroSIP*

5.3.5 Zoiper

Depois das configurações acima, foi adicionado ao *Zoiper* um dos utilizadores criados acima (nome-utilizador@ip-servidor-asterisk e a senha), como mostra a Figura 5.13.

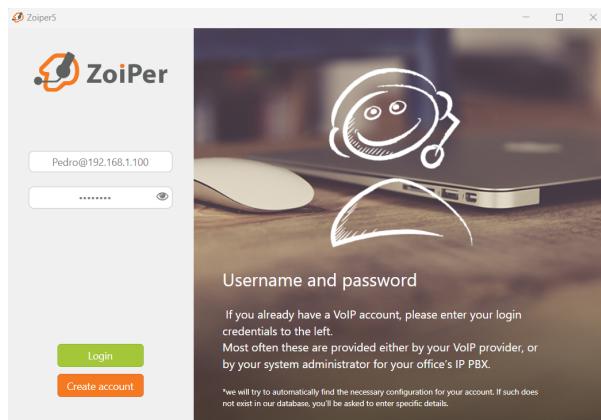


Figura 5.13: Configuração do *Zoiper*

De seguida, foi selecionado o *provider*, que é o IP do servidor *Asterisk*, como pode ser visto na Figura 5.14.

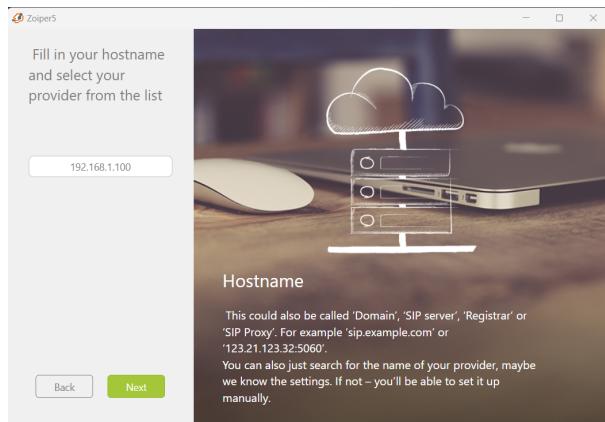


Figura 5.14: Configuração do *Zoiper*

E por fim, é feito um *discover* à procura da porta em questão que é a UDP. A porta padrão para o SIP sobre UDP é a porta 5060. Este processo de descoberta está explícito na Figura 5.15.

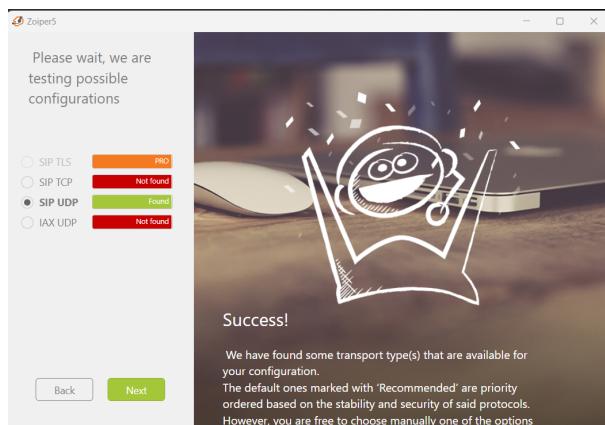


Figura 5.15: Configuração do *Zoiper*

5.3.6 *MicroSIP*

No *MicroSIP*, foram colocadas as credenciais do utilizador "Miguel", tal como o IP do servidor *Asterisk* onde o mesmo está alojado, como podemos ver na Figura 5.16.

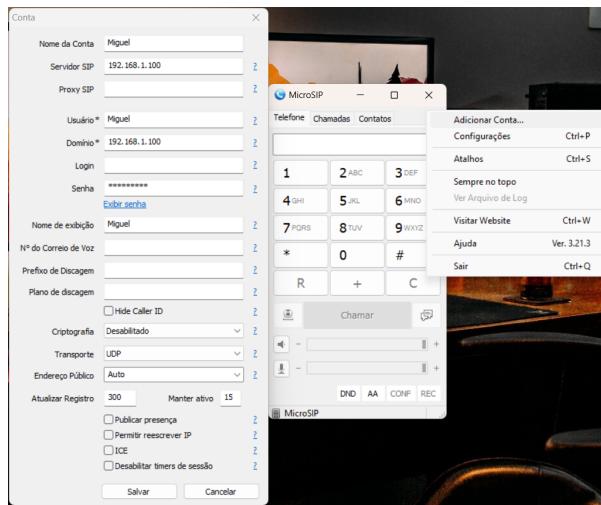


Figura 5.16: Configuração do *MicroSIP*

5.3.7 Chamada entre dois utilizadores

Nas seguintes figuras (5.17 e 5.18), podemos ver as chamadas a serem efetuadas corretamente entre os utilizadores criados.

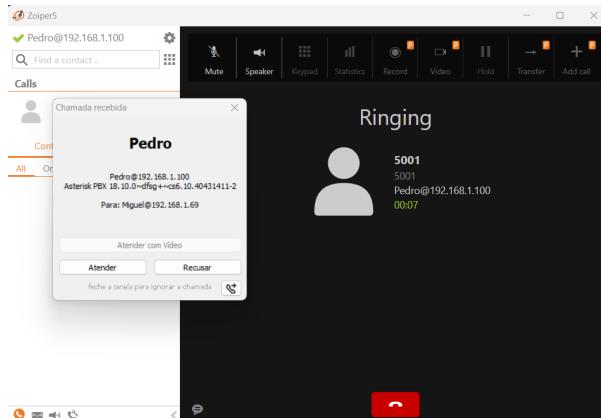


Figura 5.17: Chamada interna

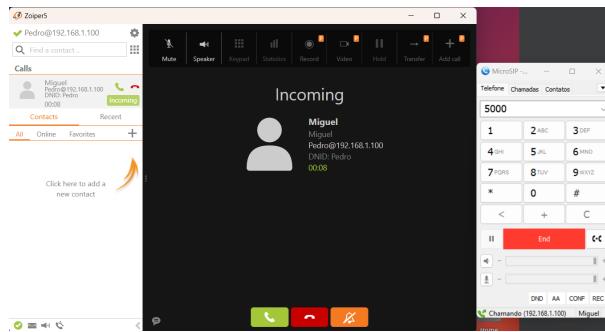


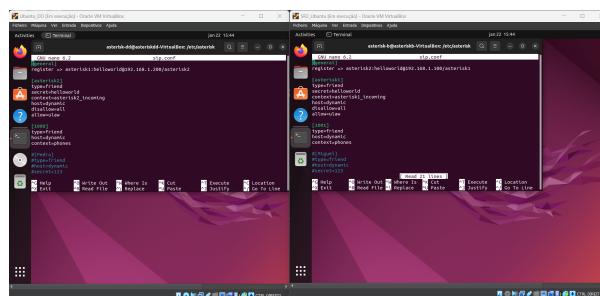
Figura 5.18: Chamada interna

5.4 Chamadas externas

Esta nova experiência tem como objetivo a realização de uma chamada entre dois utilizadores alojados cada um num servidor diferente. Para isso, foi criado uma nova VM com o ambiente *Ubuntu* onde foi instalado o *Asterisk*, seguindo os passos já demonstrados acima.

