André Filipe Ramos Estevam

Licenciado em Ciências da Engenharia Eletrotécnica e de Computadores

GSM-SIP Gateway

Dissertação para obtenção do Grau de Mestre em

Engenharia Eletrotécnica e de Computadores

Orientador: Rodolfo Oliveira, Professor Auxiliar,
Faculdade de Ciências e Tecnologia
da Universidade Nova de Lisboa

Júri

Presidente: João Rosas, Professor Auxiliar
Arguente: Luis Bernardo, Professor Auxiliar com Agregação

FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Setembro, 2017

GSM-SIP Gateway

Este documento foi gerado utilizando o processador (pdf)LaTeX, com base no template "unlthesis" [1] desenvolvido no Dep. Informática da FCT-NOVA [2].
[1] https://github.com/joaomlourenco/unlthesis [2] http://www.di.fct.unl.pt

Agradecimentos

Em todo o percurso que foi a criação deste documento, foram tantas as adversidades como os momentos felizes que me trazem agora ao final deste capítulo da minha formação académica. E por certo não estaria aqui sem a ajuda e o apoio de quem acreditou em mim.

Em primeiro lugar gostaria de agradecer ao meu orientador, Rodolfo Oliveira, por toda a orientação, critícas, partilha de conhecimento e todo o apoio ao longo desta dissertação, com todo o profissionalismo e rigor que só um excelente professor tem.

Em seguida, quero agradecer à Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa, e a todo o seu corpo docente, por todos estes anos que me acolheram e por toda a formação e conhecimento que me transmitiram, e igualmente por todas as oportunidades que me proporcionaram.

Da mesma forma, a todos os meus colegas com os quais estes anos não seriam os mesmos. Ao Bernardo Lima, Bruno Balixa, Diogo Sousa, Fábio Lopes, Gonçalo Queiroz e ao João Costa, que logo desde o primeiro ano se revelaram pessoas incríveis e amigos extraordinários.

A todos os meus colegas do NEEC, com especial destaque para o José Araújo, João Neves, Adriana Chambel, João Pombas, Alexandra Videira e João Barata, com quem aprendi bastante e vivi momentos únicos de união e companheirismo. E da mesma forma ao Rodrigo, Beatriz, Carolina, Alexandre e Alina, por este último ano cheio de desafios e aventuras que enfrentámos e conseguimos superar. Foi um prazer trabalhar com vocês e todos os outros membros, e levo comigo tudo o que aprendi convosco.

Com o mesmo sentimento, agradeço à Daniela Oliveira, David Mestre, Duarte Segurado, Fábio Carmo, Fábio Silva, Gonçalo Freitas e Joana Barruncho, por todos os momentos nas aulas, no estudo, horas de almoço, jantares e tudo mais. Em especial ao Miguel Prego, como meu padrinho de praxe e um verdadeiro irmão, e à Gisela Seixas, amiga do coração. Que a French Toast Mafia continue sempre unida e com a alegria de sempre.

O maior e infinito agradecimento vai sem dúvida para os meus pais guerreiros e à minha princesa Carolina, os meus pilares nesta vida. Não há palavras no mundo que descrevam o quanto estou grato por estarem sempre comigo. E estendo a toda a minha família, em especial aos meus avós, aos que partiram e aos que ainda cá estão, e aos meus padrinhos Fernando e Ana.

Não podia deixar de agradecer aos meus eternos companheiros Miguel Pinheiro e Laura, que sempre me apoiaram e nunca me deixaram ir abaixo em nenhum momento.

Da mesma forma, e com mesmo sentimento, agradeço a todos os meus amigos Pandas pela amizade e apoio incondicional.

Agradeço ao João Amorim, Pedro Antão, Diogo Tristão e todos os meus amigos do Alhandra Sporting Club que sempre acreditaram em mim e sempre estiveram presentes quando mais precisei. E ao Bruno Pereira, que já não estás entre nós, mas que estás sempre no meu coração meu irmão.

Para finalizar, queria agradecer à Sara Miguel, amiga do coração, ao Fernando, à Marilia, à Tania, à família Marafuz, ao senhor Gaspar e à Ivone, à Clotilde, à Jessica Roque e à equipa Hotbox, ao Nuno Pereira, Marco Moreira, Duarte Gonçalves, Filipe Perestrelo, ao Sérgio e à família Faria, à Lai e ao João, e ao João Brito. Por fim, e em especial, à Ana, por todo o apoio e críticas que ajudaram a criar esta dissertação. E também por todas as horas de conversa, pelo carinho, por todos os momentos e pela a alegria constante que fizeram e sempre farão de mim uma pessoa melhor.

A todos estes e a todos os que me esqueci, o meu eterno obrigado.

Resumo

Os sistemas de comunicação móveis estão actualmente em constante desenvolvimento. Os utilizadores procuram cada vez mais elevada eficiência e qualidade de experiência de serviços ao menor custo possível. Contudo, verifica-se um aumento de utilizadores activos na rede, sendo necessário adaptar e expandir a rede. Para satisfazer as necessidades dos utilizadores têm ocorrido várias alterações em toda a infraestrutura das redes celulares. Estas alterações permitem que a infraestrutura da rede se possa adaptar tecnicamente ao aumento de utilizadores, apresentando menores custos e encurtando a duração dos ciclos de actualização de rádio, *core* e dos serviços.

Esta dissertação aborda a temática da unificação entre o rádio e a rede de forma a atingir um elevado desempenho e capacidade de adaptação a custo reduzido. Os Rádios Definidos por *Software*, SDRs, permitem substituir parte das funções asseguradas por *hardware* específico em *software*, de forma a alcançar flexibilidade e maior eficiência espectral a custos menores.

A utilização dos SDRs permitem a implementação de uma rede GSM sem um *Mobile Switching Center*, MSC, passando as funções deste elemento de rede para o domínio do *software*. A comutação de circuitos é substuída por comutação ao nível SIP (*Session Initiation Protocol*), permitindo a integração com uma rede de serviços por IP, como é o caso de uma rede *core IMS*.

Este trabalho apresenta os passos efectuados no desenvolvimento de um protótipo experimental que oferece serviços de chamadas por voz através de uma rede de comutação de pacotes IP, integrando o rádio de uma rede GSM convencional.

Abstract

Mobile communication systems are currently evolving. Users seek higher efficiency and better user experience in terms of quality of service with minimal cost. Furthermore, the number of active users is growing every day. Therefore, there has been a great effort in changing the network infrastructure to support the user's needs. These changes allow the network to adapt itself to the increasing number of users, with lower costs and shorter upgrade cycle periods at the radio, at the core and at the service level.

This dissertation explores the unification of the radio level with the core to achieve higher performance and adaptability at lower cost. The Software Defined Radios, SDRs, allow to replace some hardware based functionalities of the radio level by software, making it possible to achieve greater flexibility and spectral optimization at lower cost.

SDRs allow the implementation of a GSM network without a Mobile Switching Center, MSC. By moving the circuit switching to packet switching, and using the SIP protocol, it is possible to integrate the GSM network in an IP based network, as it is the case of a IP Multimedia Services network, IMS.

In this document, we present a solution that unifies the radio level and the core level of the network, so that we can achieve greater performance and scalability at a lower cost. A prototype is reported, which offers voice call services over a IP packet switching network and integrates the radio level of a standard GSM network.

Índice

Lista de Figuras

Lista de Tabelas

Siglas

3GPP 3rd Generation Partnership Project.

ACK Acknowledgement.

ADC Analog to Digital Converter.

ALG Application Level Gateway.

AS Application Server.

AuC Authentication Center.

BC Billing Center.

BGCF Breakout Gateway Control Function.

BGP Border Gateway Protocol.

BSC Base Station Controller.

BTS Base Transceiver Station.

CDR Call Detail Record Drivers.

CEL Call Event Log Drivers.

CIDR Classless Inter-Domain Routing.

Codec Coder-decoder.

CS Circuit Switching.

CSCF Call Session Control Function.

DAC Digital to Analog Converter.

DoS Denial of Service.

DSP Digital Signal Processor.

DUC Digital Up Converter.

EiR Equipment Identity Register.

FDMA Frequency Division Multiple Access.

FPGA Field Programmable Gate Array.

GGSN Gateway GPRS Support Node.

GMSC Gateway MSC.

GPP General Purpose Processors.

GPRS General Packet Radio Service.

GSM Global System for Mobile Communications.

HLR Home Location Register.

HSS Home Subscriber Server.

HTTP HyperText Transfer Protocol.

IBCF Interconnection Border Control Function.

I-CSCF Interrogating-Call Session Control Function.

IAX Inter Asterisk eXchange.

ICMP Internet Control Message Protocol.

IFc Initial Filter Criteria.

IGMP Internet Group Management Protocol.

IMS-MGW IMS-Media Gateway.

IM-SSF IP Multimedia-Service Switching Function.

IMS IP Multimedia Subsystem.

IMSI International Mobile Subscriber Identity.

IP Internet Protocol.

IPsec Internet Protocol Security.

ISC IMS Service Control.

ISDN Integrated Services Digital Network.

ISP Internet Service Provider.

ISUP ISDN User Part.

Kc Session Key.

Ki Individual Subscriber Authentication Key.

LA Location Area.

LRF Location Retrieval Function.

MGCF Media Gateway Control Function.

MILS Multiple Independent Levels of Security.

MIME Multipurpose Internet Mail Extensions.

MRFC Multimedia Resource Function Controller.

MSC Mobile Switching Center.

MSISDN Mobile Subscriber ISDN Number.

MSS Mobile Switching Center Server.

NAT Name Address Translation.

NSS Network Switching Subsystem.

OMC Operation and Maintenance Center.

OpenBTS Open Base Transceiver Station.

OpenBTSCLI Open Base Trasceiver Station Command Line Interface.

OS Operating System.

OSA Open Service Architecture.

OSA-SCS Open Service Architecture-Service Capability Server.

OSPF Open Shortest Path First.

OSS Operation Subsystem.

PCRF Policy and Charging Rules Function.

P-CSCF Proxy-Call Session Control Function.

PBX Private Branch eXchange.

PS Packet Switching.

QoS Quality of Service.

QoS-SDP Quality of Service-Session Description Protocol.

RDF Routing Determination Function.

RF Radio Frequency.

RSS Radio Station Subsystem.

RTCP Real-Time Transport Control Protocol.

RTP Real-Time Transport Protocol.

S-CSCF Serving-Call Session Control Function.

SDR Software Defined Radio.

SEG Security Gateway.

SGW Signaling Gateway.

SIM Subscriber Identification Module.

SIMD Single Instruction Multiple Data.

SIP Session Initiation Protocol.

SK Separation Kernel.

SLF Subscription Locator Function.

SMS Short Message Service.

SMTP Simple Mail Transfer Protocol.

TCP Transmission Control Protocol.

TDM Time Division Multiplexing.

TDMA Time Division Multiple Access.

TM Terminal Móvel.

TMSI Temporary Mobile Subscriber Identity.

TP Trigger Point.

UDP User Datagram Protocol.

URI Uniform Resource Identifier.

URL Uniform Resource Locator.

USB Universal Serial Bus.

VLR Visitor Location Register.

VoIP Voice over Internet Protocol.

Capítulo 1

Introdução

1.1 Motivação

Os sistemas de comunicações móveis são actualmente um dos grandes focos de investigação e desenvolvimento na área das telecomunicações. Os utilizadores procuram cada vez mais atingir elevados ritmos de transferência de dados e elevada eficiência em termos de qualidade de experiência, ao menor custo possível. De forma a satisfazer as necessidades do utilizador têm ocorrido várias mudanças por forma a melhorar as estações base (BTS), e toda a infraestrutura que compõe a rede celular.

Quanto às BTS, têm sido investigados e desenvolvidos sistemas que permitem melhorar o desempenho face às frequentes alterações das versões de rádio. Caminha-se presentemente para uma abordagem em que se elimina cada vez mais a utilização de circuitos (*hardware*) específicos para as diferentes etapas dos módulos de rádio. Esta abordagem é implementada através dos Rádios Definidos por *Software* (SDRs), permitindo adicionalmente a optimização espectral e a configuração dinâmica dos canais de frequência. Os SDRs substituem parte das funções asseguradas pelo *hardware* convencional por *software*, de forma a alcançar flexibilidade e processamento de sinal com baixa latência.

Por outro lado, as redes celulares têm sofrido uma enorme expansão. A necessidade de resposta ao aumento de utilizadores activos na rede tem levado a um aumento de estações base por quilómetro quadrado e a um esforço adicional na sua gestão. Assim, as tendências descritas anteriormente procuram reunir as condições para a unificação entre o rádio e a rede, de forma a alcançar um elevado desempenho e elevada capacidade de adaptação a custo reduzido.

Neste documento é reunida informação sobre as estações base e a infraestrutura das redes celulares, com o objetivo de auxiliar a implementação de uma plataforma experimental demonstrativa desta unificação.

Em primeiro lugar, são apresentadas noções gerais de SDR, onde são referidas as suas vantagens e desvantagens e a sua integração com a tecnologia celular (GSM). Segue-se uma secção que descreve um *software* aberto que implementa diferentes versões do nível de rádio de sistemas de segunda e terceira geração (OpenBTS), bem como a descrição de um servidor SIP (*Asterisk*), referindo a sua estrutura e as suas múltiplas funcionalidades. Os dois últimos tópicos descrevem os principais protocolos de sinalização e transporte das redes celulares de quarta geração (baseadas no *Internet Protocol* (IP)), e ainda a arquitectura para serviços multimédia normalizados para as redes de quarta geração, denominada *IP Multimedia Subsystem* (IMS).

1.2 Objectivos

Neste trabalho pretende-se criar um protótipo experimental que faça a integração do nível rádio GSM com uma rede *core* IP, neste caso uma rede IMS. O objectivo é providenciar serviços de chamada por voz e de mensagens curtas, SMS, através de uma rede de comutação de pacotes IP, mas que integre o rádio de uma rede GSM convencional.

É necessário configurar e analisar o sistema de rádio, constituído por uma BTS, que suporta a tecnologia GSM. Utilizando múltiplos terminais móveis, o objetivo é avaliar o desempenho da implementação dos módulos de rádio num equipamento SDR. Pretende-se ainda configurar um servidor SIP, de forma a permitir iniciar sessões de comunicação interactiva entre terminais ligados à BTS.

Após estas configurações seguir-se-á o estudo e implementação da integração entre a arquitectura OpenBTS e a rede IMS. A integração das duas plataformas irá procurar garantir a comunicação entre o sistema de rádio e a rede e, da mesma forma, a comunicação entre terminais activos na rede.

Por fim, é imprescindível realizar o desenho e implementação de cenários de teste para validar o modelo de integração.

Os cenários irão testar o serviço de chamadas por voz utilizando os dois terminais GSM referidos, e avaliar o desempenho da rede IP/IMS.

Pretende-se ainda testar toda a sinalização de controlo entre os terminais e a BTS, e entre a BTS e a rede IMS.

Capítulo 2

Trabalho Relacionado

2.1 Software Defined Radio

Rádios Definidos por *Software* (SDRs) é um conceito inovador que tem o objectivo de tornar os rádios convencionais mais flexíveis e reconfiguráveis, com vista a reutilizar o espectro limitado e possibilitar a configuração dinâmica dos canais de frequência via *software*, sem a necessidade de modificar o *hardware*. Este conceito pretende minimizar o *hardware* específico e generalizar o *hardware* necessário de forma a baixar os custos relacionados com os módulos de rádio, possibilitando a portabilidade de aplicações de *software* entre diferentes plataformas de *hardware* e sem problemas de compatibilidade. A flexibilidade estende-se também à actualização dos parâmetros dos módulos de rádio através de uma simples actualização do *software* instalado, não necessitando de nenhuma outra modificação no *hardware* específico dos módulos de rádio.

2.1.1 Hardware

A evolução para SDRs é motivada pelos consequentes avanços em tecnologias que possibilitam reconstruir o sistema na perspectiva de um programador de *software*. Nomeadamente, o desenvolvimento de conversores de Analógico-Digital (ADC) e de Digital-Analógico (DAC) de banda larga, que permitem amostrar os canais de rádio de forma a realizar um tratamento no domínio discreto, possibilitando o armazenamento directo da amostra em memória para posterior análise com *software* dedicado [10]. Assim, usando componentes de memória dedicada, é possível criar módulos de *software* que realizem o processamento digital de sinal das amostras recebidas, executados em GPP's (*General Purpose Processors*).

Os dispositivos SDR reduzem e simplificam o *hardware* necessário, deixando o processamento de sinal para o domínio digital. Os SDRs apresentam uma interface de rádio-frequência simplificada, que consiste apenas numa antena, um amplificador e um filtro passa-banda [10]. Esta interface é acompanhada pelos conversores ADC e DAC de banda larga que transferem o sinal para memória. Para completar é necessária uma unidade de processamento, tipicamente uma FPGA (*Field Programmable Gate Array*) ou um DSP (*Digital Signal Processor*) de alta velocidade [1].

Nas figuras seguintes é possível visualizar a estrutura do *hardware* de um dispositivo SDR, nomeadamente um circuito receptor de sinal e um circuito transmissor de sinal:

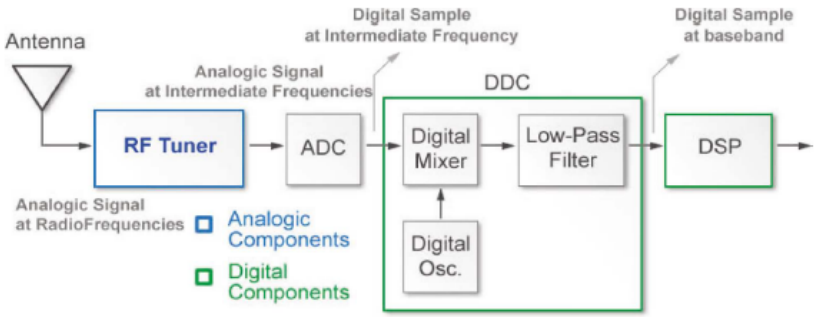

Figura 2.1: Diagrama de blocos de um receptor SDR [3].