Para realizar esta experiência, foi alterado, em primeiro lugar, o ficheiro **sip.conf** visto que tem de ser registado ambos os servidores um no outro, respetivamente, como mostra a linha **register => asterisk1:helloworld@192.168.1.200/asterisk2**, de modo a conseguirem trocar informações acerca das chamadas efetuadas e recebidas entre eles.

Nessa mesma linha, **asterisk1** e **helloworld** são o nome de utilizador (que vai ser registado no outro servidor) e a palavra passe. Os elementos seguintes representam o endereço IP do servidor remoto e o nome de utilizador criado no mesmo. De seguida, foram criados dois utilizadores, que são colocados em contextos diferentes, como vamos ver se seguida. A configuração dos ficheiros **sip** de ambos os servidores está ilustrada na Figura 5.19.

Figura 5.19: Configuração do ficheiro **sip.conf**

Posteriormente, no ficheiro **extensions.conf** foram adicionados os dois contextos: **internal** e **remote**. Um dos contextos tem como objetivo realizar chamadas entre servidores (**remote**) e o outro realizar chamadas internas (**internal**). No contexto **phones**, foram adicionados os dois contextos configurados, ou seja, é concedida a permissão aos utilizadores **1000** e **1001**, criados anteriormente, de fazer chamadas internas e externas. Toda esta configuração é possível ver na Figura 5.20.

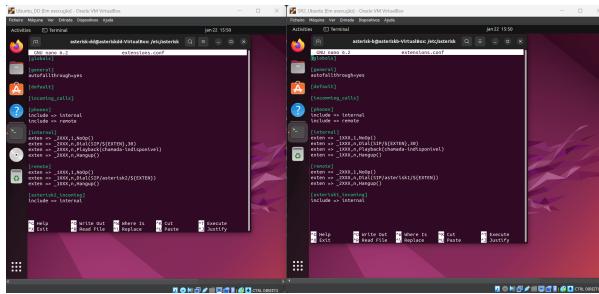


Figura 5.20: Configuração do ficheiro **extensions.conf**

No contexto **internal**, foram definidas 4 regras:

- **exten => 1XXX,1,NoOp()**. Este comando indica uma *extension label* para números que coincidam com o padrão **1XXX**. O comando **NoOp()** corresponde a uma operação nula que pode ser usada para fins de *debug* ou documentação.
- **exten => 1XXX,n,Dial(SIP/EXTEN,30)**. Aqui, o comando **Dial** é utilizado para estabelecer uma chamada. O parâmetro **SIP/EXTEN** indica que a chamada será encaminhada para um dispositivo **SIP** correspondente ao número discado. Já o parâmetro **30** indica que a chamada durará no máximo 30 segundos.
- **exten => 1XXX,n,Playback(chamada-indisponivel)**. Para o caso da chamada não ser atendida por algum motivo, este comando reproduzirá uma mensagem de áudio indicando que a parte chamada está indisponível no momento.
- **exten => 1XXX,n,Hangup()**. Este comando é responsável por encerrar a chamada, seja ela atendida ou não.

No contexto **remote**, foram definidas 3 regras:

- **exten => 2XXX,1,NoOp()**. Da mesma forma que no contexto anterior, este é uma *extension label* para números que coincidam com o padrão **2XXX**.
- **exten => 2XXX,n,Dial(SIP/asterisk2/EXTEN)**. Neste comando, o destino da chamada é um dispositivo indicado pelo prefixo **asterisk2/** (remetendo para o segundo servidor *Asterisk* com o endereço IP 192.168.1.200). O **EXTEN** é utilizado para substituir a variável pela extensão discada , neste caso a **2XXX**.

- **exten => 2XXX,n,Hangup()**. Tal como visto anteriormente, este comando encerra a chamada.

De seguida, foram adicionados os utilizadores VoIP. No *Zoiper* foi definido o utilizador com a extensão **1000**, como pode ser visto na Figura 5.21

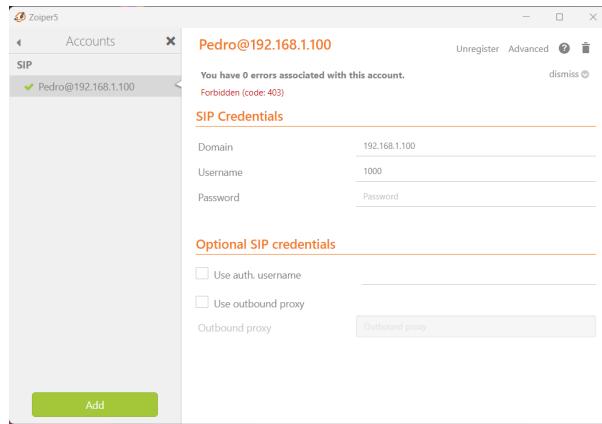


Figura 5.21: Adição do utilizador **1000** ao *Zoiper*

O utilizador com a extensão **1001** ficou atribuído ao *MicroSIP*, como se encontra ilustrado na Figura 5.22.