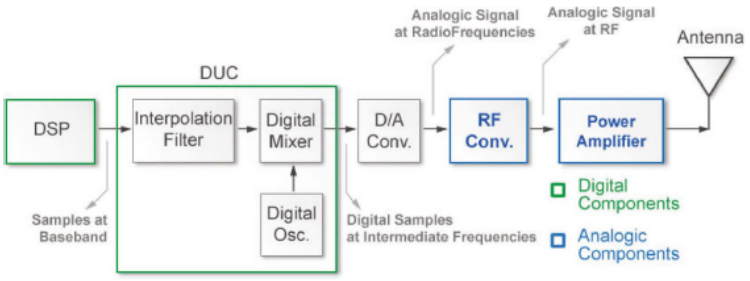

Figura 2.2: Diagrama de blocos de um transmissor SDR [3].

A estrutura do receptor SDR divide-se em três partes: uma interface de rádio frequência, constituída pela antena e pelo bloco *RF Tuner* (para tratamento das amostras de forma analógica), o bloco do conversor Analógico-Digital, o bloco *Digital Down Converter* (para conversão do sinal para bandas inferiores ou *downconversion*) e o bloco DSP (para processamento do sinal de forma mais rápida e eficiente).

Embora os dispositivos SDRs mais comuns operem como receptores de sinal, a tecnologia também integra esquemas de transmissão. O transmissor SDR apresenta a sequência inversa de funcionamento do circuito receptor, com destaque para algumas alterações: o bloco DUC (*Digital Up Converter*, para translação da banda base para frequências superiores), o conversor Digital- Analógico e o *RF Converter* [3].

O sinal de banda base, tipicamente gerado pelo DSP, é recebido do DUC. Após realizar o processo de *upconversion*, o DUC transfere o sinal para o DAC, que transforma as amostras para o domínio analógico. De seguida, o *RF Converter* irá converter o sinal para frequências altas e, posteriormente, é amplificado e direcionado para a antena. Desta forma, os restantes módulos de *hardware* dos rádios convencionais que não foram referidos (i.e., filtros, amplificadores, moduladores e desmoduladores, codificadores de sinal e do canal de frequência), que habitualmente eram implementados por *hardware* específico na BTS, podem ser implementados em *software* e executados em qualquer computador ou sistema distribuído [1].

Ao eliminar o *hardware* específico e generalizando o *hardware* necessário, a implementação é realizada somente em *software* e os seus custos são mais reduzidos pelos factores de escala associados à adopção de *hardware* generalizado. Porém, com a implementação em *software*, surgem novos desafios com complexidade acrescida, que exigem uma abordagem diferente da habitual.

2.1.2 Software

Os dispositivos SDRs abrem a possibilidade de implementar uma arquitectura distribuída possibilitando, por exemplo, que um módulo de *software*/aplicação possa ser executado numa variedade de processadores disponíveis no momento [11]. O módulo de *software* é uma abstração de um componente de *hardware* que, de alguma forma, processa o fluxo de amostras de dados recebido e retorna o resultado desse processamento para o próximo módulo.

Contudo, existem vários desafios e oportunidades relacionadas com uma arquitectura desta natureza. Em primeiro lugar, tem de existir portabilidade de aplicações para que possam ser movidas entre plataformas sem ser necessário modificar as mesmas, criando assim flexibilidade no sistema.

2.1.3 Portabilidade

Um dos entraves à portabilidade é o facto de não haver um *standard* para aplicações e dispositivos, como processadores, da plataforma. Cada processador tem funcionalidades e linguagens diferentes que podem ameaçar a sua compatibilidade com as aplicações [11]. Outro problema relacionado são múltiplos protocolos de transporte usados que, enquanto não existir um *standard*, irá sempre gerar problemas na portabilidade.

Para conseguir atingir a portabilidade em SDR é necessário definir um *standard* para as plataformas de forma a que as aplicações ou módulos de *software* possam ser movidos e reutilizados sem problemas de compatibilidade, tornando o SDR um sistema flexível.

Em alternativa a um *standard* também pode ser explorada a criação de camadas de abstração entre as aplicações e a plataforma, evitando indirectamente a incompatibilidade entre as mesmas.

2.1.4 Desenvolvimento de aplicações SDR

Em segundo lugar, existe o desenvolvimento das aplicações SDR [11]. Para um engenheiro de rádio sem qualquer formação em arquitecturas de *software*, desenvolver aplicações para SDRs pode tornar-se desafiante. O mesmo pode acontecer para um engenheiro de *software* em relação às comunicações de rádio.

Para o desenvolvimento de aplicações em SDR é necessário haver ferramentas de abstração da arquitectura do sistema de comunicações, para que seja possível ao engenheiro de *software* desenvolver os módulos de *software* pretendidos. No entanto, numa fase inicial, será necessário alguém com o conhecimento da arquitectura para criar estas ferramentas de abstração.

2.1.5 Requisitos computacionais

Por outro lado, é preciso rever os requisitos computacionais de uma implementação em SDR. Para implementar uma estação base de uma rede celular de segunda geração (GSM), factores como o consumo energético, a temperatura, o custo do equipamento e o espaço ocupado são fáceis de adaptar. Porém, quando se considera o desempenho do sistema em função desses factores, a implementação pode tornar-se um desafio prolongado.

Uma das soluções discutidas para o compromisso desempenho/consumo energético é aumentar o número de núcleos dos processadores da plataforma, com cada núcleo a ter o seu próprio processador de *Single Instruction Multiple Data*, SIMD, de modo a explorar o paralelismo dos algoritmos em execução [11]. Este processador contém vários fluxos de instruções e operam com vários fluxos de dados em execução ao mesmo tempo, o que ajuda a acelerar o processamento de sinal através da execução em paralelo.

Outra solução sugerida pelo mesmo autor, no mesmo prisma de explorar o paralelismo, passa por optimizar os momentos de execução das aplicações considerando o número de unidades de processamento disponíveis no momento.

2.1.6 Segurança

Por fim, a flexibilidade de um SDR apresenta elevados desafios no domínio da segurança, tanto para investigadores como para organizações de certificação e segurança de sistemas.

Um dos desafios de segurança está associado à possibilidade de carregar e instalar remotamente actualizações de *software* numa plataforma SDR, e haver a ameaça constante dessa actualização conter *software* malicioso que perturbe o normal funcionamento da plataforma [11]. Este facto agrava-se quando a actualização é feita com uma ligação sem fios, que expõe o sistema à obtenção ilegal do *software* ou mesmo a alteração de *software* durante a transferência, constituindo violações de privacidade e integridade. Uma das soluções para este problema é a utilização de certificados digitais, que consistem numa assinatura digital realizada por terceiros de confiança. Estes terceiros são verificados através uma cadeia de confiança com um certificado de raiz na plataforma. Contudo,

quando o certificado expira ou é revocado causa problemas para verificar as operações de *download* de *software*.

Outra solução publicada é a pré-execução do novo *software* descarregado numa *sandbox*, que consiste num mecanismo de segurança para executar programas que não foram verificados em separado, de forma a não danificar a máquina principal [11]. Porém, o desempenho deste mecanismo é subjetivo, dado que o código malicioso pode não revelar o seu comportamento neste teste.

Por outro lado, garantir que um sistema flexível e generalizado garanta alta segurança é uma tarefa complexa. Existe uma arquitectura que permite processar informação em diferentes níveis de segurança simultaneamente, denominada arquitectura *Multiple Independent Levels of Security* (MILS).

A importância desta arquitectura reside no *Separation Kernel* (SK), que permite que *software* localizado em partições diferentes seja executado no mesmo processador, em diferentes intervalos de tempo. Assim, este módulo permite a separação dos dados e limitação de danos, na medida em que um erro numa partição não afecte os processos nas outras partições.

O SK é o único módulo que consegue funcionar em modo de supervisor, e as outras partições funcionam em modo utilizador. Isto significa que as partições nunca poderão alterar parâmetros no SK, em qualquer circunstância. Nesta arquitectura o *hardware* de suporte necessário já se encontra disponível em microprocessadores comerciais, pelo que não é necessário fabricar *hardware* específico com custos adicionais.

Contudo, a arquitectura MILS demonstra algumas desvantagens no seu funcionamento:

- Elevado consumo de memória devido às partições separadas;
- Desempenho do processamento menos dinâmico devido à necessidade de garantir recursos de processamento a todas as partições;
- Elevado custo do *switches* devido ao elevado numero de processos separados em execução.

No entanto, com os avanços na tecnologia de processadores e dispositivos de memória é espectável que estas desvantagens percam impacto ao longo do tempo.

2.1.7 Global system for mobile communications(GSM)

As redes móveis de rádio digital, das quais o sistema GSM constitui um *standard* global, oferecem serviços de voz e mensagens de texto curtas a clientes com terminais móveis. O GSM opera maioritariamente na banda dos 900 MHz e 1800 MHz. Este sistema também suporta serviços de *roaming*, que consiste na possibilidade de usar o terminal para comunicar noutras redes, fora da sua própria rede, desde que suportem a tecnologia GSM.

O GSM é baseado na tecnologia de comutação de circuitos e combina os métodos *Time Division Multiple Access* (TDMA) e *Frequency Division Multiple Access* (FDMA) para transmitir sinal. O TDMA é um sistema em que cada canal de frequência (200 kHz) é dividido em 8 *slots* de tempo, cada um com 25 kHz, e assim cada utilizador utiliza um tempo específico na transmissão, impedindo problemas de interferência.

Por outro lado, o FDMA é um método que consiste em dividir a banda de frequência disponibilizada à BTS em vários canais de frequência individuais, com o objectivo de ser atribuído um canal para cada utilizador durante todo o tempo da sua ligação. A largura de banda de cada canal é dimensionada com uma "banda de guarda"entre frequências adjacentes para não haver a necessidade de lidar com desvios de frequência e minimizar a interferência entre canais adjacentes. De acordo com a norma GSM, a divisão será feita a partir de 25 MHz de banda em 124 canais espaçados por 200 kHz.

Na figura seguinte é possível visualizar a arquitectura de uma rede convencional GSM e os blocos que fazem parte da sua constituição. A arquitectura divide-se em 3 categorias: *Radio Station Subsystem* (RSS), ou o nível rádio, *Network Switching Subsystem* (NSS), o nível *core*, e *Operation Subsystem* (OSS), nível de operações. Nas secções seguintes, são descritas as três categorias.

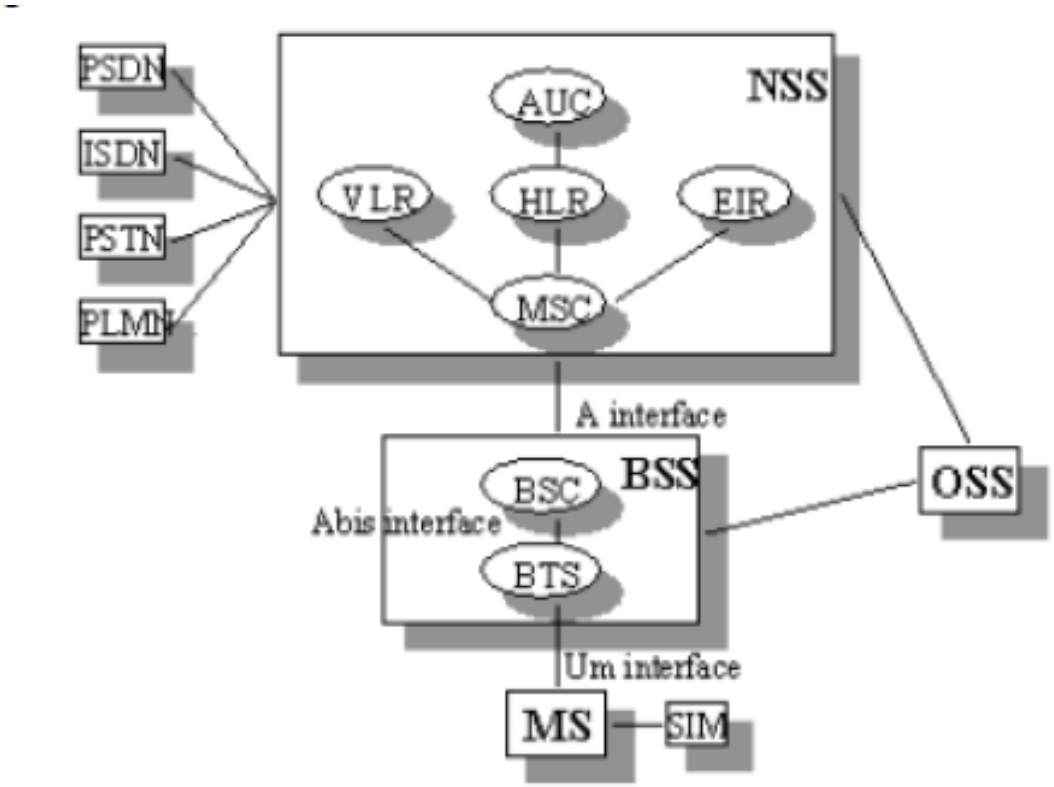

Figura 2.3: Arquitectura da rede GSM [6].

2.1.7.1 Radio Station Subsystem - RSS

Esta secção da rede é responsável por gerir os recursos de rádio, o tráfego e a sinalização entre um terminal e a NSS. É composta por 3 elementos: o Terminal Móvel (TM), a *Base Transceiver Station* (BTS), ou estação base, e o *Base Station Controller* (BSC).

O terminal móvel é constituído pelo equipamento terminal e pelo cartão *Subscriber Identification Module* (SIM). O cartão SIM guarda informação secreta e pessoal do utilizador e, quando ligado à rede, guarda também informação dinâmica que identifica o terminal na rede.

Antes de se ligar à rede, a informação contida no cartão SIM inclui os seguintes componentes/identificadores:

- *International Mobile Subscriber Identity* (IMSI): consiste num identificador do utilizador do terminal em função da sua rede ou operador;
- Ki: é uma chave secreta de autenticação individual do utilizador, utilizada para autenticar e/ou encriptar dados pessoais como os contactos telefónicos;

Após a ligação à rede, o SIM armazena nova informação para satisfazer os procedimentos de autenticação e encriptação no futuro:

- Kc: chave secreta de encriptação, usada como chave de sessão. É gerada através de um algoritmo de geração de chave de encriptação, utilizando a chave Ki e um número aleatório gerado pela rede;
- *Temporary Mobile Subscriber Identity* (TMSI): é uma identificação temporária atribuída ao utilizador que visita a rede. É utilizado na rede visitada em vez do IMSI para proteger a identidade do utilizador contra ameaças externas;
- *Location Area* (LA): é definida por um conjunto de células dentro das quais a localização do terminal é conhecida. Este parâmetro é guardado no VLR (*Visitor Location Register*) e no terminal, e é actualizada em cenários de mobilidade do terminal para outra LA;

A BTS contém o equipamento necessário para receber e enviar sinais de rádio como antenas, processadores e amplificadores de sinal, e equipamento para encriptação e decriptação de sinal [6]. As células são normalmente divididas em sectores de 120º, para alargar o espectro de frequências disponível em cada célula, usando cada sector com uma gama de frequências associada. O alcance das antenas define o tamanho da célula, que pode variar entre dezenas de metros até dezenas de quilómetros.

A determinação da área da célula é suportada pela distribuição de tráfego na zona e a procura de utilizadores pela rede. Assim, existindo maior tráfego, a célula terá uma área menor para evitar problemas de alocação de frequências e bloqueio de chamadas por

haver demasiados utilizadores na célula. Por outro lado, com a célula mais reduzida, é necessário mais equipamento para criar mais células na mesma zona.

O BSC é o controlador que gere um conjunto de BTSs [6]. As funções resumem-se à gestão dos recursos de rádio nas BTSs para manutenção da ligação entre utilizadores, de forma a evitar o término antecipado de chamadas activas ou falhas no envio de mensagens curtas. Efetua operações de multiplexagem e transcodificação dos canais de voz. Também efectua a comutação entre canais de rádio e *slots* de *Time Division Multiplexing* (TDM) das suas ligações com o *Mobile Switching Center* (MSC).

2.1.7.2 Network Switching Subsystem - NSS

Também chamada a rede central, é responsável pela comutação de chamadas e gestão da mobilidade dos terminais que constituem a rede. Permite ainda a interligação da rede celular com a rede fixa (PSTN). Na rede GSM, o NSS realiza comutação de circuitos para providenciar os serviços tradicionais desta rede, nomeadamente chamadas de voz e mensagens curtas.

Um dos módulos mais importantes desta secção é o *Mobile Switching Center* (MSC), que faz a ligação entre vários BSCs vizinhos e tem interfaces com todos os restantes elementos da NSS. Este módulo está também ligado à rede telefónica para providenciar conectividade entre utilizadores da rede celular e da rede fixa, bem como assegurar a conectividade entre MSCs da rede celular. O objectivo é tornar possível para cada utilizador móvel comunicar com qualquer outro utilizador no mundo, esteja este utilizador na rede de telefonia fixa ou um terminal móvel [6].

As funções do MSC incluem processamento e encaminhamento de chamadas de voz entre utilizadores ligados, bem como a sinalização da chamada para a gestão da mobilidade de utilizadores e coordenação de processos de *handover* entre BTSs. O MSC também faz a geração de registos para a taxação de serviços de chamadas de voz e de mensagens curtas.