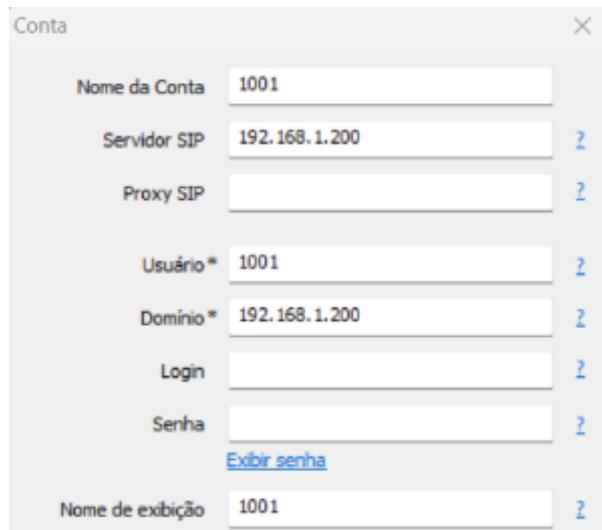


Figura 5.22: Adição do utilizador **1001** ao *MicroSIP*

Na Figura 5.23, é possível ver que foi estabelecida com sucesso a chamada entre os dois utilizadores, alojados em servidores *Asterisk* distintos.

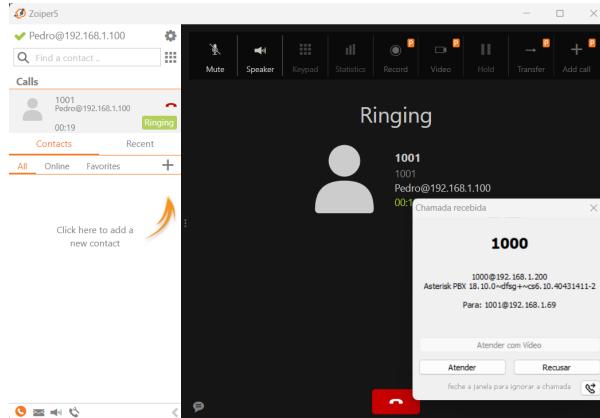


Figura 5.23: Chamada efetuada com o protocolo SIP

5.5 Gravação de chamadas no *Ubuntu*

Esta experiência tem como objetivo gravar o audio de uma chamada num ficheiro **.wav**. Este formato de ficheiro é muito utilizado para armazenar áudio em computadores, sendo um dos formatos de ficheiros de áudio sem perda de dados mais populares.

Como é possível ver na Figura 5.24, foi adicionado ao ficheiro **extensions** a linha "**exten => 2XXX,n,MixMonitor(RECORDING-PATH/EXTEN.wav)"**" nos contextos *remote* e *internal*. A função "**MixMonitor**" é usada para gravar chamadas e os parâmetros seguintes definem o caminho e o nome da gravação. O **RECORDING-PATH** é uma variável que está definida com o caminho, no qual irá ser guardada a captura de áudio. Já a variável **EXTEN**, remete para o número discado, para o qual é feito a chamada.

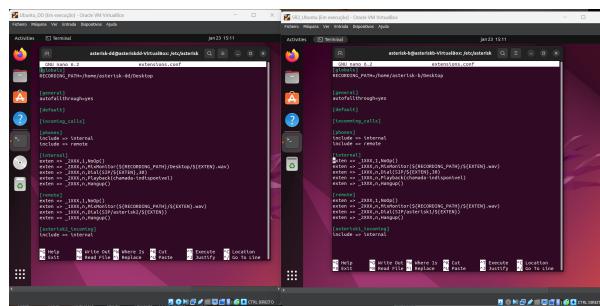


Figura 5.24: Configuração do ficheiro **extensions**

De seguida, é efetuada a chamada entre dois utilizadores, como mostra a Figura 5.25.

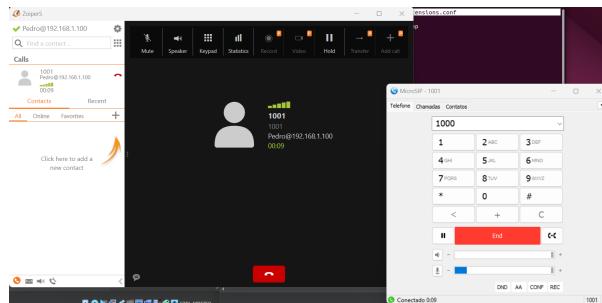


Figura 5.25: Chamada efetuada entre dois utilizadores

Depois de terminada a chamada, o ficheiro é guardado com sucesso na diretoria desejada, como podemos ver na Figura 5.26.

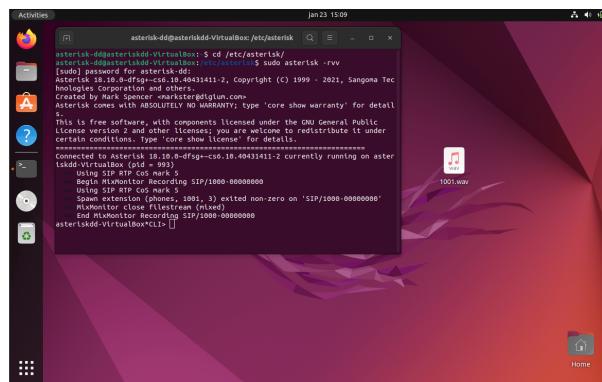


Figura 5.26: Ficheiro wav

5.6 Chamadas através do protocolo IAX2

De modo a poder ter contacto com ambos os protocolos oferecidos pelo *Asterisk* foi também realizada uma chamada com o protocolo IAX2 com recurso à distribuição *FreePBX*.

5.6.1 Servidores FreePBX

Primeiro foi necessário proceder à instalação dos servidores *FreePBX*, num ambiente virtualizador para que não seja necessário a modificação do sistema hospedeiro, sendo o processo feito de forma idêntica para ambos os servidores, como ilustra a Figura 5.27.

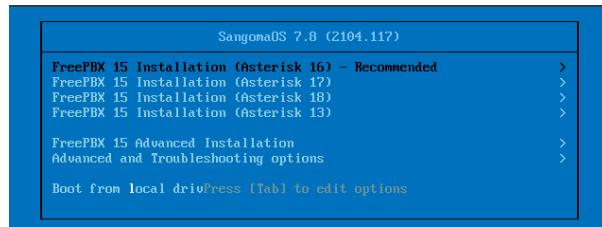


Figura 5.27: Instalação da ferramenta *FreePBX*

Ainda no processo de instalação, para que este possa ser concluído com sucesso, é obrigatório a definição de uma *password* para a conta default, neste caso a conta *root*, para que se possa administrar o servidor com as permissões máximas, como mostra a Figura 5.28.