Uma das interfaces com este módulo é o *Gateway MSC*, que tem o objectivo de interligar o MSC com outras redes públicas. A função de *Gateway* centra-se na interrogação à base de dados de utilizadores para verificar a localização e o estado do utilizador destinatário.

Esta base de dados corresponde ao *Home Location Register* (HLR), que contém informação de carácter permanente, como informações para configuração de canais de voz, e de carácter dinâmico, como a localização e estado dos utilizadores móveis [6]. Este elemento pode gerir centenas de milhares de utilizadores do operador de comunicações.

A informação de estado permanente consiste em:

- *Mobile Subscriber ISDN Number* (MSISDN), que consiste no número público do terminal móvel (ex: 96*******);

- Perfil de serviços do utilizador, como informações desvio de chamadas e barramentos;
- IMSI, como anteriormente referido, é um identificador do utilizador em função da sua rede ou operador;

Por outro lado, a informação de estado dinâmico inclui:

- Informação sobre a localização do terminal através do endereço do MSC/VLR onde este se encontra registado;
- Configuração de serviços subscritos, como por exemplo, o nº para o qual se efectua o desvio de chamada;

Existe ainda uma base de dados dinâmica, que é uma das interfaces do MSC, chamada *Visitor Location Register* (VLR), que tem o objectivo de diminuir o número de interrogações ao HLR e reduzir a sinalização na rede. É geralmente incorporado no MSC e disponibiliza, localmente, informação dos utilizadores dentro da sua área geográfica (LA).

Sempre que um terminal entra numa LA do VLR, este informa o HLR e requisita os serviços subscritos pelo utilizador a serem disponibilizados pelo MSC. O VLR é também responsável por monitorizar o estado do terminal: disponível, desligado ou ocupado. É uma das interfaces de suporte com mais impacto no MSC.

Quanto à autenticação de utilizadores o MSC tem uma interface com o *Authentication Center* (AuC), incorporado no HLR, e é uma base de dados que disponibiliza funções de autenticação e encriptação. Este recurso oferece chaves e algoritmos de encriptação para garantir a segurança das identidades dos utilizadores e de informação transmitida *over the air*.

Paralelamente ao AuC, a rede GSM possui o *Equipment Identity Register* (EiR), com funções semelhantes, mas direcionadas para o equipamento do utilizador. Este módulo possui listas com os equipamentos subscritos na rede de forma a, posteriormente, ser possível identificar dispositivos não autorizados e realizar funções como *Denial of Service*.

2.1.7.3 Operation Subsystem - OSS

Esta secção da rede está ligada a todo o equipamento existente na NSS e na RSS. É responsável pela operação e manutenção da rede, oferecendo uma perspectiva global da rede e permitindo assistência local na resolução de ocorrências. O OSS garante também operações administrativas e comerciais como o tratamento de subscrições de serviços, taxação dos serviços subscritos por cada utilizador, e operações, configuração e segurança da rede, bem como tarefas de manutenção da rede [6].

Por norma, a responsabilidade do funcionamento do OSS cai sobre as operadoras de telecomunicações, que conseguem também reproduzir relatórios estatísticos com os dados

do desempenho da rede, distribuição demográfica de utilizadores, serviços preferidos pelos clientes, entre outros.

Um dos módulos desta secção é o *Operation and Maintenance Center* (OMC), uma entidade funcional através da qual a operadora monitoriza e controla todos os elementos da rede. As suas funções incluem, por exemplo, actividades de relatório do tráfego na rede, o aprovisionamento de clientes e a recolha de registos de taxação dos serviços subscritos pelo utilizador.

Existe ainda o *Billing Center* (BC) que pode, em alternativa ao OMC, realizar a recolha dos registos de taxação dos serviços subscritos. O BC consegue ainda realizar o processamento dos registos de taxação para gerar facturas em função dos serviços usados por cada utilizador.

2.2 OpenBTS e Asterisk

2.2.1 OpenBTS

Open Base Transceiver Station (OpenBTS) é um *software* de licença livre que implementa as camadas inferiores da pilha de protocolos GSM, mais precisamente a camada física, a camada de *data link* e a camada de rede [4]. Pretende-se usar o OpenBTS como um ponto de acesso de GSM baseado em *software*.

Este *software* utiliza um *router SIP* ou um *Private Branch eXchange (PBX)* para executar funções de controlo de chamada/sessão, através do módulo de *software* aberto denominado *Asterisk*. O *Asterisk* permite que dois terminais ligados ao *router SIP* efectuem chamadas entre si, e que estes se conectem a outras redes, tanto à rede telefónica tradicional (PSTN) como também a redes de voz sobre o protocolo *Internet Protocol* (VoIP) [2].

O OpenBTS tem como principal objectivo permitir que terminais compatíveis com as normas de comunicações móveis, como o GSM e o UMTS, sejam usados como terminais SIP em redes VoIP.

Na figura 2.4 é possível observar os componentes da aplicação OpenBTS, sendo que os componentes com cantos rectos são componentes de *hardware*, e os componentes curvos são componentes de *software*.

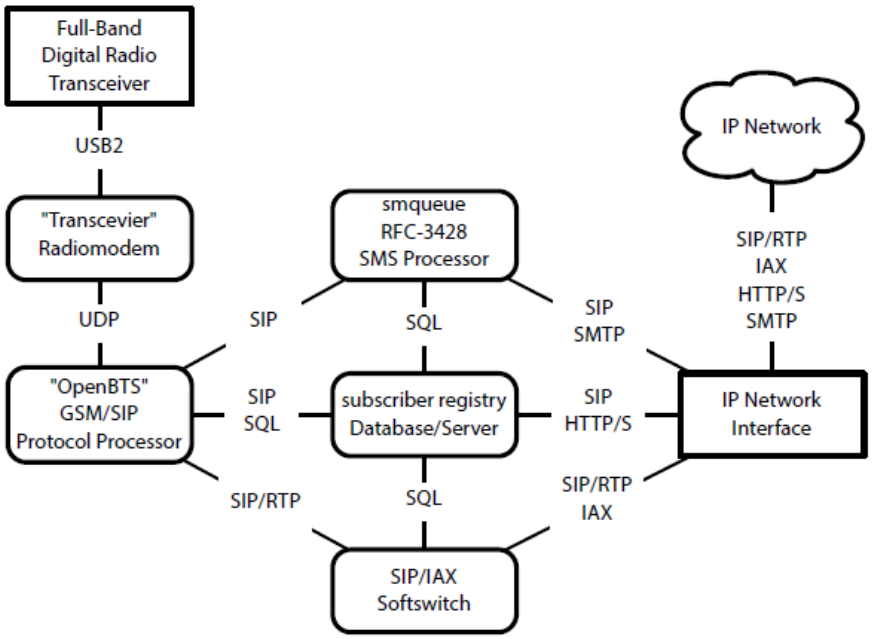

Figura 2.4: Componentes da aplicação OpenBTS, incluindo os protocolos de comunicação entre os diferentes blocos [4].

A tecnologia que torna o OpenBTS possível, do ponto de vista de *hardware*, são os rádios definidos por *Software* (SDRs). O SDR reduz e generaliza o *hardware* específico, e transfere para *software* parte das funções outrora suportadas por *hardware*.

O componente "Full-Band Digital Radio Transceiver" representa a plataforma de *hardware* e suporta as funções de rádio da BTS, que neste caso é consubstanciado na placa *Universal Software Radio Peripheral* (USRP). Ligado a este módulo está um módulo de *software* que funciona como um *modem* de rádio que controla o *hardware*, gere a ligação USB entre ambos e realiza funções de processamento de sinal [4].

O módulo "OpenBTS GSM/SIP Protocol Processor" realiza a maioria das funções da pilha de protocolos do GSM das camadas acima do rádio, incluindo:

- Funções de *Time Division Multiplexing*;
- Funções de *Forward Error Correction* ou codificação de canal, uma técnica utilizada para controlar erros na transmissão de dados sobre conexões com elevado ruído;
- Controlo de temporização e controlo de potência;
- Função de gestão de recursos de rádio;
- *Gateway* GSM/SIP para gestão da mobilidade, controlo de chamadas (sinalização) e mensagens de texto.

Este módulo comunica com três outros módulos que complementam as funções descritas. Os canais de comunicação SIP, descritos no diagrama, e ligados a este módulo, traduzem o fluxo de informação sobre a sessão e os respectivos utilizadores. Por outro lado, existem canais de comunicação RTP com o módulo "Softswitch", que corresponde ao módulo do *Asterisk*, para transporte de dados de áudio ou vídeo.

O módulo "SMQueue" é um processador de serviços de mensagens de texto, que funciona como um servidor que armazena e encaminha as mensagens no sistema OpenBTS. Tal como no *Short Message Service Center*, SMSC, normalizado pelo 3GPP para dar suporte às mensagens curtas, as mensagens são armazenadas numa lista de espera até que seja confirmada a entrega, ou que o sistema determine que a entrega é impossível, de forma semelhante ao que acontece num servidor de *email*. Os endereços suportados são endereços ISDN ou nomes de utilizador SIP.

O "Subscriber registry" é um módulo que verifica e autentica cada utilizador no sistema OpenBTS, com recurso a uma base de dados com as informações de utilizadores subscritos. Este módulo procura substituir tanto o registo SIP do *Asterisk* como o *Home Location Register* (HLR) numa rede convencional GSM.

O módulo "Softswitch" corresponde ao *router* SIP ou o PBX referido anteriormente, sendo que é implementado na aplicação *Asterisk*. O *software* OpenBTS usa este módulo para realizar funções de controlo de chamadas, que normalmente seriam da responsabilidade do MSC numa rede convencional de GSM.

Finalmente, restam a interface de ligação à *Internet* e os seus respectivos canais de comunicação. Esta interface será utilizada sempre que um utilizador deseje estabelecer uma sessão com outro utilizador numa rede diferente.

Entre os protocolos assinalados no diagrama nesta parte, é de destacar: o *Inter Asterisk eXchange* (IAX), que tem como propósito estabelecer a comunicação entre servidores *Asterisk*, e o *Simple Mail Transfer Protocol* (SMTP), que é o protocolo utilizado para enviar as mensagens curtas, SMS, como *emails*, a partir da *internet*.

O ponto chave nesta integração OpenBTS-SIP é que cada terminal GSM, em comunicação com a BTS, é visto pela rede VoIP como um terminal SIP. Fazendo o paralelismo com o cenário típico da rede GSM, o nome de utilizador SIP atribuído ao terminal (SIP URI) corresponderá ao identificador IMSI definido no cartão SIM. O endereço IP do utilizador SIP é o endereço IP que está atribuído à BTS. Nesta situação o OpenBTS permanece invisível à rede VoIP, funcionando apenas como um canal de comunicação entre terminais [4].

2.2.2 Asterisk

O *Asterisk* é uma implementação de *software* de licença livre que permite comunicação entre terminais de utilizadores, tais como telefones tradicionais (PSTN) ou terminais VoIP [2]. Como possui componentes de baixo nível e alto nível, permite transformar um

computador num servidor de comunicações de voz, de forma a simplificar o processo de desenvolvimento de aplicações IP.

É possível integrar o *Asterisk* com soluções multiprotocolo e de tempo real, como é o caso do *OpenBTS Release 4.0*. O OpenBTS adota a versão 11 do *Asterisk*, utilizando as suas funcionalidades de encaminhador (*SIP router*) [4].

O *Asterisk* abstrai as complexidades dos protocolos de comunicação e permite desenvolver soluções como sistemas telefónicos comerciais (IP PBXs), distribuidores de chamadas telefónicas, *VoIP gateways* ou *conference bridges* (salas de conferência virtuais para vários utilizadores comunicarem entre si) [2].

A arquitectura do *Asterisk* é baseada em módulos, situados no *core* do sistema. Cada módulo é um componente que pode ser configurado dinamicamente de modo a oferecer uma determinada funcionalidade. Este conceito permite maior flexibilidade na medida em que, cada utilizador escolhe quais os módulos que pretende utilizar para a sua solução, bem como a configuração que deseja para cada módulo.

Na figura 2.5 é possível visualizar o diagrama de um sistema *Asterisk*. A zona *Asterisk Components*, que corresponde ao *core* do *Asterisk*, interage com os módulos do sistema. A zona *Local OS* demonstra alguns serviços locais e a interacção com o *core*. A zona *Network* apresenta a integração com outras redes.

O *Asterisk* é composto por vários tipos de módulos e cada módulo tem funcionalidades diferenciadas. As configurações possíveis de cada módulo podem ser encontradas no manual de utilizador do *Asterisk*, e centram-se nos seguintes tópicos:

- *Channel Drivers*: comunicam com dispositivos externos ao *Asterisk* e traduzem a sinalização ou o protocolo em si à rede *core*;
- *Dialplan Applications*: oferecem ao sistema funcionalidades relativas a chamadas telefónicas. Uma aplicação pode oferecer funções simples como responder ou desligar uma chamada, ou comportamentos mais complexos como modo de espera, *voicemail* e funcionalidades de conferência;
- *Dialplan Functions*: estas funções podem ser usadas para recuperar, estabelecer ou manipular configurações das chamadas;
- CODECs (*COder/DECoder*): é um módulo que oferece codificação e descodificação de vários tipos de áudio e vídeo, que são diferentes entre dispositivos, e de forma a converter os dados nos diferentes formatos;
- *File Format Drivers*: são usadas para converter ficheiros multimédia para armazenamento interno num formato específico, ou fazer o processo inverso para partilhar os ficheiros na rede;
- *Call Detail Record Drivers*: ou *drivers* CDR, são usadas para armazenar registos de chamadas num disco ou base de dados;

- *Call Event Log Drivers*: ou *drivers* CEL, semelhantes às CDR, mas focam-se mais no que acontece no sistema *Asterisk* durante uma chamada;

- *Bridge Drivers*: oferecem vários métodos para criar ligações de forma a suportar chamadas multimédia entre utilizadores.

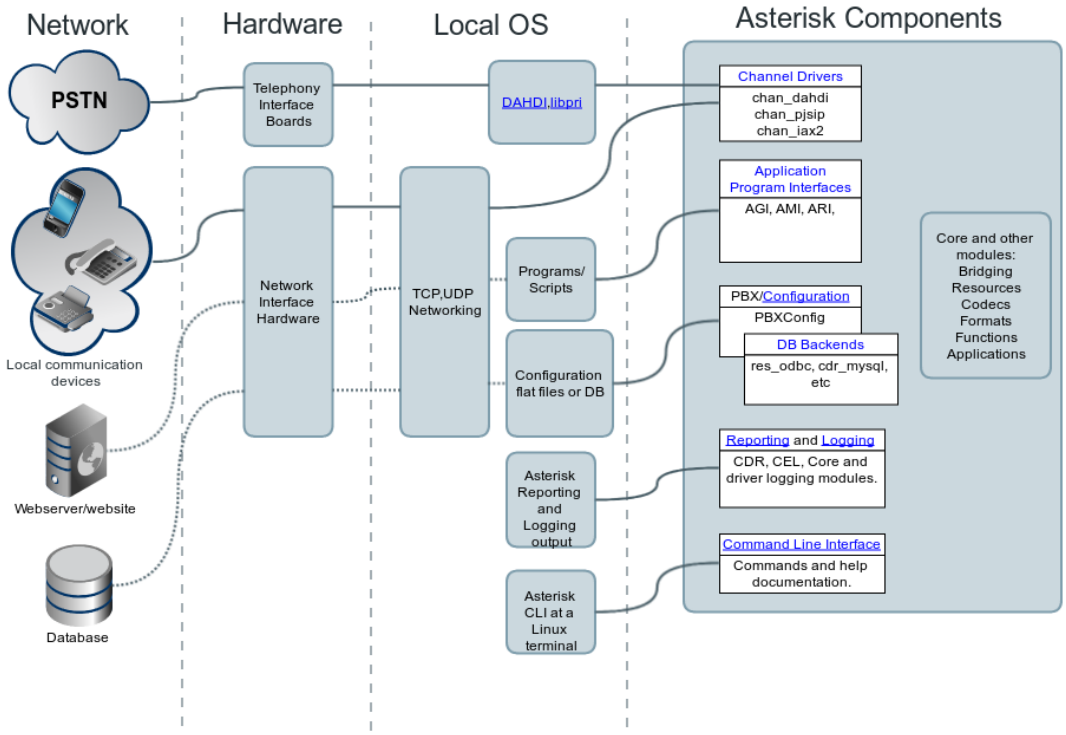

Figura 2.5: Diagrama de um sistema *Asterisk*. Este diagrama não é exaustivo, apenas demonstra algumas das relações mais comuns entre certos componentes [5].

Entre os módulos descritos no *core* falta especificar o *PBX core*, que constitui o componente essencial do sistema e que estabelece grande parte da estrutura do sistema. O *PBX core* está responsável por ler e compilar os ficheiros de configuração, especialmente do *dialplan*.

O *dialplan* consiste numa lista de instruções que o sistema deve usar para lidar com chamadas recebidas ou efectuadas por utilizadores do sistema, e define todo o comportamento do *Asterisk*. O *PBX core* tem ainda a função de carregar todos os outros módulos distintos referidos anteriormente, que oferecem todas as ferramentas adicionais para a solução em causa[2].

2.3 IP, SIP, RTP e RTCP

2.3.1 Internet Protocol

O objectivo da camada de rede é transportar pacotes da origem ao destino com o melhor esforço, mas sem garantias, e sem ter em conta se as máquinas estão na mesma rede ou em redes diferentes [9]. A *Internet* contém redundância nas suas conexões, com recurso a *backbones* e com os *Internet Service Providers* (ISPs) a possuírem múltiplas conexões entre si. Isto significa que existem vários caminhos possíveis entre duas máquinas que desejem comunicar entre si, sendo que cabe ao protocolo *Internet Protocol* (IP) decidir quais as ligações a usar.

O IP é a base que suporta o funcionamento de toda a *Internet*. É implementado na camada de rede e, ao contrário da maioria de outros protocolos desta camada, foi desenhado desde o início para suportar outros protocolos como, por exemplo, o TCP e UDP.

A filosofia deste protocolo e dos restantes da camada de rede é mantê-la o mais simples possível, tanto no seu funcionamento como nas suas decisões de encaminhamento de pacotes [9].

2.3.1.1 IPv4

O datagrama IP *version 4* consiste numa parte de cabeçalho e noutra parte opcional que pode conter os dados da aplicação ou outras informações (*payload*). O cabeçalho tem uma parte fixa de 20 *bytes* e uma parte opcional de tamanho variável, sendo que os *bits* são transmitidos da esquerda para a direita, de cima para baixo, com o *bit* mais alto do primeiro campo a ser enviado primeiro [9].

A figura seguinte demonstra o cabeçalho fixo do datagrama do protocolo IPv4:

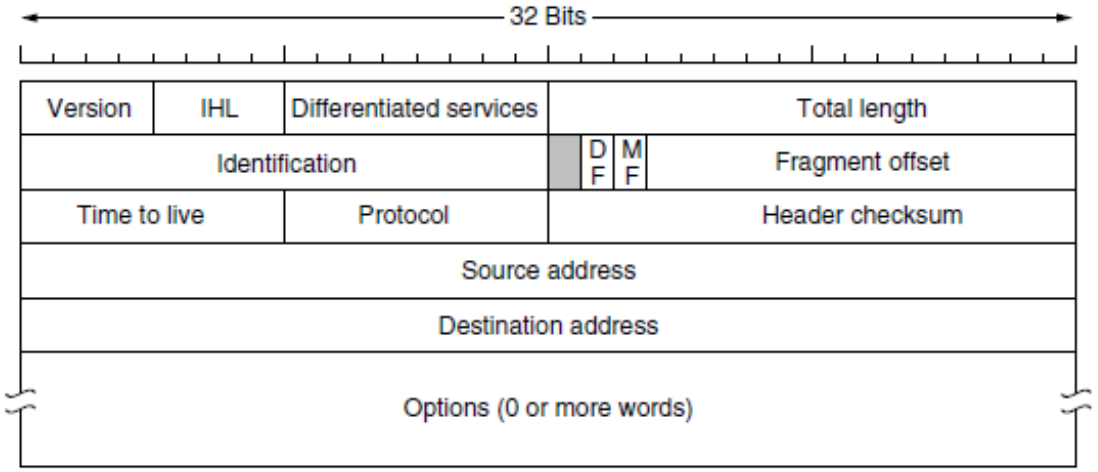

Figura 2.6: O cabeçalho do pacote IPv4 (*Internet Protocol*) [9].

O campo *Version* tem a informação da versão do IP a que o datagrama pertence. Ao incluir a versão em cada datagrama é possível realizar períodos de transição entre

versões do protocolo, mesmo que sejam longos períodos. Hoje em dia, devido à escassez de endereços IPv4, estamos num período de transição entre IPv4 e IPv6, a nova versão do protocolo.

O campo IHL existe para dar a conhecer o tamanho de todo o cabeçalho, em 32 *bits*, com um mínimo de 5 *bits* de tamanho. À semelhança deste campo, o *Total length* indica o tamanho de todo o datagrama, cabeçalho e *payload*, com um máximo de 65,535 *bytes*. Este limite pode vir a ser ultrapassado dada a necessidade de transmitir mais dados por pacote, no futuro.

O campo *Differentiated Services* é usado para marcar o pacote com a sua prioridade ou classe de serviços (primeiros 6 *bits*) e para guardar informação de congestão que o pacote possa ter sofrido na rede (2 *bits* restantes).

O campo *Identification* é necessário em situações em que a rede decide dividir o pacote em fragmentos, para facilitar a transmissão. Com esta informação o destino consegue determinar a que pacote pertence o fragmento que acabou de chegar, dado que todos os fragmentos terão o mesmo valor neste campo. Em seguida vem um *bit* que, por norma, não é usado, e assim pode ser utilizado para detetar tráfego malicioso.

DF é abreviatura para *Don't Fragment*, que existe para informar os *routers* para não fragmentar este pacote em qualquer circunstância. Caso não seja possível é devolvida uma mensagem de erro à origem.

Na mesma linha de raciocínio MF significa *More Fragments*, um campo adicionado para indicar que existem mais fragmentos do pacote que poderão ser recebidos. Quando um pacote é fragmentado, apenas o último fragmento não possui este campo preenchido, permitindo assim identificar o elemento final do pacote. Existe, ainda, o campo *Fragment* para indicar onde, no pacote actual, o dado fragmento pertence.

O campo TTL é um contador para indicar o tempo de vida de um pacote na rede. O valor máximo é 255. É suposto este valor ser decrementado uma vez em cada *hop*, para ser transmitido para outro *hop*.

Quando a camada de rede cria o pacote, precisa de saber o que fazer com o mesmo. O campo *Protocol* dá a conhecer qual o processo de transporte que o pacote deve realizar, sendo o TCP ou o UDP os protocolos de transporte mais utilizados.

O campo *Header Checksum* é usado para proteger o cabeçalho do pacote IPv4 contra a corrupção de dados do pacote. Este valor é calculado a cada *hop* pois existe, no mínimo, um valor que muda sempre, o TTL. Os campos S*ource Address* e *Destination Address* contêm os endereços IP das interfaces de rede da origem e do destino.

Finalmente o campo *Options* tem tamanho variável, e foi idealizado para conter informação de versões mais recentes, não incluídas na versão original do protocolo. A existência deste campo permite ainda explorar novas ideias para o protocolo e para transportar informação menos requisitada, evitando-se assim que sejam alocados *bits* para informação que é raramente necessária. Atualmente este campo entrou em desuso e é raramente utilizado.

Por outro lado, a parte de *payload* do pacote IP consiste nos dados que o pacote IP transporta. Esta parte tem dimensão variável, sendo que o limite é definido pelo protocolo de rede usado e, em alguns casos, pelos *routers* que formam a rede [9].

2.3.1.2 IPv6

Infelizmente o protocolo IP tornou-se vítima da sua popularidade e, mesmo com tecnologias como CIDR e NAT, os endereços começaram a escassear. Por esta e outras razões, surgiu a oportunidade de criar um novo protocolo que mantém os aspectos positivos do IPv4, melhora os aspectos negativos e adiciona novas funcionalidades que se revelaram necessárias com a utilização do IPv4. Assim, foi especificado o IPv6 [9].

A primeira lacuna que o IPv6 procura resolver é a escassez de endereços. O IPv6 tem endereços maiores, com 128 *bits*, que resulta numa amostra quase ilimitada de endereços e que certamente não irão escassear num futuro próximo.

Em segundo lugar está a simplificação do cabeçalho do pacote. O cabeçalho do IPv6 apenas contém sete campos, o que permite acelerar o processamento dos pacotes nos *routers* e, por consequência, aumentar a velocidade de transmissão e diminuir atrasos na rede.

A terceira funcionalidade que distingue o IPv6 é o suporte melhorado para as opções no cabeçalho. Tornou-se essencial melhorar este campo do IPv4 dado que vários campos que eram obrigatórios no cabeçalho desse protocolo tornaram-se opcionais. Além disso, o IPv6 representa as opções de maneira diferente, tornando possível aos *routers* ignorar as opções que é suposto processarem. Finalmente o IPv6 também apresenta melhorias na segurança e na qualidade de serviço (QoS).

Contudo, e apesar das semelhanças com o IPv4 e de ser também um protocolo da camada de rede, o IPv6 não é compatível com o IPv4. São necessários mecanismos de tradução de endereços para lidar com esta lacuna, como *gateways*, *tunneling* e a utilização de prefixos.

No entanto, o IPv6 integra-se e é compatível com outros protocolos de *internet*, incluindo TCP, UDP, ICMP, IGMP, OSPF, BGP e DNS. O protocolo SIP, que iremos descrever em seguida, também se inclui nos protocolos de *internet* compatíveis com o IPv6. A figura 2.7 demonstra o cabeçalho fixo do datagrama do protocolo IPv6.

O cabeçalho dos datagramas IPv6 herdaram alguns dos campos que existiam no IPv4. É o caso do campo *Version*, que neste protocolo é 6, o campo *Differentiated Services*, que distingue as classes de serviços dos datagramas e guarda informação da possível congestão sofrida pelo pacote na rede, e os campos *Source Address* e *Destination Address*, que indicam os endereços IP das interfaces de rede da origem e do destino.

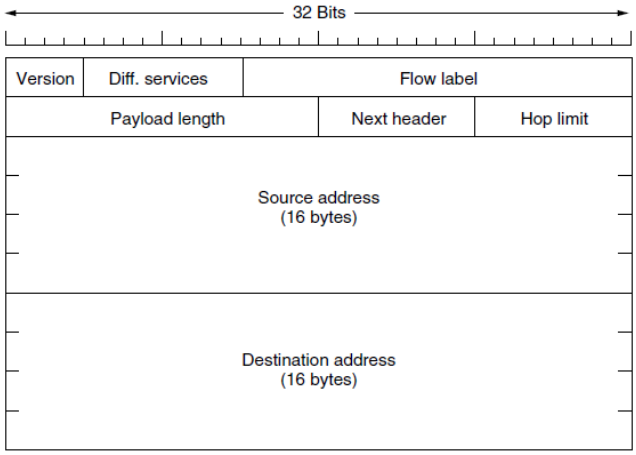

Figura 2.7: O cabeçalho do pacote IPv6 (*Internet Protocol*) [9].

O campo *Flow Label* possibilita que sejam marcados grupos de pacotes que têm o mesmo tipo de requisitos e que devem ser tratados da mesma maneira pela rede. Uma *Flow* com requisitos de qualidade de serviço é especificada com o endereço de origem, endereço de destino e com o número de *Flow*.

Em substituição do campo *Total Length* foi introduzido o campo *Payload Length*, dado que o significado do campo mudou: o espaço reservado para o cabeçalho já não é tomado em conta no tamanho total do pacote. Assim, o pacote IPv6 passa a ter apenas 65 515 *bytes*.

O campo *Next Header* foi uma das razões pela qual o cabeçalho pôde ser simplificado. O *Next Header* contém informação sobre qual dos cabeçalhos de extensão, que podem ser vários ou nenhum, estão associados ao pacote. Se o cabeçalho recebido pelo destino for o último, este campo conterá informação sobre qual o processo de transporte que deverá dar seguimento ao pacote.

Por fim, o campo *Hop Limit* serve para impedir que os pacotes permaneçam eternamente na rede. Na prática, este campo segue a mesma filosofia do campo *Time to live* no IPv4. Contudo, o valor neste campo mudou de segundos para número de *hops* dado que, para todos os efeitos, os *routers* nunca usavam este valor como segundos [9].

2.3.2 Session Initiation Protocol

Session Initiation Protocol (SIP) é um protocolo que descreve como estabelecer e configurar chamadas telefónicas, videoconferências e outras ligações de multimédia pela *Internet* [9].

Foi desenhado com o intuito de ser leve, simples e flexível, de forma a facilitar a integração com outras aplicações da *Internet* existentes. Por exemplo, o SIP define números de telefone como *Uniform Resource Locators* (URLs) de forma a que seja possível serem incluídos em páginas *web*. Assim, torna-se possível iniciar uma chamada telefónica através de um *browser* com um simples clique. Porém, estes números definidos por URLs usam a

designação SIP: sip:elsa@fct.unl.pt, para um utilizador chamado Elsa num terminal com o nome do domínio fct.unl.pt. Estes endereços SIP podem também conter endereços IPv4 ou IPv6, ou ainda os próprios números de telefone.

O SIP é um protocolo da camada de aplicação que pode ser executado sobre *User Datagram Protocol* (UDP) ou *Transmission Control Protocol* (TCP), conforme desejado. As suas funções consistem apenas em controlar o estabelecimento da sessão, gestão da sessão e encerramento da sessão. Para outras funções são usados outros protocolos, como o *Real-Time Transport Protocol* (RTP/RTCP) para transporte de dados.

Este protocolo suporta vários serviços como a localização do utilizador que inicia a chamada, a determinação das capacidades de comunicação do mesmo utilizador e o estabelecimento e fim da sessão. Outras funções, incluindo o modo de espera de chamadas, monotorização de chamadas, e métodos de encriptação e autenticação também são suportados.

O SIP possui métodos baseados nas *Multipurpose Internet Mail Extensions* (MIME) para facilitar a sua interacção com outras aplicações que usam IP. As MIME são incluídas nas mensagens trocadas entre utilizadores que pretendem iniciar uma sessão entre si. Os seis métodos definidos no protocolo estão listados na tabela seguinte:

Método	**Descrição**
INVITE	Pedido de estabelecimento de sessão
ACK	Confirmação de que a sessão foi estabelecida
BYE	Pedido de encerramento de sessão
OPTIONS	Pedido de informação das capacidades de comunicação de um utilizador
CANCEL	Cancelamento de um pedido pendente
REGISTER	Informar o servidor da localização actual do utilizador

Tabela 2.1: Métodos do protocolo SIP (adaptada) [9].

O estabelecimento de sessão depende do tipo de ligação. Numa ligação TCP, é primeiro criada a ligação TCP entre os utilizadores e é enviada uma mensagem *INVITE* pelo utilizador que deseja iniciar a sessão. Por outro lado, e caso se trate de uma ligação UDP, é enviada uma mensagem *INVITE* inserida num pacote UDP. Em ambos os casos os cabeçalhos na segunda linha, e seguintes, definem a estrutura da mensagem. Estes cabeçalhos incluem os requisitos de comunicação do utilizador que a envia[9].

Caso o utilizador aceite estabelecer uma sessão, a resposta inclui um código de 3 dígitos à semelhança dos que são usados no protocolo *Hypertext Transfer Protocol* (HTTP), que em caso de sucesso toma o valor 200. A ligação é estabelecida após ser enviada uma mensagem ACK a confirmar a recepção da mensagem 200.

Para terminar a sessão o utilizador apenas tem de enviar a mensagem *BYE* e, após receber confirmação ACK, a sessão é encerrada. O mesmo acontece para cancelar um pedido de início de sessão com a mensagem *CANCEL*.

O método *OPTIONS* é tipicamente usado antes do estabelecimento da sessão para confirmar se, de facto, a máquina utilizada pelo utilizador de destino suporta VoIP ou

outro tipo de sessão. Desta forma, é possível determinar se é viável continuar com o processo de estabelecimento de sessão ou, em caso de não ser viável, ser enviada uma mensagem de erro ao utilizador.

Por fim, o método *REGISTER* consiste na capacidade do protocolo SIP de se anunciar a um servidor SIP e utilizar os seus recursos. Esta mensagem é enviada ao servidor de localização SIP, que mantém informação actualizada sobre a localização dos utilizadores [9].

2.3.3 Real-Time Transport Protocol

O *Real-Time Transport Protocol* nasceu da necessidade de existir um protocolo de transporte genérico, independente das funcionalidades de qualquer aplicação, que oferece funções de transporte de dados. Embora seja considerado um protocolo de transporte, este é implementado na camada de aplicação [9].

A sua função principal é juntar vários fluxos de dados num único fluxo de pacotes UDP (*multiplexing*), que pode ser enviado a um único destino ou a vários destinos. Dado que este protocolo apenas funciona sobre UDP convencional, os seus pacotes não recebem tratamento especial nos *routers*, excepto se forem activadas medidas de qualidade de serviço nos mesmos *routers*.

Assim, não existem garantias de confirmação de recepção do pacote e não ocorrem retransmissões de pacotes. Os pacotes podem sofrer atrasos, podem ser corrompidos ou serem perdidos na rede.

O formato RTP contém várias funcionalidades de forma a ajudar o receptor a processar a informação multimédia que recebe. Cada pacote é enviado com um número de sequência para que o receptor possa determinar se existe algum pacote em falta na transmissão. Desta forma, o receptor pode resolver esta situação e saltar uma *frame* de vídeo, caso seja uma transmissão de vídeo, ou interpolar o valor do pacote em falta, caso se trate de uma transmissão de áudio.

Cada transmissão RTP pode conter várias amostras, e podem estar codificadas da forma que a aplicação em causa requere. Para facilitar a integração com a codificação de cada aplicação, o protocolo tem vários perfis definidos com vários formatos de codificação permitidos para cada perfil. Estas funcionalidades são especificadas no cabeçalho do pacote.

Apesar de oferecer a especificação da codificação, o RTP não interfere em como a codificação da informação é feita, sendo a aplicação em causa responsável por realizar esta função.

Outra funcionalidade que o RTP oferece, para responder às necessidades de aplicações em tempo real, é associar carimbos temporais à primeira amostra em cada pacote. Estes carimbos são sempre relativos ao início da transmissão, logo a diferença entre carimbos é a informação relevante nesta funcionalidade.

Com este mecanismo o destinatário pode realizar algum *buffering* e reproduzir a amostra no tempo certo após o início da transmissão, independentemente do momento que o pacote que contém a amostra chegou. Os carimbos temporais ajudam não só a reduzir os efeitos de atrasos na rede, como também permitem a sincronização de várias transmissões a ocorrerem ao mesmo tempo [9].

2.3.4 Real-Time Transport Control Protocol

Existe um protocolo associado ao RTP denominado *Real-Time Transport Control Protocol* (RTCP), cujo objectivo é gerir *feedback*, realizar funções de sincronização e lidar com a interface de utilizador, sem qualquer função de transporte de dados.