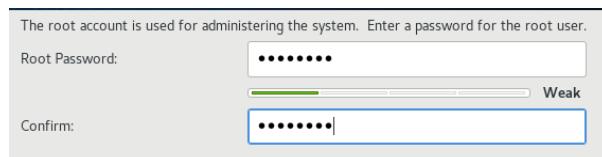


Figura 5.28: Configuração da *password* da conta default *root*

Visto que o *FreePBX* possui uma GUI *Web*, seria necessário que estes servidores obtem-se um endereço IP partilhado do adaptador de rede da máquina anfitriã, nesse sentido foi configurada a placa de rede de ambos os servidores como ilustrado na Figura 5.29.

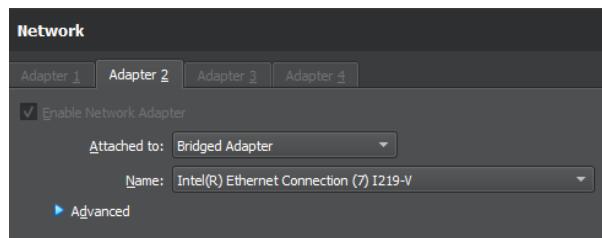


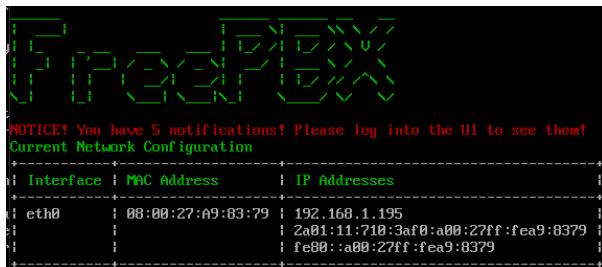
Figura 5.29: Placa de rede configurada para os servidores *FreePBX*

Após estar concluído o processo de instalação e configuração, o servidor obteve um endereço IP como mostra a Figura 5.30, na gama da rede do sistema hospedeiro, cuja utilização passa pelo acesso à interface *Web* do mesmo.

Current Network Configuration		
Interface	MAC Address	IP Addresses
eth0	00:00:27:D0:85:6C	192.168.1.193 2a01:11:710:3af0:a00:27ff:fe7d:856c fe80::a00:27ff:fe7d:856c

Figura 5.30: Endereço IP para acesso à interface Web no servidor *PBX01*

O mesmo acontece para o segundo servidor como é possível ver na Figura 5.31, que fica associado ao endereço IP 192.168.1.195.



Current Network Configuration		
Interface	MAC Address	IP Addresses
eth0	00:00:27:A9:83:79	192.168.1.195 2a01:11:710:3af0:a00:27ff:fea9:8379 fe80::a00:27ff:fea9:8379

Figura 5.31: Endereço IP para acesso à interface Web no servidor *PBX01*

De modo a testar a conectividade entre ambos os servidores, para assegurar que não existe nenhuma *firewall* a negar o acesso, foi feito um *ping* como ilustra a Figura 5.32.

```
[root@freepbx ~]# ping 192.168.1.193
PING 192.168.1.193 (192.168.1.193) 56(84) bytes of data.
64 bytes from 192.168.1.193: icmp_seq=1 ttl=64 time=0.768 ms
64 bytes from 192.168.1.193: icmp_seq=2 ttl=64 time=0.374 ms
64 bytes from 192.168.1.193: icmp_seq=3 ttl=64 time=0.391 ms
64 bytes from 192.168.1.193: icmp_seq=4 ttl=64 time=0.370 ms
```

Figura 5.32: Teste de conectividade entre servidores

5.6.2 Configurações na interface web

Visto que existe a possibilidade de interagir visualmente com os servidores *FreePBX*, através da GUI oferecida na interface Web, de modo a ter contacto com uma gama mais ampla de tecnologias, foi optada por ser usada esta abordagem ao invés da tradicional linha de comandos.

Quando aberta a página *web* do servidor foi configurado um utilizador que vai funcionar como administrador para que se possa alterar e adicionar novas configurações, como mostra a Figura 5.33.

The screenshot shows the 'Initial Setup' section of the FreePBX Administration interface. It includes fields for 'Administrator User' (Username: Administrator, Password: 123456, Confirm Password: 123456), 'System Notifications Email' (Notifications Email address: 43202138955@sec.pt), and 'System Identification' (System identifier: VoIP Server).

Figura 5.33: Configuração da conta de administração da interface Web

Para que se possa haver a comunicação entre os dois servidores através do protocolo IAX2 foi necessário adicionar um *IAX2 Trunk* que permitesse o estabelecimento de chamadas VoIP entre os dois dispositivos, como mostra a Figura 5.34.

The screenshot shows the 'IAX2 Settings' tab with a new trunk named 'IAx2-Trunk1'. Configuration options include 'Hide CallerID' (Yes), 'CID Options' (Allow Any CID selected), 'Maximum Channels' (1), 'Asterisk Trunk Dial Options' (Override selected), 'Continue if Busy' (Yes), 'Disable Trunk' (Yes), and 'Monitor Trunk Failures' (Yes).

Figura 5.34: Adição de um *IAX2 Trunk* no primeiro servidor

No primeiro servidor, com o objetivo de contactar o servidor onde se encontra o segundo cliente VoIP, foi preciso proceder a algumas configurações na secção *Outgoing* do *IAX2 Trunk*, nomeadamente a definição do endereço IP do segundo servidor como mostra a Figura 5.35

The screenshot shows the 'Outgoing' tab of the IAX2 Trunk configuration. The 'Trunk Name' is set to 'PBX1'. In the 'PEER Details' section, the configuration parameters are listed as follows:

```
host=192.168.1.195
username=PBX1
secret=sr2pbx
type=peer
qualify=yes
context=from-trunk
disallow=all
allow=gsm
```

Figura 5.35: *Outgoing IAX2 Trunk* no primeiro servidor

Adicionalmente, na secção *Incoming* do *IAX2 Trunk*, também foram configurados alguns detalhes que permitam este servidor receber chamadas vindas do servidor com o endereço IP representado na Figura 5.36

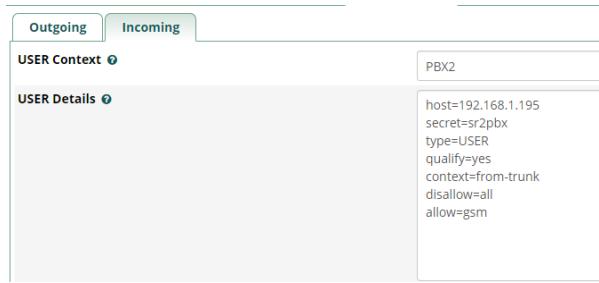


Figura 5.36: Incoming IAX2 Trunk no primeiro servidor

No segundo servidor, as configurações serão realizadas de forma semelhante, apenas com as adaptações necessárias para o servidor em questão. Exemplo disso pode ser visto na Figura 5.37 onde existe a modificação do *IAX2 Trunk*.