A gestão de *feedback* pode ser usada para difundir informação sobre atrasos e variação de atrasos (*jitter*) na rede, bem como informação sobre: largura de banda, congestão e outras informações disponibilizadas à origem da transmissão.

Estas informações do estado da rede podem ser usadas no processo de codificação da rede para aumentar a velocidade de transmissão quando a rede está a funcionar bem, ou reduzir a velocidade quando existem problemas na rede. Ao providenciar *feedback* contínuo, os algoritmos de codificação podem ser adaptados a todo o momento para oferecer a melhor qualidade de serviço, dadas as circunstâncias e o estado da rede.

Um problema que surge com este mecanismo é que esta informação do RTCP é sempre enviada para todos os utilizadores na rede. Para uma aplicação que difunde dados para um grupo de utilizadores (*multicast*), a largura de banda utilizada pelo RTCP iria crescer significativamente com o aumento de utilizadores no grupo.

Para prevenir que isto aconteça, os emissores que usam RTCP forçam a redução da utilização de largura de banda para um valor mínimo, de forma a continuar a difundir a informação de estado da rede sem consumir largura de banda instalada. Para realizar esta funcionalidade cada participante no grupo *multicast* precisa de conhecer a largura de banda estabelecida pelo RTCP (informação que é enviada pelo emissor) e o número de utilizadores do grupo, que pode ser estimado ao escutar a informação RTCP que é transmitida a todo o momento na rede.

O RTCP também oferece a sincronização entre fluxos de transmissão de dados, e oferece uma forma de identificação da origem da transmissão ao destinatário [9].

2.4 IMS

O *Internet Protocol* (IP) *Multimedia Subsystem* (IMS) foi desenvolvido com base num novo paradigma: a visão de integrar terminais móveis que comunicam sobre IP. Estes terminais têm câmaras integradas, ecrãs tácteis e recursos para executar várias aplicações, tornando-se em dispositivos que estão sempre ligados.

Este facto levou a que as aplicações fossem redefinidas, deixando de ser entidades isoladas que apenas comunicam com a interface de utilizador. A nova geração de aplicações

são entidades ponto a ponto que oferecem a partilha de multimédia entre utilizadores [7].

A capacidade de estabelecer ligações ponto a ponto, entre terminais móveis que comunicam sobre IP, é o ingrediente necessário para corresponder a este novo paradigma. Esta nova visão ultrapassa a arquitectura das redes convencionais.

As aplicações IP necessitam de um mecanismo que permita chegar ao utilizador de destino para comunicarem entre si. Como estamos a falar de terminais móveis, também tem que haver um mecanismo de acesso tendo em conta a mobilidade destes dispositivos. Precisa-se de um sistema global, uma rede IMS.

A rede IMS é uma rede que permite que as aplicações em dispositivos móveis estabeleçam ligações ponto a ponto, com vista à partilha de voz, vídeo, ficheiros e multimédia [7]. A capacidade de combinar a mobilidade dos terminais com uma rede IP é um factor fundamental para garantir serviços de multimédia ao cliente.

2.4.1 Requisitos da arquitectura

De forma a concretizar uma rede de serviços multimédia sobre IP é necessário rever os requisitos para uma arquitectura que suporte os motivos da sua criação, dado que as redes convencionais não conseguem corresponder a esta nova visão. De seguida, referem-se os requisitos mais importantes.

É fundamental o cliente ter conectividade IP para aceder aos serviços IMS. A conectividade IP pode ser obtida através da rede do seu operador, rede *home*, ou através da rede visitada, no caso de o cliente estar em *roaming* [7].

A figura 2.8 demonstra as duas opções descritas. O diagrama à esquerda, o terminal, em *roaming*, obteve um endereço IP através da rede visitada. Por outro lado, o diagrama à direita, o terminal em *roaming*, obteve o endereço IP através da rede *home*.

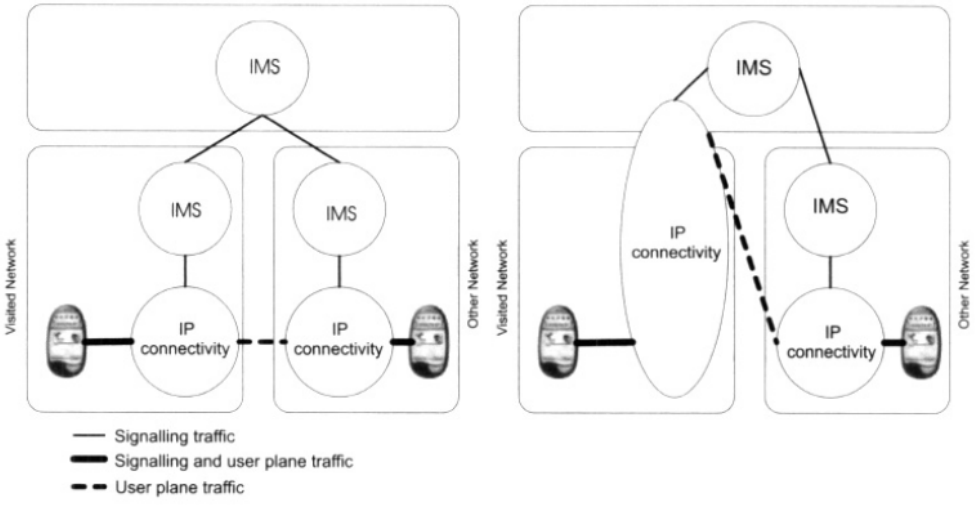

Figura 2.8: Opções de conectividade numa rede *home* e numa rede visitada [7].

Outro requisito necessário, é tornar o acesso independente à rede IMS, de forma a que os serviços IMS estejam disponíveis em qualquer rede com conectividade IP. Sendo que é necessário haver este tipo de acesso e conectividade IP, mesmo em redes visitadas, terá que existir suporte a cenários de *roaming*. Desta forma, e independentemente do acesso, o utilizador consegue ter conectividade IP sem limitações geográficas.

O modelo de *roaming* IMS defende uma configuração da rede em que a rede visitada oferece conectividade IP e acesso à rede, enquanto que o resto das funcionalidades e serviços IMS são suportados pela rede *home*. Esta cooperação entre a rede *home* e a rede visitada permite a optimização de recursos no plano do utilizador.

Uma preocupação que surge com serviços de multimédia sobre IP é garantir qualidade de serviço (QoS) aos seus clientes. Na *Internet*, alguns pacotes chegam atrasados em relação à transmissão e, outros, são mesmo descartados. No sistema IMS, o acesso independente e as redes de transporte, combinado com a arquitectura IMS, oferecem qualidade de serviço ponto a ponto.

Durante o estabelecimento de sessão SIP, o terminal do utilizador negoceia os requisitos de QoS e reserva os recursos necessários na rede de acesso. Em seguida, e durante toda a sessão, o terminal codifica cada tipo de multimédia diferente com o protocolo apropriado e envia os pacotes usando um protocolo de transporte, tipicamente TCP ou UDP, sobre IP.

Um outro requisito para o uso correcto dos recursos *media* é a aplicação de políticas de controlo sobre o IP, mediante a interacção entre a rede de acesso e o *core IMS*. Por outras palavras, é a capacidade de autorizar e controlar o tráfego de *media* IMS, com base na sinalização da sessão IMS (SIP).

A segurança é um requisito fundamental em qualquer sistema de telecomunicações. O sistema IMS oferece um nível de segurança semelhante ao do protocolo *General Packet Radio Service* (GPRS) e às redes de comutação de circuitos: só após o processo de autenticação é que o utilizador pode aceder aos serviços que subscreveu.

Do ponto de vista do operador de telecomunicações, a capacidade de taxar a utilização de serviços do utilizador é essencial. A arquitectura IMS permite vários modelos de taxação diferentes à escolha do *Internet Service Provider*,ISP, bem como a possibilidade de taxação *online* ou *offline* [7].

O ISP pode escolher taxar apenas o utilizador que estabeleceu a sessão ou taxar ambos os utilizadores presentes na sessão, com base na utilização dos recursos alocados ao nível de transporte. Contudo, o ISP também pode escolher correlacionar a informação de taxação gerada ao nível do transporte e do *core* do sistema IMS.

O sistema IMS tem o objectivo de gerir a rede de comunicações de uma forma normalizada e de convergência entre múltiplas tecnologias de acesso, com qualidade de serviços e fiável. No entanto, nem todos os utilizadores conseguem obter terminais que possibilitam o acesso à rede IMS de um dia para o outro. Logo, o requisito implícito nesta situação é o suporte de comunicação com outras redes, nomeadamente a rede de telefonia fixa (PSTN), a rede de telefonia móvel e a *Internet*.

Sendo o IMS um sistema de serviços de multimédia, terá que existir controlo sobre os serviços oferecidos, existindo dois modelos para o controlo de serviços: modelo da rede visitada (*roaming*) ou o modelo da rede *home*.

Na rede 2G convencional, o controlo de serviço é efectuado na rede visitada. Isto significa que, quando um utilizador está em *roaming*, existe uma entidade na rede visitada (*Mobile Switching Center Server* (MSC Server)), que oferece os serviços subscritos pelo utilizador na rede *home*, e controla o tráfego realizado. Porém, esta abordagem foi descartada devido à sua complexidade e dado que não trazia nenhum valor acrescido, comparado com o outro modelo.

O sistema IMS adota o modelo da rede *home*, em que a entidade que tem acesso aos serviços subscritos está sempre localizada na rede *home*. Esta entidade, na rede *home*, também é responsável por interagir com as plataformas de serviços do IMS [7].

Por fim, como é possível visualizar na figura 2.9, o sistema IMS assume uma arquitectura por camadas. Isto significa que, os serviços do utilizador e de transporte, estão separados da sinalização e dos serviços de gestão de sessão. O objectivo é garantir que as dependências entre camadas são mínimas, possibilitando a que seja possível, por exemplo, adicionar novas redes de acesso ao sistema ao longo do tempo [7].

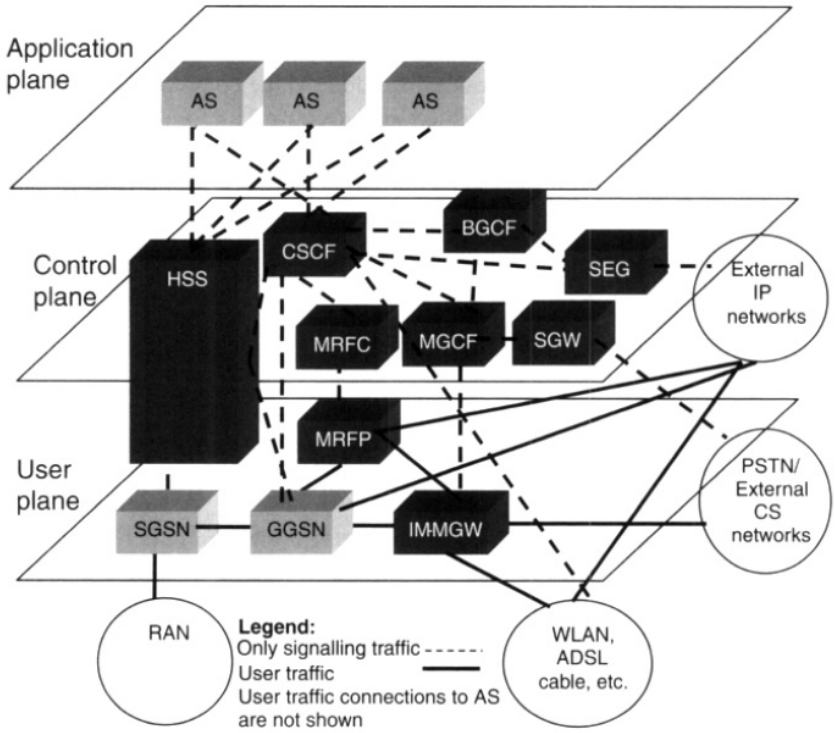

Figura 2.9: Arquitectura IMS por camadas [7].

As entidades da arquitectura IMS podem ser classificadas em seis categorias distintas, com diferentes módulos. Os módulos mais relevantes são os seguintes:

- Gestão da sessão e encaminhamento (CSCFs): P-CSCF, I-CSCF, S-CSCF, E-CSCF;

- Bases de dados: HSS, SLF;
- Serviços: AS, RFC e MRFP;
- Funções de *interworking*: BGCF, MGCF, IMS-MGW e SGW;
- Funções de suporte: PCFR, SEG, IBCF, TrGW e LRF;
- *Charging*;

2.4.2 Entidades da arquitectura

Gestão de sessão e encaminhamento

O *Proxy-Call Session Control Function* (P-CSCF) é o primeiro contacto dos utilizadores no sistema IMS, e o único ponto de acesso de entrada na rede. Todo o tráfego de sinalização SIP, entre o terminal de utilizador e a rede, passa por este módulo.

As suas funções incluem: compressão SIP entre o terminal e o P-CSCF (e vice-versa), segurança IPSec (túnel entre o terminal e o P-CSCF), detecção de sessões de emergência e interacção com o *Policy and Charging Rules Function* (PCRF). Neste último caso, é recebida informação do PCRF e o P-CSCF irá difundir informação de autorização IP QoS e regras de taxação à *gateway* de acesso (GGSN) [7].

O *Interrogating-Call Session Control Function* (I-CSCF) é o ponto de contacto para todas as ligações destinadas a um utilizador na rede do operador, podendo existir várias instâncias deste módulo nessa rede. É responsável por obter o nome do S-CSCF que está a servir o utilizador, ou atribuir um S-CSCF baseado na informação recebida do HSS.

Esta atribuição sucede quando o utilizador se regista na rede, ou quando este não se encontra registado, mas possui serviços subscritos (como por exemplo: *voice mail*). Posteriormente à atribuição e durante toda a sessão, o I-CSCF realiza o encaminhamento de informação SIP para o S-CSCF.

Em seguida está um dos módulos mais importantes do sistema IMS e que está localizado na rede *home*: o *Serving-Call Session Control Function* (S-CSCF). Realiza funções de controlo de sessão, bem como os processos de registo de utilizadores e ainda decisões de encaminhamento.

Enquanto o utilizador se encontra em sessão/chamada, o S-CSCF mantém o estado de sessão, interagindo com as plataformas de serviços e com os módulos de cobrança de serviços. Podem existir vários S-CSCF na rede do operador e cada um ter funcionalidades diferentes.

Por fim, o módulo *Emergency-Call Session Control Function* (E-CSCF) suporta os serviços de emergência, como os serviços das forças policiais e bombeiros. A principal tarefa deste módulo consiste em selecionar o centro de emergência onde o pedido será recebido, com base na localização do utilizador [7].

Bases de dados

O *Home Subscriber Server* (HSS), esquematizado na Figura 2.10, é a base de dados de utilizadores e dos dados relacionados com os serviços subscritos na rede IMS. A informação armazenada no HSS inclui identidades de utilizadores, informação de registo e serviços multimédia subscritos, condições de acesso e informação de *triggers* (eventos a originar por condições prévias).

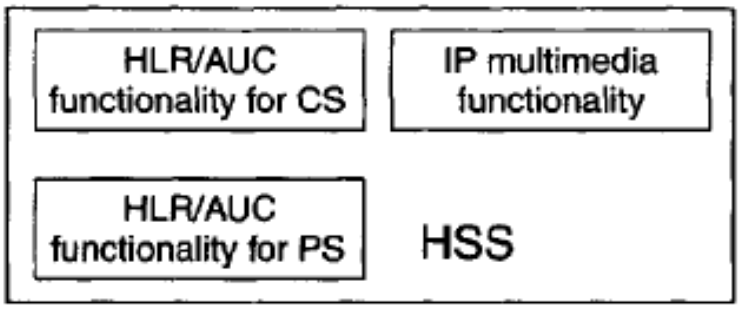

Figura 2.10: Estrutura do HSS [7].

Este módulo contém o *Home Location Register* (HLR) e o *Authentication Center* (AuC), de modo a permitir a integração do sistema IMS com os domínios de comutação de pacotes (PS) e comutação de circuitos (CS). O HLR providencia a localização de terminais para suportar as entidades do domínio PS, como SGSN e GGSN do GPRS, bem como as entidades do domínio CS, como o MSC do GSM.

Por outro lado, o AuC guarda uma chave secreta para cada utilizador, que é utilizada dinamicamente para gerar dados de segurança. Estes dados serão utilizados tanto na autenticação do utilizador, como na encriptação das comunicações entre o terminal móvel e a rede. Assim, é possível fornecer protecção da integridade de dados nas comunicações.

O *Subscription Locator Function* (SLF) é um módulo de suporte, cujo funcionamento é relevante quando existem vários HSSs implementados na rede do operador. Este módulo contém um mecanismo que permite ao I-CSCF, S-CSCF e ao AS encontrar o endereço do HSS que tem armazenado os dados de um dado utilizador [7].

Serviços

Os *Application Servers* (AS) não são apenas entidades puras relacionadas com o IMS. Na verdade, estes são executados sobre o IMS. As suas principais funções são:

- a possibilidade de processar e ter impacto numa sessão SIP originada na rede IMS;
- a capacidade de originar pedidos SIP;
- a capacidade de enviar informação de cobrança de serviço para as funções de *charging*;

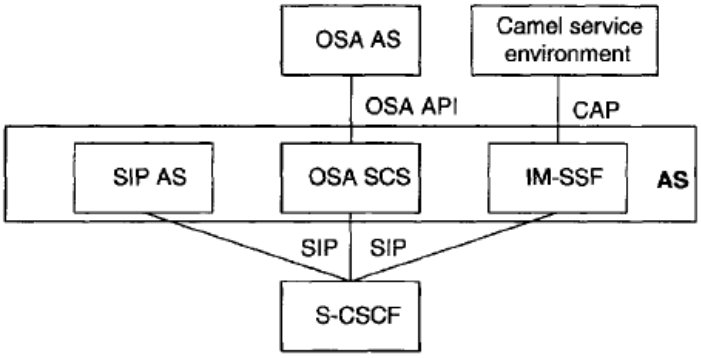

Figura 2.11: Tipos de *Application Servers*, AS, e as relações entre si [7].

O sistema IMS define ainda dois tipos de AS (esquematizados na figura 2.11), que oferecem suporte a serviços não baseados em SIP: *Open Service Architecture* (OSA) *Service Capability Server* (OSA-SCS) e *IP Multimedia Service Switching Function* (IM-SSF). Assim, cada AS pode estar dedicado a um único serviço, e o utilizador pode utilizar mais do que um AS em simultâneo. Por exemplo, utilizar um AS (AS1) que oferece um serviço de multimédia, e outro AS (AS2), que realiza o ajuste da aplicação ao visor do terminal do utilizador.

Existem ainda dois blocos que disponibilizam mecanismos relacionados com a adaptação dos ritmos de transmissão, mediante a exigência de cada serviço. O bloco *Multimedia Resource Function Controller* (MRFC) gere as comunicações SIP destinadas e enviadas para o S-CSCF, bem como as comunicações de controlo do MRFP [7].

Por fim, o *Multimedia Resource Function Processor* (MRFP) fornece recursos ao nível do plano de utilizador e que são geridos pelo MRFC. O seu comportamento resume-se ao seguinte:

- mistura de *streams* multimédia de chegada;
- fonte de *streams* multimédia (para anúncios multimédia);
- processamento dos *streams* de multimédia.

Funções de Interworking

Estas funções são necessárias para interligar redes diferentes e garantir o acesso independente aos serviços IMS. Tornam-se fundamentais para disponibilizar serviços de voz e vídeo entre a rede IMS e uma rede CS.

O *Breakout Gateway Control Function* (BGCF) é responsável pela escolha de uma rede, por onde enviar um pedido SIP destinado a uma rede de CS. O S-CSCF necessita de enviar um pedido SIP e o BGCF pode optar, por escolher a mesma rede onde o BGCF se encontra (escolhendo um MGCF), ou por escolher uma rede externa, reenviando a sessão para outro BGCF [7].

Quando o BGCF escolhe a rede, o *Media Gateway Control Function* (MGCF) permite a comunicação entre a rede IMS e a rede CS, e realiza conversão de protocolos entre as duas redes. Quando o pedido SIP é recebido pelo MGCF, este realiza uma conversão entre o protocolo SIP e a sinalização *ISDN User Part* (ISUP). Por fim, envia o pedido à *Signaling Gateway* (SGW), que realiza a conversão da sinalização nos dois sentidos, ao nível do transporte.

Finalmente, o bloco *IMS-Media Gateway* (IMS-MGW) fornece a interligação do plano do utilizador entre a rede IMS e a rede de CS. A sinalização de uma chamada CS, originada fora da rede IMS e destinada a um utilizador IMS, é encaminhada para o MGCF. Como mencionado anteriormente, o MGCF realiza a conversão de protocolos e envia o pedido de sessão SIP ao I-CSCF, para encerrar a sessão/chamada.

Funções de suporte

O módulo *Policy and Charging Rules Function* (PCRF) é responsável pelo controlo e supervisão das decisões de cobrança de serviços, com base na informação recolhida do P-CSCF. No caso de o operador utilizar este módulo, cabe ao PCRF gerar as regras de cobrança de serviços bem como a autorização de fluxos IP, de acordo com os meios de comunicação utilizados na negociação de *QoS Session Description Protocol* (QoS SDP) [7].

Além disso, o PCRF poderá ainda receber relatórios de eventos no plano de transporte como, por exemplo, quebras de ligação com o destinatário. Estes relatórios serão enviados ao P-CSCF, permitindo a libertação de todos os recursos associados à sessão IMS.

O módulo *Interconnection Border Control Function* (IBCF) disponibiliza funções específicas para a interligação de dois domínios de operadores diferentes:

- possibilita comunicações entre aplicações IMS IPv4 e IPv6;
- funções de ocultação da topologia da rede;
- funções de controlo do plano de transporte;
- examina a informação de sinalização SIP e seleciona o procedimento de sinalização;
- gera registos de taxação;

Do mesmo modo que um *Application Level Gateway* (ALG), modifica a informação SIP e SDP, de forma a que as diferentes versões IPv4 e IPv6 possam comunicar entre si.

O bloco *Security Gateway* (SEG) protege o tráfego do plano de controlo entre diferentes domínios de segurança em que cada domínio é gerido por uma única autoridade administrativa. Localiza-se na fronteira do domínio de segurança da rede, oferecendo um reforço das políticas de segurança a outros SEGs, pertencentes ao domínio do destino. No sistema IMS, todo o tráfego no interior da arquitectura é encaminhado para os blocos SEGs.

Por fim, o bloco *Location Retrieval Function* (LRF) assiste o bloco E-CSCF, fornecendo informação da localização dos utilizadores que iniciam chamadas de emergência [7]. Para

esta funcionalidade, o LRF pode conter um servidor de localizações ou uma interface para um servidor de localizações externo. Além disso, o LRF ainda poderá fazer mapeamento da localização do utilizador através da função *Routing Determination Function* (RDF).

Capítulo 3

Arquitectura GSM-SIP

Neste capítulo pretende-se apresentar parte da arquitectura desenhada para o protótipo experimental, nomeadamente a integração do nível de rádio GSM com o protocolo SIP. Esta integração é fundamental dado que irá permitir, posteriormente, a integração da rede GSM com uma rede *core IMS*. Como tal, será dada especial atenção ao OpenBTS e à forma como ocorrem as operações de registo e início de sessão entre dois terminais GSM.

Assim sendo, o capítulo divide-se em quatro tópicos:

- Descrição do cenário experimental;
- Descrição do cenário OpenBTS em detalhe, com especial atenção ao tráfego GSM e SIP;
- Registo de um terminal móvel GSM na rede OpenBTS;
- Interligação de dois terminais GSM, incluindo a sinalização trocada entre terminais.

3.1 Cenário Experimental

Como referido anteriormente, esta secção da arquitectura procura fazer a integração do nível rádio GSM com o protocolo SIP. Ou seja, o objectivo é fazer corresponder a cada terminal ligado à estação base (BTS, nó rectangular amarelo na figura 3.1) um SIP URI, que não é mais do que um endereço SIP que permite identificar um utilizador numa rede IP, como é o caso de uma rede IMS. O cenário experimental da arquitectura GSM-SIP está ilustrado na figura 3.1.

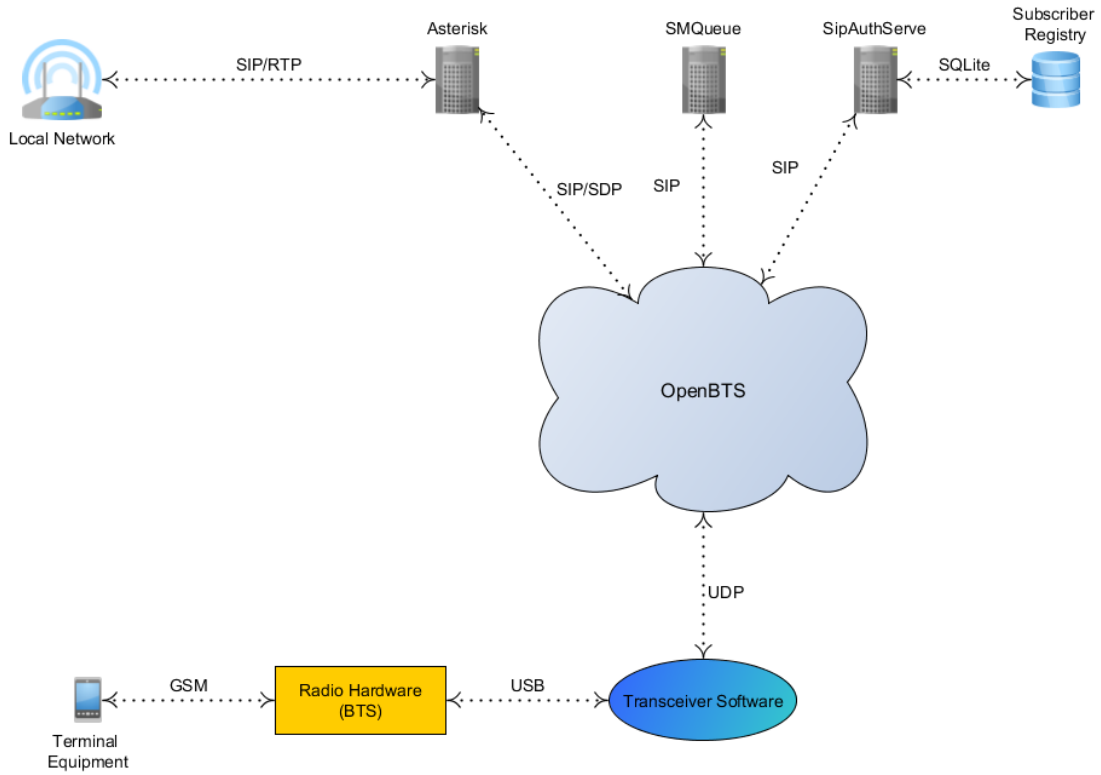

Figura 3.1: Representação do cenário experimental da arquitectura GSM-SIP.

Assim, em termos de *hardware*, temos uma estação base (Ettus USRP B100) a receber ligações de terminais GSM, que por sua vez se encontra conectada, via USB, a uma máquina com o *software* que controla a BTS (Intel Core i5-4460 CPU @ 3.20 GHz x 4, 8 GB RAM). Esta máquina encontra-se ligada em rede com uma rede local, que por sua vez irá permitir comunicar com o *core IMS*.

A rede local é suportada por um *Access Point* (na figura 3.1 corresponde ao módulo *Local Network*). A máquina OpenBTS, que controla a BTS, tem uma ligação *Ethernet* a este AP, tal como a máquina em que está a ser lançado o *core IMS*. Este módulo tem a função de permitir o acesso externo ao sistema OpenBTS, ou seja, irá permitir que os terminais OpenBTS possam comunicar com terminais fora do sistema, através do protocolo SIP.

Em termos de *software*, a responsabilidade é assumida pelo OpenBTS, que controla e gere todos os outros módulos de *software*. O módulo *Transceiver Software* funciona como um *modem* de rádio que controla o *hardware*, gere a ligação USB entre ambos e realiza funções de processamento de sinal.

Todos os outros módulos operam em torno das comunicações SIP e da conversão GSM-SIP. O módulo *Asterisk* é o ponto de entrada e saída de todas as mensagens SIP que envolvem os utilizadores OpenBTS. Este módulo contém toda a configuração necessária para gerir o fluxo de comunicações SIP, incluindo o nível de segurança e autenticação entre utilizadores, os *codecs* permitidos e proibidos para as comunicações, entre outras.

O módulo *SIPAuthserve* é um servidor de autenticação que processa as mensagens *SIP*

INVITE, ou seja, todos os pedidos de estabelecimento de sessão que chegam ao *Asterisk*. Este servidor também controla o acesso à base de dados *Subscriber Registry* que contém as informações de todos os terminais registados, possibilitando a resolução de um MSISDN num IMSI, e por sua vez encontrar o SIP URI correspondente.

Por sua vez, o módulo *SMQueue* processa todas SMS trocadas entre terminais registados no OpenBTS e todas as *SIP MESSAGES* recebidas pelo *Asterisk*. A conversão de *SIP MESSAGES* em SMS será uma operação importante para os serviços experimentais que irão ser abordados no capítulo cinco deste documento.

3.2 Configuração do OpenBTS/Asterisk

Neste protótipo a configuração do OpenBTS força a que toda a comunicação SIP seja recebida directamente pelo *Asterisk*. Assim, todas as mensagens SIP vindas de redes externas são recebidas por este módulo, que depois decide como processar e encaminhar as mensagens SIP recebidas para os outros módulos.

O *Asterisk* permite configurações diferentes para sessões que são originadas no OpenBTS para destinos externos à rede, e para sessões cujos destinatários são utilizadores registados na rede OpenBTS. É possível verificar a configuração do *Asterisk* no anexo deste documento.

A configuração implementada permite que cada terminal registado no OpenBTS tenha um SIP URI associado baseado no seu IMSI e no IP do OpenBTS (no formato: IMSI@<OpenBTSIPaddress>). O objectivo é que cada terminal seja reconhecido e detectado como um utilizador SIP numa rede IMS.

Para tal, foi configurada uma extensão (ex: 100) que, na perspectiva de um utilizador com um terminal móvel, é o número identificador do cliente IMS (ex: Alice) na rede OpenBTS. Ou seja, qualquer utilizador na rede OpenBTS que deseje contactar a Alice tem de digitar no terminal a extensão 100. Esta extensão faz correspondência com o SIP URI da Alice, e é enviado um pedido de estabelecimento de sessão para a Alice.

Para sessões originadas pelo OpenBTS, o *Asterisk* constrói e envia a mensagem *SIP INVITE* para a rede local com o SIP URI do utilizador de destino. Neste cenário, a mensagem irá ser difundida na rede IMS, e é localizado o utilizador de destino. Caso este exista, encaminha o pedido de sessão para esse utilizador. Se não existir, a mensagem é descartada.

A configuração escolhida para pedidos de sessões externas, ou seja, com destino a utilizadores OpenBTS, permite que todos os clientes registados na rede IMS, que pertençam a um certo domínio (neste caso open-ims.test), possam estabelecer sessões com utilizadores registados no OpenBTS, desde que conheçam o seu SIP URI. Os SIP URIs dos utilizadores OpenBTS foram adicionados manualmente aos contactos dos clientes IMS. Com este SIP URI, serão descobertos o IMSI e o MSISDN do terminal de destino, caso exista e esteja activo na rede OpenBTS, e o pedido de sessão será encaminhado para o respectivo telemóvel.

Por fim, também é necessário configurar as comunicações internas, ou seja, sessões entre terminais registados no OpenBTS. Para os terminais comunicarem entre si, são usados os MSISDNs associados a cada terminal. O OpenBTS gera uma mensagem *SIP INVITE* e cria um SIP URI baseado no MSISDN de destino e destinado ao *Asterisk* (MSISDN@<OpenBTSIPaddress>:<AsteriskPort>).

A configuração do *Asterisk* estabelece a correspondência MSISDN-IMSI de forma a encaminhar o pedido de sessão para o terminal de destino. O *Asterisk* encaminha o pedido de sessão para o terminal (novamente através do OpenBTS) mas cria um novo SIP URI, desta vez baseado no seu IMSI (IMSI@<OpenBTSIPaddress>:<OpenBTSPort>). Desta forma, o OpenBTS recebe o pedido e toma conhecimento do terminal de destino através do IMSI inserido no SIP URI.

3.3 Registo GSM

A operação de registo GSM corresponde à autenticação do terminal móvel na rede. De modo a poder comunicar com outros terminais, sejam da rede móvel ou da rede fixa, e estejam onde estiverem, o terminal tem que confirmar à rede que é quem diz ser para o utilizador poder usufruir dos serviços que subscreveu. O processo de autenticação de um terminal GSM é ilustrado na figura 3.2.

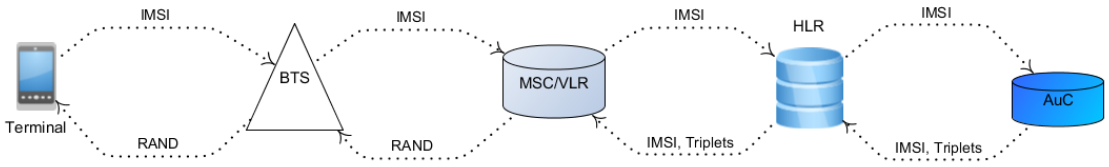

Figura 3.2: Processo de autenticação de um terminal GSM na rede do seu operador.

O terminal envia um pedido de acesso à rede à BTS com o seu IMSI. O pedido é encaminhado para o MSC/VLR. O MSC encaminha o pedido para o HLR e pede que o HLR envie os *authentication triplets*.

Quando recebe o pedido de acesso e o IMSI do terminal, o HLR vai verificar o IMSI na sua base de dados para confirmar que o IMSI é válido e pertence à rede. Assim que o IMSI é validado, o pedido de acesso e o IMSI são encaminhados para o AuC (*Authentication Center*).

O AuC vai usar o IMSI para procurar a *Individual Subscriber Authentication Key*, Ki. A ki é um número de 128 *bits* que apenas existe no AuC e no cartão SIM do terminal. O AuC irá também gerar um número aleatório de 128 *bits* denominado RAND. Estes parâmetros irão ser usados para gerar os *authentication triplets*, como é sugerido na figura 3.3.

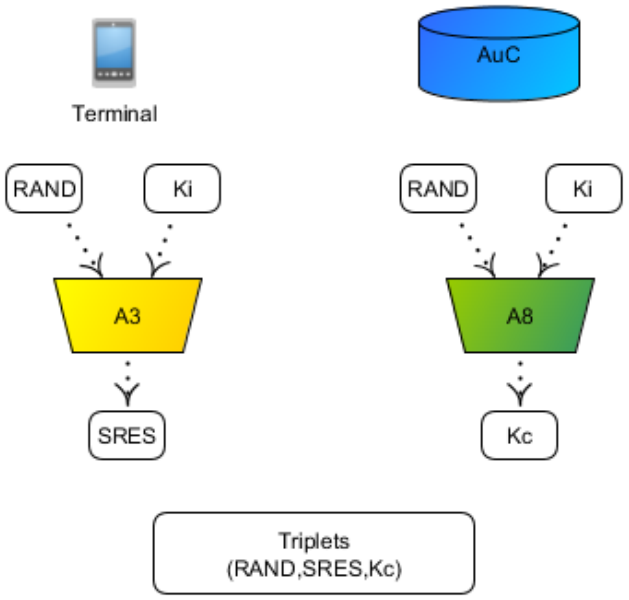

Figura 3.3: Algoritmos que geram os parâmetros de autenticação de um terminal numa rede GSM.