Figura 5.37: Adição de um IAX2 Trunk no segundo servidor

Tal como a configuração anteriormente apresentada para o servidor *PBX01*, no segundo servidor foram configuradas as mesmas definições com alteração apenas do endereço IP e do utilizador, como pode ser visto na Figura 5.38.

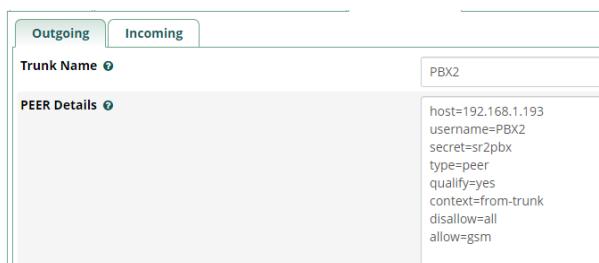


Figura 5.38: Outgoing IAX2 Trunk no segundo servidor

De igual modo, nesta configuração como o pretendido era receber chamadas do primeiro servidor foi esse o endereço IP colocado para que se possa estabelecer a comunicação, como mostra a Figura 5.39



Figura 5.39: Incoming IAX2 Trunk no segundo servidor

A definição das *Outbound Routes*, quando se utiliza o protocolo IAX2 são particularmente importantes. Estas referem-se aos caminhos ou configurados num servidor VoIP para determinar como as chamadas de saída são tratadas.

Uma vez que estas rotas estejam determinadas o servidor seleciona um *Trunk*, adequado a utilizar para a chamada. Neste caso vai utilizar o *Trunk* configurado na Figura 5.40.

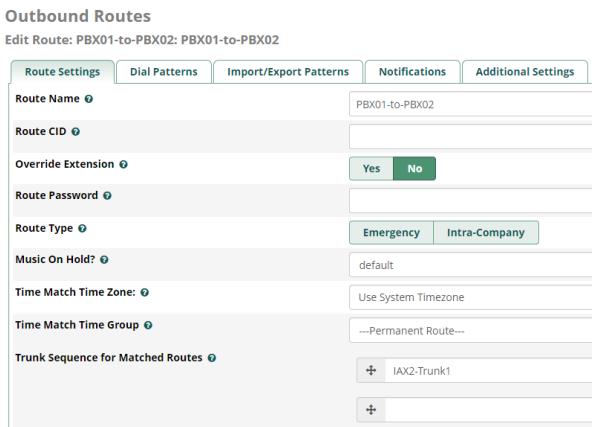


Figura 5.40: Configuração de Outbound Routes no primeiro servidor

Para que uma chamada seja considerada como válida a ser transmitida pela rota acima definida foi necessário primeiro definir regras, chamadas de *Dial Patterns*, que são definidos com base nos dígitos marcados pelo autor da chamada e são utilizados para determinar a forma como a chamada deve ser processada e para onde deve ser enviada, como pode ser visto na Figura 5.41

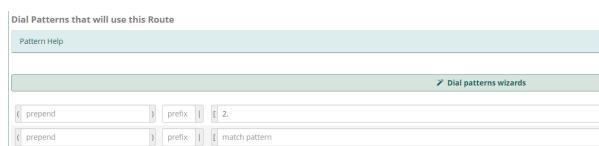


Figura 5.41: Dial patterns que utilizam a rota do servidor PBX01

Como o objetivo da experiência passava pela capacidade de haver comunicação em ambos os sentidos, no servidor *PBX02* a rota configurada corresponde ao servidor *PBX01*, como se encontra ilustrado na Figura 5.42

Outbound Routes

Edit Route: PBX02-to-PBX01: PBX02-to-PBX01

Route Settings	Dial Patterns	Import/Export Patterns	Notifications	Additional Settings
Route Name	PBX02-to-PBX01			
Route CID				
Override Extension	Yes No			
Route Password				
Route Type	Emergency Intra-Company			
Music On Hold?	default			
Time Match Time Zone:	Use System Timezone			
Time Match Time Group	---Permanent Route---			
Trunk Sequence for Matched Routes	<input type="button" value="+"/> IAX2-Trunk2 <input type="button" value="+"/>			

Figura 5.42: Configuração de *Outbound Routes* no segundo servidor

No caso do presente servidor, visto que este vai receber chamadas do servidor *PBX01*, o padrão que se pretende validar é, se caso o número que tenta contactar o cliente começa pelo número 1, como ilustra a Figura 5.43

Dial Patterns that will use this Route

Pattern Help

✓ Dial patterns wizards

(prepend)	prefix	(1.
(prepend)	prefix	(match pattern

Figura 5.43: *Dial patterns* que utilizam a rota do servidor PBX02

De acordo com o *Dial Plan* configurado anteriormente, para corresponder às suas normas, no servidor *PBX01* os utilizadores têm de ser configurados com a extensão 1XX, para que as suas chamadas possam chegar ao servidor vizinho, tal como mostra na Figura 5.44.

Extension: 100

General	Voicemail	Find Me/Follow Me	Advanced	Pin Sets	Other
Edit Extension					
This device uses CHAN_SIP technology listening on Port 5060 (UDP)					
Display Name	Miguel				
Outbound CID	100				
Emergency CID					
Secret	XXXXXXXX				

Figura 5.44: Adição do utilizador com a extensão 100

Por outro lado, no servidor *PBX02*, os utilizadores têm de ser configurados com a extensão 2XX, como ilustra a Figura 5.45, pelo mesmo motivo referido anteriormente.

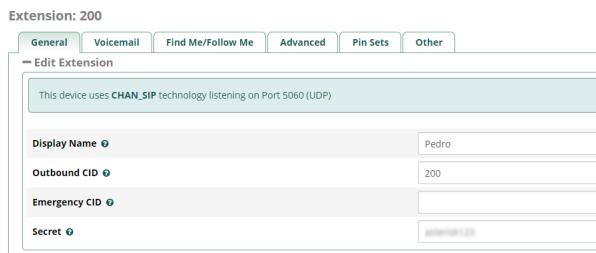


Figura 5.45: Adição do utilizador com a extensão 200

5.6.3 Chamada de teste

Para validar que todas as configurações foram feitas corretamente, e que é possível o estabelecimento de uma chamada VoIP entre os dois utilizadores, foram usadas as mesmas ferramentas que no protocolo SIP.

Efetuando uma chamada da extensão 200, que se encontra no *MicroSIP*, para a extensão 100, que se encontra no *Zoiper*, o resultado foi o ilustrado na Figura 5.46.

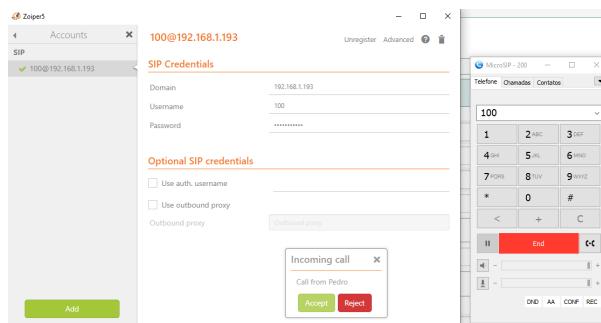


Figura 5.46: Chamada efetuado com o protocolo IAX2

Através da análise dos pacotes no capítulo 6 é possível obter uma melhor compreensão de como funciona este processo de comunicação.

Capítulo 6

Análise do tráfego *WireShark*

Neste capítulo são apresentadas as capturas de tráfego realizadas para entender como se processa uma chamada VoIP. O objetivo era analisar o tráfego de rede que é gerado durante uma chamada para identificar os protocolos e mensagens que são utilizados.

6.1 Captura do tráfego

De forma a possibilitar uma aprendizagem mais aprofundada de como se processa uma comunicação VoIP, foi necessário capturar o tráfego de rede que ocorre durante a chamada. Para isso, foi usado o *Wireshark* para capturar todo o tráfego UDP que passa pela interface de rede do servidor *Asterisk* e *FreePBX* através do comando *tcpdump*. Os filtros aplicados ao comando permitem especificar alguns detalhes da captura.

- *-w* - Guarda a captura de tráfego num arquivo passado como argumento.
- *-p* = Captura apenas pacotes IP.
- *-n* - Não resolve os endereços IP para nomes de domínio.
- *-s 0* - Captura todos os pacotes, independentemente de seu tamanho.
- *udp* - Captura apenas pacotes UDP.

O comando final a ser executado para a captura foi *tcpdump -w - -p -n -s 0 udp > /tmp/capture-asterisk.pcap*, sendo o argumento final o local onde o arquivo vai ser guardado, bem como o nome do mesmo.

6.2 Chamada com o protocolo *SIP*

O protocolo SIP funciona com comunicação bidirecional. Para cada mensagem SIP, um dispositivo envia um pedido e o outro dispositivo recebe e responde posteriormente

sendo a sinalização uma parte fundamental de uma chamada VoIP. Isto significa que, o SIP estabelece uma chamada ligando dois endereços SIP antes de transferir dados entre os interlocutores, o chamador e o recetor.

Na Figura 6.1 é possível ver como é feita a comunicação entre dois clientes através deste protocolo.

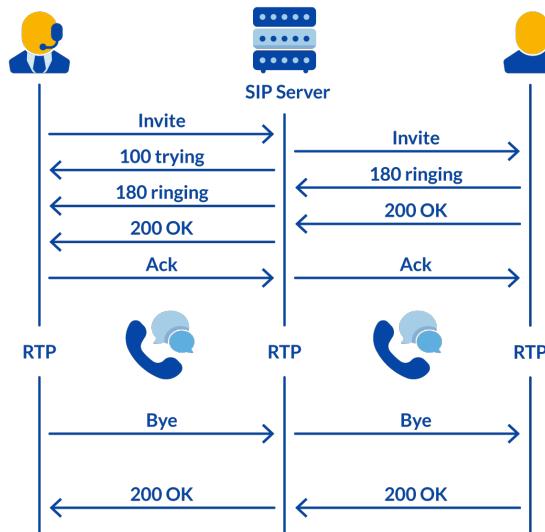


Figura 6.1: Exemplo de uma chamada usando o SIP

O SIP utiliza frequentemente o UDP para a maior parte das comunicações, pois permite uma transmissão mais rápida, pelo que as chamadas podem começar mais depressa. No entanto, o UDP não garante a entrega de mensagens, pelo que algumas mensagens (pacotes) podem perder-se. Nesses casos é necessário recorrer ao TCP quando a fiabilidade é crítica.

6.2.1 Inicialização da chamada

Este processo é o primeiro passo para estabelecer uma chamada usando SIP no *Asterisk* e envolve a troca de mensagens entre o indivíduo que liga e o que recebe, e os *SIP proxies* responsáveis pelo encaminhamento da chamada.

- *1º Etapa* - O autor da chamada inicia a chamada enviando uma mensagem SIP *INVITE* para o endereço SIP da parte chamada. Esta mensagem contém informações sobre o chamador, a parte chamada e os parâmetros de chamada pretendidos.
- *2º Etapa* - A tecnologia de comunicação do recetor da chamada começa a avisá-lo de que está a receber uma chamada através do *180 Ringing* ao qual este devolve uma resposta com um *200 OK* ao servidor do autor da chamada.

- 3^o Etapa - O servidor do autor da chamada ao receber a mensagem que do outro a ligação foi aceite envia uma mensagem de *ACK* para que a comunicação entre ambas as partes possa começar.

Estes pacotes encontram-se ilustrados na Figura 6.2 correspondente a uma captura feita aquando a ligação de um cliente ao outro.

192.168.1.100	192.168.1.200	SIP/SDP	1074 Request: INVITE sip:1001@192.168.1.200:5060
192.168.1.200	192.168.1.100	SIP	579 Status: 100 Trying
192.168.1.200	192.168.1.100	SIP	595 Status: 180 Ringing
192.168.1.100	192.168.1.69	SIP	604 Status: 180 Ringing
192.168.1.200	192.168.1.100	SIP/SDP	903 Status: 200 OK (INVITE)
192.168.1.100	192.168.1.200	SIP	454 Request: ACK sip:1001@192.168.1.200:5060

Figura 6.2: Inicialização da chamada com o protocolo SIP

6.2.2 Transmissão de áudio

Após a troca de mensagens SIP de convite e confirmação, a chamada é estabelecida. Durante esta etapa, os pacotes trocados entre ambos os clientes é feito com recurso ao protocolo RTP que contem os dados de áudio que são enviados entre os servidores *Asterisk*, de forma contínua, representando o áudio bidirecional. Cada pacote RTP contém uma sequência de amostras de áudio e informações de tempo com o objetivo de garantir a sincronização.