O RAND e a Ki são os parâmetros de entrada do algoritmo A3, cujo resultado é um número de 32 *bits* denominado *Signed Response* ou SRES. No entanto, o RAND e o Ki também são os parâmetros de entrada do algoritmo de geração da chave secreta de encritptação, Kc. Esta chave secreta é usada no algoritmo A5, um algoritmo de encriptação e desencriptação dos dados transmitidos entre o terminal e a rede.

Assim, o SRES, o RAND e a Kc formam o *triplet* de autenticação. Este *triplet* é único para um IMSI e não pode ser usado em nenhum outro IMSI. O AuC pode gerar vários triplets a enviar para o MSC/VLR de modo a evitar pedidos constantes ao HLR para autenticar o terminal de novo. Tendo gerado o *triplet* de autenticação, o AuC encaminha o *triplet* de volta para o HLR. Posteriormente, o HLR encaminha o *triplet* para o MSC/VLR.

O MSC guarda a Kc e o SRES no VLR, mas encaminha o RAND para o terminal (através da BTS que o está a servir) e pede que o terminal se autentique perante a rede. O terminal tem o Ki guardado cartão SIM. Também estão presentes neste cartão os algoritmos A3 e A8. Desta forma, o RAND e a Ki são introduzidos nos algoritmos A3 e A8 para gerar o SRES e a Kc, respectivamente.

A partir deste ponto todas a comunicações entre o terminal e a rede são encriptadas com o algoritmo A5, e ambas as partes conseguem desencriptar e autenticar as comunicações com o mesmo algoritmo. O processo de autenticação fica assim concluído.

3.3.1 Sinalização SIP

A funcionalidade principal do OpenBTS é tornar o terminal móvel num terminal SIP. Assim, é necessário analisar o processo de registo e autenticação de um terminal do ponto

de vista do OpenBTS, e com especial foco no protocolo SIP. No anexo deste documento é possível encontrar uma captura *wireshark* do processo de registo de um terminal GSM na rede OpenBTS.

No entanto, cada terminal é registado manualmente na base de dados através da linha de comandos (no caso desta implementação usou-se o *linux*, distribuição *Ubuntu*). O comando de registo pede um nome para o terminal, o IMSI do terminal (obtido através da consola do OpenBTS, OpenBTSCLI), um MSISDN que deverá ser único na rede, e uma chave secreta Ki (nº de 128 *bits*, opcional).

Assim escolheu-se para o terminal 1:

- Nome - "Terminal1";
- IMSI - 268032103518121 ;
- MSISDN - "3334445551";
- Ki - vazio ,

e para o terminal 2:

- Nome - "Terminal2";
- IMSI - 268069632902597;
- MSISDN - "3334445552";
- Ki - vazio .

Como se pode verificar, optou-se por deixar a chave secreta Ki sem nenhum valor. Assim, se a base de dados não tem a chave secreta Ki para um dado terminal, o sistema OpenBTS não faz autenticação. Desta forma, é possível registar manualmente qualquer terminal GSM nesta rede.

Um terminal registado consegue comunicar com outros terminais registados e activos nesta rede e com outros utilizadores registados noutras redes, mediante o uso do seu SIP URI.

A configuração pré-definida do OpenBTS aloja a aplicação OpenBTS no porto 5062 e a aplicação *SIPAuthServe* no porto 5064, ambos com o endereço IP 192.168.0.100 estático. A sinalização correspondente à operação de registo está demonstrada na figura 3.4.

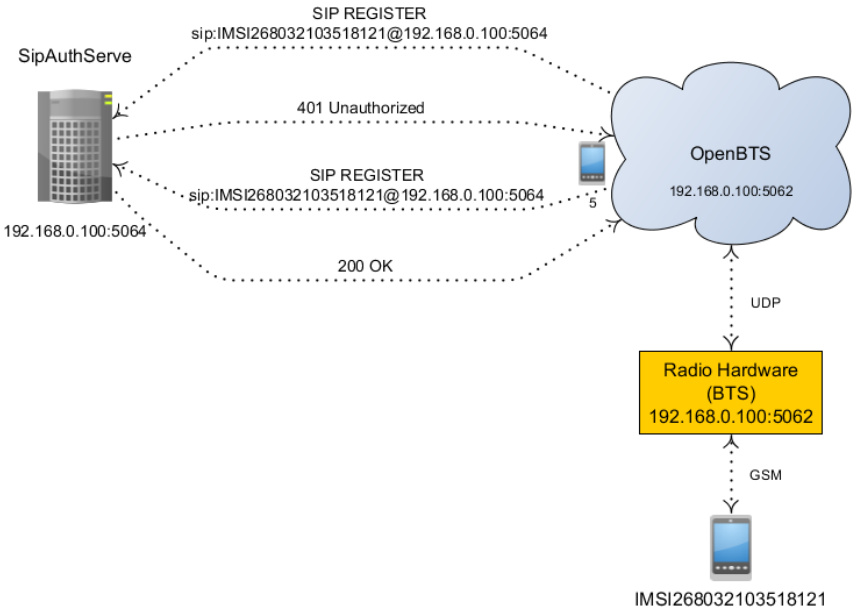

Figura 3.4: Representação da sinalização trocada na operação de registo de um terminal.

O processo de registo é gerido pelo *SIP Authorization Server*, *SIPAuthServe*, que recebe e processa pedidos *SIP REGISTER*. Estes pedidos são gerados pelo OpenBTS, e enviados ao *SIPAuthServe*, quando um terminal móvel se tenta registar.

O OpenBTS começa por gerar um pedido *SIP REGISTER* e enviar ao *SIPAuthServe*, que contém o IMSI do terminal que se pretende autenticar. Como se trata de uma mensagem *SIP REGISTER* terá que ter um SIP URI, como se tratasse do registo de um terminal SIP. Assim o OpenBTS gera um SIP URI com o IMSI do terminal no formato: IMSI@<OpenBTSIPaddress>.

O *SIPAuthServe* responde com um *401 Unauthorized*, que informa o esquema de autenticação necessário para concluir o processo. Com esta informação o OpenBTS pode agora construir um novo pedido *SIP REGISTER* com o esquema e parâmetros necessários para a autenticação do terminal com sucesso.

Assim, o OpenBTS responde novamente com um *SIP REGISTER*. No caso de sucesso, o *SIPAuthServe* responde com um *200 OK* e conclui a autenticação do terminal na rede.

3.4 Interligação entre dois terminais GSM

O processo de interligação entre dois terminais GSM corresponde ao início de uma sessão GSM entre os dois terminais. Estes dois terminais podem estar em zonas geográficas distantes, e até podem mesmo pertencer a dois operadores diferentes. Na figura 3.5 está demonstrado um caso do início de uma sessão GSM-GSM em que os terminais A e B estão em zonas geográficas distantes e estão registados em HLRs diferentes.

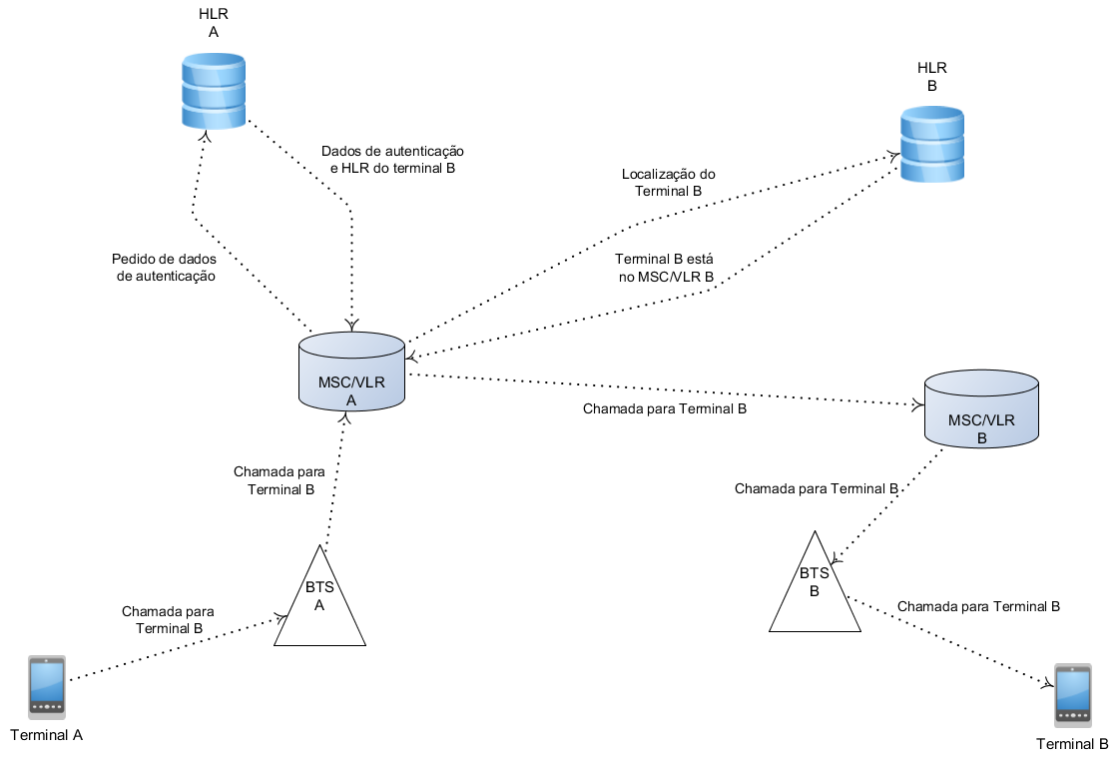

Figura 3.5: Diagrama que ilustra o processo de estabelecimento de chamada entre dois terminais GSM.

O utilizador introduz no terminal, terminal A, o número de utilizador (MSISDN) com quem pretende iniciar uma chamada, e clica no botão de iniciar chamada. O terminal A envia um pedido de sessão com o MSISDN de destino para a BTS que o está a servir. Assim que recebe este pedido e verifica que o terminal não está registado nesta BTS, encaminha o pedido para o seu MSC/VLR, MSC/VLR A.

O MSC/VLR A contacta o seu HLR, HLR A, de forma a obter a informação sobre o utilizador de destino (terminal B), caso nunca o tenha contactado antes e já tenha essa informação na sua posse. O HLR A verifica que o terminal B não pertence à sua rede, e devolve ao MSC/VLR A a localização do HLR a que o terminal pertence e os dados de autenticação do terminal de destino.

O MSC/VLR A interroga o segundo HLR, HLR B, para pedir a localização do terminal de destino. O HLR B responde com a localização do MSC/VLR que está a servir o terminal B. O MSC/VLR A envia o pedido de sessão para o segundo MSC/VLR, MSC/VLR B, que está a servir o terminal B.

O MSC/VLR B encaminha o pedido para o terminal B, através da BTS na qual este utilizador está registado. Assim que recebe o pedido, o terminal B toca e o utilizador aceita a chamada. O terminal B envia sinalização de chamada aceite para o terminal A, e a chamada fica então estabelecida.

3.4.1 Sinalização SIP

Na rede do OpenBTS, o estabelecimento de ligação entre dois terminais GSM é diferente. O OpenBTS emula as funções do MSC/VLR e do HLR através dos módulos *Asterisk* e do *Subscriber Registry*, respectivamente. Assim, o estabelecimento de ligação resume-se à interação entre estes módulos, com recurso ao protocolo SIP para a comunicação entre estas entidades.

Todo o processo do estabelecimento de sessão entre o terminal 2 e o terminal 1 está ilustrado na figura 3.6. No anexo deste documento é possível encontrar a captura *wireshark* do estabelecimento de sessão de dois terminais GSM registados na rede OpenBTS.

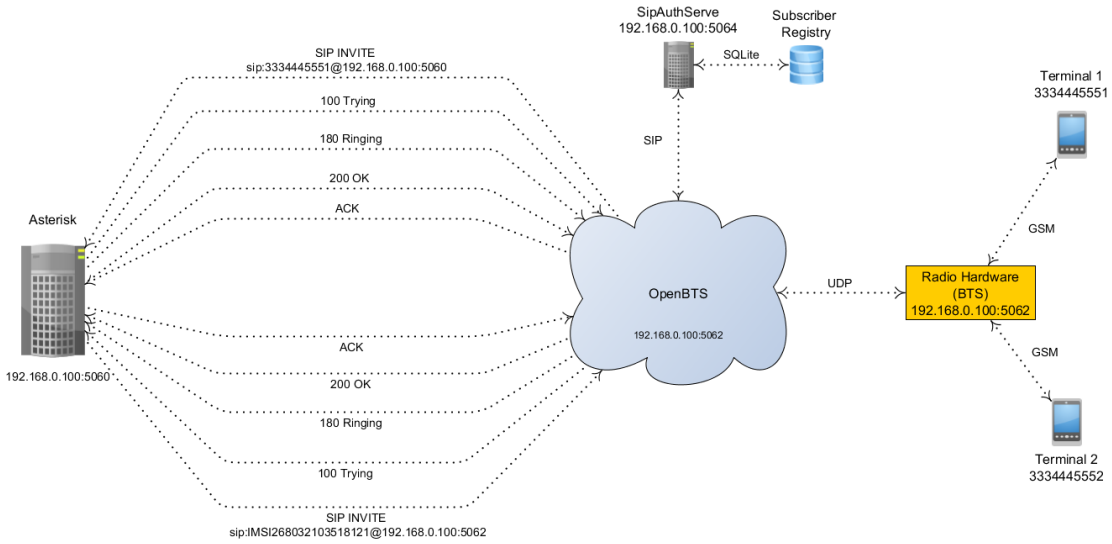

Figura 3.6: Representação da sinalização trocada no início de sessão entre dois terminais GSM.

Ambos os terminais estão registados com o seu IMSI no *Subscriber Registry*, e ambos têm um MSISDN atribuído. Para iniciar chamada o utilizador introduz no terminal 2 o MSISDN do terminal 1 e clica no botão de iniciar chamada. O OpenBTS recebe o pedido de sessão e confirma com o *Subscriber Registry*, através do *SIPAuthServe*, que o MSISDN do terminal 1 é válido e pertence à rede.

Após esta verificação, o OpenBTS gera um *SIP INVITE* com um SIP URI de destino que será baseado no MSISDN do terminal 1 (MSISDN@<OpenBTSIPaddress>:<AsteriskPort>). O *SIP INVITE* é encaminhado para o *Asterisk*. O *Asterisk* recebe o pedido e envia uma resposta *100 Trying*, para o informar que está a tentar encontrar o utilizador de destino.

A configuração do *dialplan* do *Asterisk* permite-lhe verificar que, no SIP URI inserido no *SIP INVITE*, está o MSISDN do terminal 1. Na sua configuração, está disponível a correspondência IMSI-MSISDN para todos os terminais registados na rede OpenBTS.

Assim, o *Asterisk* irá gerar um novo *SIP INVITE* com um novo SIP URI, desta vez baseado no IMSI do terminal 1 (no formato: IMSI@<OpenBTSIPaddress>:<OpenBTSPort>),

e envia este novo *SIP INVITE* de volta para o OpenBTS. O OpenBTS recebe o novo *SIP INVITE* e devolve um aviso *100 Trying*.

Com este novo *SIP INVITE* o OpenBTS consegue agora saber o IMSI do terminal 1 e envia o pedido de chamada para o terminal 1. O terminal 1 notifica o OpenBTS que o terminal está a tocar e à espera que o utilizador atenda a chamada. O OpenBTS gera um aviso *180 Ringing* e encaminha este aviso para o *Asterisk*.

Quando o utilizador do terminal 1 aceitar a chamada o OpenBTS recebe uma notificação. Posteriormente, o OpenBTS gera e envia ao *Asterisk* uma mensagem *200 OK* que informa que o utilizador atendeu a chamada. O *Asterisk* responde com uma mensagem de *acknowledgement*, confirmando que recebeu a mensagem *200 OK*.

Por fim, o *Asterisk* encaminha mensagem *200 OK* para o OpenBTS com destino ao terminal 2. O OpenBTS confirma que recebeu a mensagem *200 OK* com uma mensagem de *acknowledgement*, e notifica o terminal 2 que o terminal 1 atendeu a chamada. E a sessão GSM-GSM fica assim estabelecida.

3.4.1.1 Fim de sessão GSM -> GSM

O processo de fim de sessão está ilustrado na figura 3.7. O terminal 1 decide terminar a sessão com o terminal 2 e clica no botão de desligar a chamada. Neste momento é enviada uma notificação ao OpenBTS. O OpenBTS gera uma mensagem *SIP BYE*, com o SIP URI de destino (no formato: IMSI@<OpenBTSIPaddress>:<OpenBTSPort>) e envia ao *Asterisk*. O *Asterisk* confirma que recebe o pedido de fim de sessão com o envio de uma mensagem *200 OK* ao OpenBTS.