Como mostra a captura tirada na Figura 6.3, com o auxílio do *Wireshark*, foi possível tirar algumas conclusões de como se sucede a transmissão de áudio entre os servidores.

Source	Destination	Protocol	Length	Info
192.168.1.100	192.168.1.69	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x1D82A56, Seq=12928, Time=1760
192.168.1.200	192.168.1.100	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x1772B50F, Seq=566, Time=1920
192.168.1.100	192.168.1.69	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x1D82A56, Seq=12929, Time=1920
192.168.1.200	192.168.1.100	RTP	214	PT=ITU-T G.711 PCMU, SSRC=0x1772B50F, Seq=567, Time=2080

Figura 6.3: Captura dos pacotes RTP do protocolo SIP

Cada pacote RTP é constituído por um cabeçalho e uma carga útil. O cabeçalho, colocado no início do pacote, contém informações críticas, como a identificação da fonte, o número de sequência e o tipo de carga útil. Tal informação pode ser vista no painel de informação adicional de um pacote RTP, como ilustra a Figura 6.4.

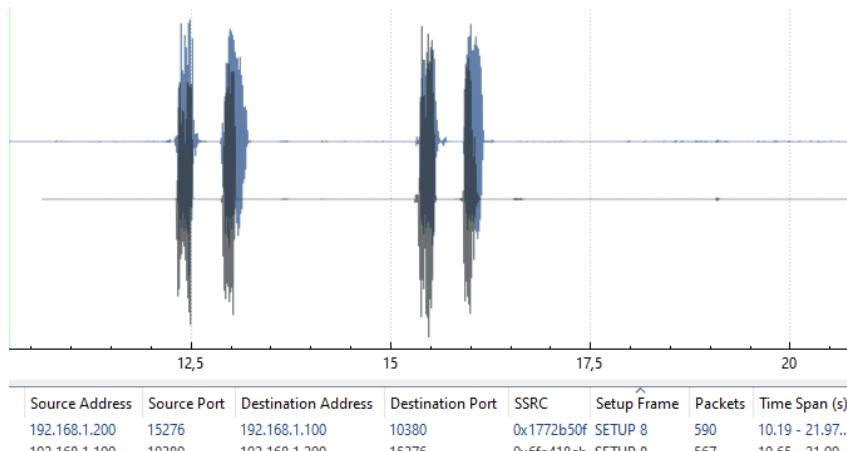
Por outro lado, a carga útil contém os dados de áudio propriamente ditos, divididos em pequenos pedaços geriveis. O número de sequência é crucial para resolver problemas de entrega fora de ordem, pois ajuda a manter a ordem correta dos pacotes na extremidade receptora e assim assegurar que não há qualquer tipo de falha na transmissão de áudio.

A Figura 6.5 mostra a gravação da chamada efetuada entre dois clientes, capturada com auxilio do *Wireshark*, de forma a assegurar e confirmar que não existe qualquer tipo de *jitter* ou cortes durante a comunicação.

```

▼ Real-Time Transport Protocol
  > [Stream setup by SDP (frame 8)]
    10... .... = Version: RFC 1889 Version (2)
    ..0. .... = Padding: False
    ...0 .... = Extension: False
    .... 0000 = Contributing source identifiers count: 0
    0.... .... = Marker: False
    Payload type: ITU-T G.711 PCMU (0)
    Sequence number: 575
    [Extended sequence number: 66111]
    Timestamp: 3360
    Synchronization Source identifier: 0x1772b50f (393393423)
    Payload [truncated]: fe7d7eff7efffd7c7e7e7c7dfcfefdf7c7

```

Figura 6.4: Informações dos pacotes RTP**Figura 6.5:** Gravação do áudio da chamada

6.2.3 Encerramento da chamada

Por último, quando a chamada é encerrada, uma mensagem SIP de encerramento (*BYE*) é enviada pelo servidor de origem à qual o servidor de destino responde com uma mensagem **200 OK** para confirmar o encerramento da chamada, como pode ser visto na Figura 6.6.

Source	Destination	Protocol	Length	Info
192.168.1.200	192.168.1.100	SIP	658	Request: BYE sip:1000@192.168.1.100:5060
192.168.1.100	192.168.1.200	SIP	510	Status: 200 OK (BYE)
192.168.1.100	192.168.1.69	SIP	487	Request: BYE sip:1000@192.168.1.69:54417;transport=UDP
192.168.1.69	192.168.1.100	RTCP	90	Receiver Report Source description Goodbye
192.168.1.69	192.168.1.100	SIP	390	Status: 200 OK (BYE)

Figura 6.6: Fim da chamada usando o SIP

6.3 Chamada com o protocolo *IAX2*

Um outro protocolo disponível para comunicação VoIP em servidores *Asterisk* é o IAX2, usado para estabelecer, manter e encerrar chamadas VoIP entre dois servidores *FreePBX*. O protocolo usa um único canal UDP para transportar tanto a sinalização quanto os dados de voz, o que o torna mais eficiente e reduz a latência.

Um exemplar de uma comunicação feita entre dois utilizadores por meio de um servidor IAX2 pode ser visualizado na Figura 6.7.

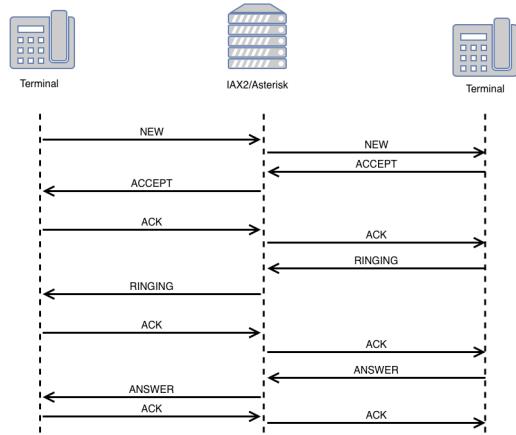


Figura 6.7: Exemplo de uma chamada usando o IAX2

6.3.1 Inicialização da chamada

Este processo é o primeiro passo para estabelecer uma chamada usando IAX2 no *Asterisk* e envolve a troca de mensagens entre o indivíduo que liga e o que recebe, através dos servidores *FreePBX* responsáveis pelo encaminhamento da chamada.

- *1^o Etapa* - O pacote *NEW*, que contém informações sobre alguns parâmetros da chamada incluindo o endereço do autor da chamada e do receptor, é a primeira mensagem enviada numa chamada IAX2. O segundo servidor, ao receber o pacote anterior, responde com o pacote *ACCEPT* indicando que recebeu a mensagem com sucesso e que está disponível para receber chamadas.
- *2^o Etapa* - O servidor do autor chamada de forma a confirmar a receção com sucesso do pacote envia um *ACK* momento no qual, o dispositivo de comunicação do receptor da chamada, começa a tocar de forma a avisá-lo de que estão a tentar comunicar com ele através do pacote *180 Ringing*.
- *3^o Etapa* - No momento em que a chamada é aceite é enviado um pacote *ANSWER* para o servidor do autor da chamada que responde com um *ACK* de forma a confirmar de que recebeu a mensagem com sucesso e de que a chamada pode iniciar.

Estes pacotes encontram-se ilustrados na Figura 6.8 correspondente a uma captura feita aquando a ligação de um cliente ao outro.

192.168.1.195	192.168.1.193	IAX2	144 IAX, source call# 8352, timestamp 10ms NEW
192.168.1.193	192.168.1.195	IAX2	71 IAX, source call# 12154, timestamp 10ms ACCEPT
192.168.1.195	192.168.1.193	IAX2	60 IAX, source call# 8352, timestamp 10ms ACK
192.168.1.193	192.168.1.195	IAX2	54 Control, source call# 12154, timestamp 13ms RINGING
192.168.1.193	192.168.1.195	IAX2	54 Control, source call# 12154, timestamp 16ms RINGING
192.168.1.195	192.168.1.193	IAX2	60 IAX, source call# 8352, timestamp 16ms ACK
192.168.1.193	192.168.1.195	IAX2	54 Control, source call# 12154, timestamp 3354ms ANSWER
192.168.1.195	192.168.1.193	IAX2	60 IAX, source call# 8352, timestamp 3351ms ACK

Figura 6.8: Inicialização da chamada com o protocolo IAX2

6.3.2 Transmissão de áudio

Tal como no protocolo anterior, após estabelecida a chamada, a transmissão de áudio continua a ser feita através do protocolo RTP, como mostra a captura tirada na Figura 6.9.

192.168.1.193	192.168.1.174	RTP	87 PT=GSM 06.10, SSRC=0x2A3DC1FE, Seq=31234, Time=63360
192.168.1.174	192.168.1.193	RTP	87 PT=GSM 06.10, SSRC=0xA0A04187, Seq=33934, Time=1355212094

Figura 6.9: Captura dos pacotes RTP no protocolo IAX2

6.3.3 Encerramento da chamada

Por último, quando a chamada é encerrada, uma mensagem IAX2 de encerramento (*HANGUP*) é enviada pelo servidor de origem à qual o servidor de destino responde com uma mensagem **ACK** para confirmar o encerramento da chamada, como pode ser visto na Figura 6.10.

192.168.1.195	192.168.1.193	IAX2	60 IAX, source call# 8352, timestamp 17702ms HANGUP
192.168.1.193	192.168.1.195	IAX2	54 IAX, source call# 12154, timestamp 17702ms ACK

Figura 6.10: Fim da chamada usando o IAX2

Capítulo 7

Conclusões

Este capítulo apresenta uma reflexão de todo o trabalho desenvolvido para o cumprimento e conclusão do projeto B. São apresentados os objetivos alcançados e os conhecimentos acerca da plataforma sobre a qual trabalhámos, bem como as suas potencialidades.

7.1 Objetivos alcançados

Ao longo do projeto foi possível adquirir algum *know-how* sobre o *Asterisk* e algumas das potencialidades que ele envolve, como realização de chamadas entre dois utilizadores, gravação das mesmas e a configuração das regras para definir esses comportamentos durante uma chamada. Com o auxílio do *Wireshark*, foi possível ver os pacotes (protocolos) trocados durante uma chamada e compreendê-los (qual a finalidade e o seu conteúdo na experiência em questão).

7.2 Opinião acerca da tecnologia estudada

Foi extremamente útil a realização do projeto, uma vez que todo o seu ambiente, bem como os recursos que lhe estão relacionados são interessantes no quotidiano. O *Asterisk* é uma solução de telefonia VoIP flexível, uma vez oferece suporte a diversos protocolos como o SIP e o IAX2, proporcionando eficiência nas comunicações internas e externas.

Referências

- [1] Asterisk. *FreePBX*. URL: <https://www.asterisk.org/asteriskexchange/freepbx/> (acedido em 10/01/2024).
- [2] Zoiper Community. *What is Exactly a Zoiper?* URL: <https://community.zoiper.com/2966/what-is-exactly-a-zoiper> (acedido em 10/01/2024).
- [3] ExtraHop. *Real-Time Transport Protocol (RTP)*. URL: <https://www.extrahop.com/resources/protocols/rtp/> (acedido em 10/01/2024).
- [4] fcc. *voice-over-internet-protocol-voip*. URL: <https://www.fcc.gov/general/voice-over-internet-protocol-voip> (acedido em 11/10/2023).
- [5] VoIP Info. *IAX*. URL: <https://www.voip-info.org/iax/> (acedido em 10/01/2024).
- [6] Liveagent. *MicroSIP*. URL: <https://www.liveagent.com/integrations/micsip/> (acedido em 10/01/2024).
- [7] TechTarget. *private branch exchange (PBX)*. URL: <https://www.techtarget.com/searchunifiedcommunications/definition/private-branch-exchange> (acedido em 10/01/2024).
- [8] techtarget. *Session-Initiation-Protocol*. URL: <https://www.techtarget.com/searchunifiedcommunications/definition/Session-Initiation-Protocol> (acedido em 10/01/2024).
- [9] techtarget. *VoIP*. URL: <https://www.techtarget.com/searchunifiedcommunications/definition/VoIP> (acedido em 11/10/2023).
- [10] WINSSCP. *WinSCP Introduction*. URL: <https://winscp.net/eng/docs/introduction> (acedido em 10/01/2024).
- [11] Wireshark. *What is Wireshark?* URL: https://www.wireshark.org/docs/wsug_html_chunked/ChapterIntroduction.html#ChIntroWhatIs (acedido em 10/01/2024).