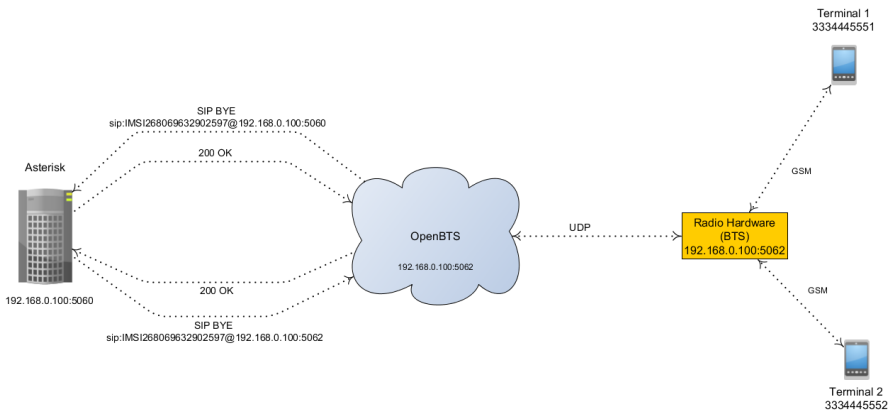

Figura 3.7: Representação da sinalização trocada no fim de sessão entre dois terminais GSM.

O *Asterisk* encaminha de novo o pedido de fim de sessão para o OpenBTS, e este confirma que recebeu o pedido com uma mensagem *200 OK*. Por fim, o OpenBTS envia a notificação ao terminal 2 a informar que a sessão foi terminada por parte do terminal 1.

Capítulo 4

Arquitectura IMS-GSM

Este capítulo tem por objectivo apresentar a arquitectura completa, desenhada para o protótipo experimental, e a forma como ocorrem as operações de início e fim de sessão. Esta arquitectura integra o nível de rádio GSM com uma rede *core IMS*, fazendo uso do protocolo SIP para a comunicação entre clientes IMS, registados no *core*, e terminais GSM, registados no OpenBTS.

4.1 Descrição da plataforma de teste

A plataforma de teste é ilustrada na figura 4.1.

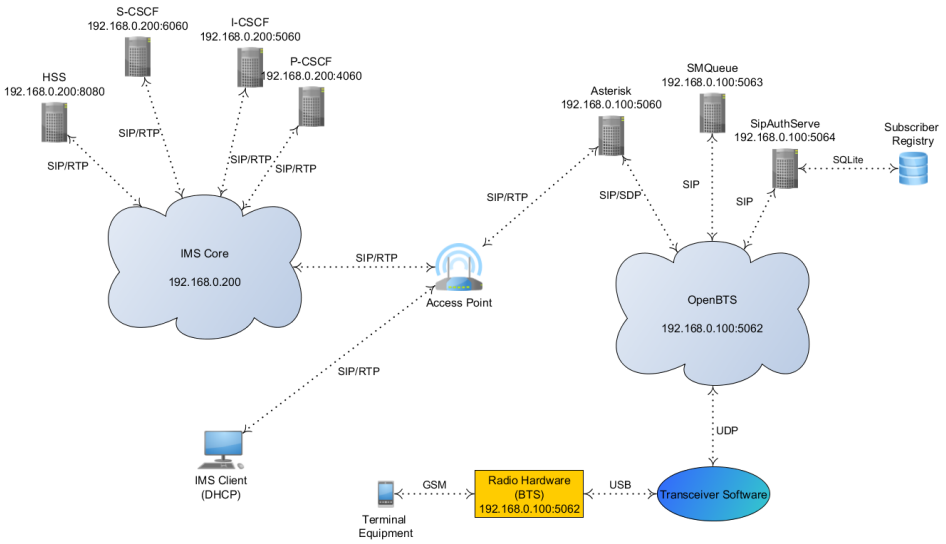

Figura 4.1: Representação da arquitectura IMS-GSM.

No lado direito da figura 4.1 é possível reconhecer a arquitectura GSM-SIP que foi descrita e abordada no capítulo anterior, que, como se disse, assume um papel fundamental no funcionamento desta plataforma de teste.

O *Access Point* é usado para criar uma rede local (*Linksys*), formada pela máquina que aloja o OpenBTS, a máquina que aloja o *IMS Core* e a máquina em que o cliente IMS está a aceder à rede. Para facilitar a configuração do *IMS Core* e do OpenBTS, ambos estão ligados por cabo Ethernet e possuem endereços IP estáticos (OpenBTS: 192.168.0.100, *IMS Core*: 192.168.0.200). O *IMS Client* acede à rede local por Wi-fi e é atribuído um endereço IP pelo *Access Point* (através do protocolo DHCP).

Por fim, o *IMS Core* é uma rede IMS típica, embora seja um *core* virtualizado que corre através do *software* VMWare. Este *core* irá encaminhar as mensagens SIP entre utilizadores, gerir utilizadores, autenticar e registar novos utilizadores, entre outras funções.

O módulo HSS funciona como a base de dados de utilizadores na rede e será requisitado na autenticação de utilizadores e no encaminhamento das mensagens SIP, mais precisamente quando for necessário descobrir o S-CSCF associado ao utilizador destino. Na perspectiva deste cenário, os módulos *Proxy-CSCF*, *Interrogating-CSCF* e *Serving-CSCF* também terão um papel preponderante na comunicação SIP entre utilizadores:

- Proxy-CSCF (P-CSCF): é o ponto de acesso de um utilizador à rede IMS. Toda a comunicação SIP, de e para um utilizador, tem que passar pelo respectivo P-CSCF;

- Interrogating-CSCF (I-CSCF): A sua principal missão é descobrir o S-CSCF associado ao utilizador para o qual a mensagem SIP se destina, e encaminhar a mesma mensagem para o S-CSCF que foi descoberto. Toda a comunicação SIP que provém de um destino externo ao *core IMS* é recebido por este módulo;

- Serving-CSCF(S-CSCF): faz a gestão do utilizador a que está associado e o encaminhamento das mensagens SIP de, e para, este utilizador. Tem de ter conhecimento de qual o P-CSCF a que o utilizador está a aceder no *core IMS*.

4.2 Cenários de interligação

4.2.1 Estabelecimento de ligação GSM -> IMS

O estabelecimento de ligação GSM -> IMS é um cenário definido por uma sessão que é originada por um terminal registado no OpenBTS e terminada num cliente registado no *core IMS*. A sinalização SIP no processo de estabelecimento da ligação está ilustrada nas figuras 4.2 e 4.3. No anexo deste documento é possível encontrar a captura *wireshark* do estabelecimento de sessão entre um terminal GSM e um cliente IMS.

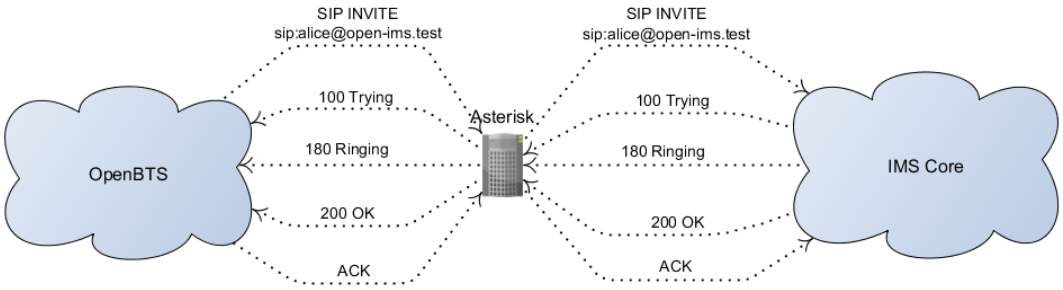

Figura 4.2: Sinalização de início de sessão *Asterisk*/OpenBTS - *IMS Core*.

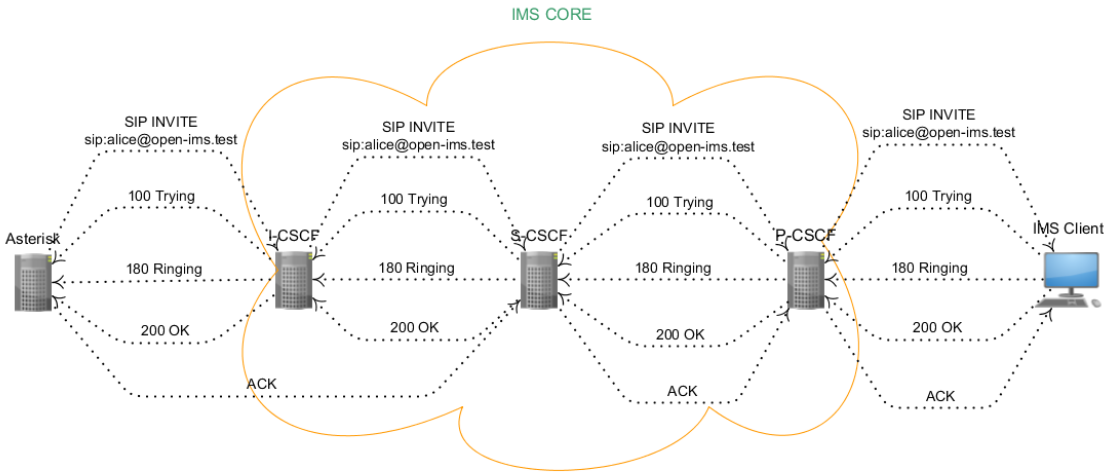

Figura 4.3: Sinalização de início de sessão *IMS Client* - *IMS Core*.

O processo começa com a marcação de uma determinada extensão no terminal (ex: 100) que, na configuração do *Asterisk*, irá corresponder a um determinado SIP URI (ex: alice@open-ims.test). O utilizador clica no botão e é enviado um pedido de sessão *SIP INVITE* do OpenBTS para o *Asterisk*. Este reconhece que o SIP URI pertence a uma rede externa e encaminha a mensagem *SIP INVITE* para o *core IMS*. Em cada salto (*hop*) que a mensagem *SIP INVITE* é encaminhada, é enviada uma resposta para o nó que a transmitiu com um aviso *100 Trying*, para o informar que está a tentar encontrar o utilizador de destino.

O pedido *SIP INVITE* é recebido pelo *Interrogating node*, I-CSCF, que recebe por definição todos os pedidos de sessão destinados a utilizadores registados no *core* e vindos de redes externas. A função deste servidor é encontrar o *Serving node*, S-CSCF, que gere a sessão do cliente IMS com o SIP URI de destino.

Assim que recebe o *SIP INVITE*, o S-CSCF encaminha este pedido de sessão para o *Proxy node*, P-CSCF. Posteriormente o P-CSCF irá finalmente fazer chegar a mensagem *SIP INVITE* ao utilizador destino. O cliente IMS de destino irá responder com uma mensagem *180 Ringing*, que será propagada até ao terminal, e que significa que o pedido de sessão

chegou ao destino. Neste momento estamos à espera que o cliente aceite o início de uma nova sessão.

Assim que o utilizador aceita o pedido de sessão é enviada uma mensagem *200 OK* ao terminal na rede OpenBTS, que indica que a sessão foi aceite. O terminal responde com uma mensagem de *acknowledgement*, reconhecendo que a sessão foi iniciada. Desta forma, a ligação fica estabelecida até que uma das partes termine a sessão, ou que algum problema ocorra na rede.

4.2.1.1 Fim da sessão GSM -> IMS

O processo do fim de sessão é um processo mais directo, como é possível verificar nas figuras 4.4 e 4.5. No anexo deste documento é possível encontrar a captura *wireshark* do processo de fim de sessão entre um terminal GSM e um cliente IMS.

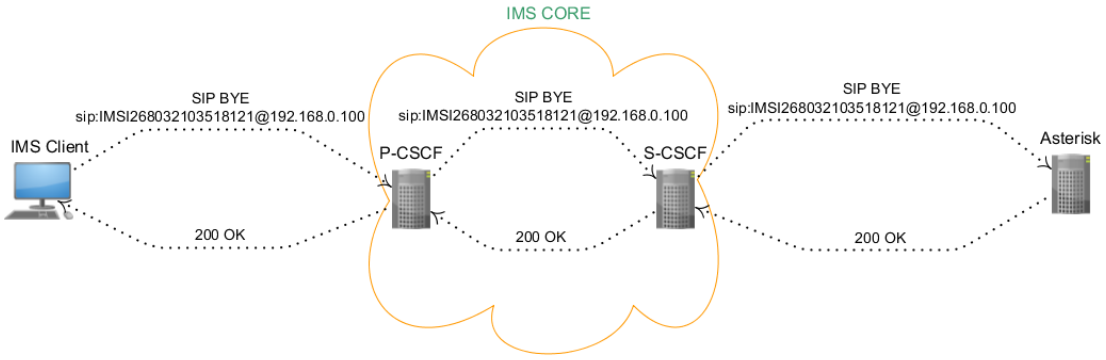

Figura 4.4: Sinalização de fim de sessão *IMS Client - IMS Core*.

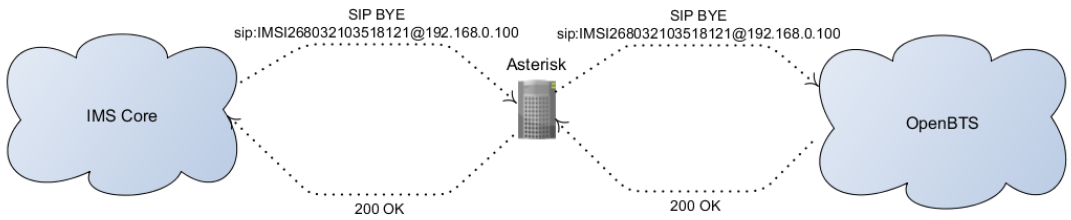

Figura 4.5: Sinalização de fim de sessão *IMS Core - Asterisk*/OpenBTS.

Quando o cliente IMS (ex: Alice) decide terminar a sessão (clicar no botão para terminar a sessão), é enviada uma mensagem *SIP BYE* ao/s outro/s utilizador/es da sessão. Este/s recebe/m a mensagem e reconhece/m que aquele utilizador deseja abandonar a sessão, e responde/m com uma mensagem afirmativa *200 OK*.

A sessão termina assim para os dois utilizadores ou pode continuar, caso seja uma sessão de grupo. No caso de ser o terminal móvel a terminar a sessão o processo é semelhante, dado que a mensagem *SIP BYE* cumpre o mesmo ciclo.

4.2.2 Estabelecimento de ligação IMS -> GSM

O estabelecimento de ligação IMS -> GSM representa o cenário em que uma sessão que é originada por um cliente IMS, registado no *core IMS*, e terminada num terminal registado no OpenBTS. A sinalização SIP no processo de estabelecimento da ligação está ilustrada nas figuras 4.6 e 4.7. No anexo deste documento é possível encontrar a captura *wireshark* do estabelecimento de sessão entre um cliente IMS e um terminal GSM.

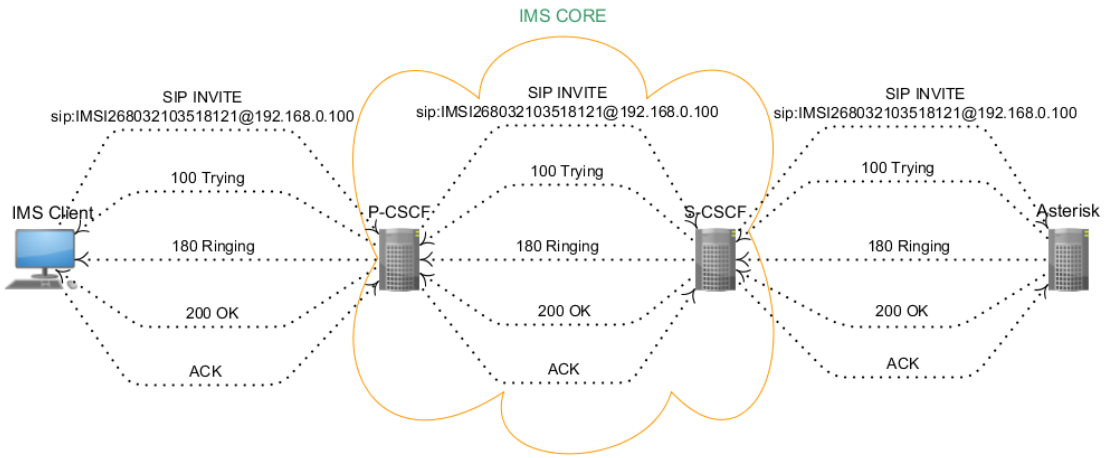

Figura 4.6: Sinalização de início de sessão *IMS Client - IMS Core*.

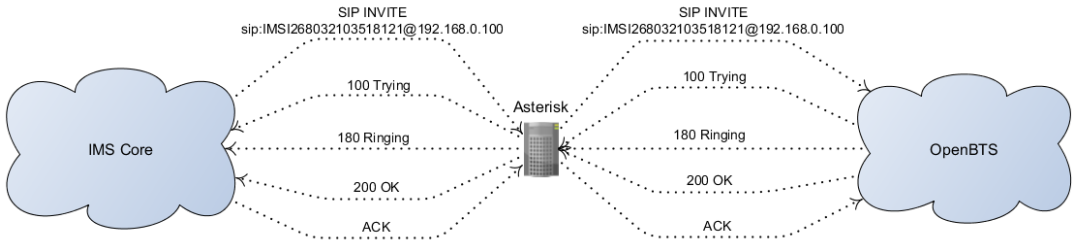

Figura 4.7: Sinalização de início de sessão *IMS Core - Asterisk*/OpenBTS.

O processo é semelhante ao início de uma sessão GSM -> IMS. O cliente IMS (ex: Alice) terá adicionado nos seus contactos um utilizador com um SIP URI (no formato: IMSI@<OpenBTSIPaddress>) correspondente a um terminal móvel registado no OpenBTS. Para iniciar uma sessão basta a Alice selecionar esse utilizador e o processo começa.

É enviado um pedido de sessão *SIP INVITE* para o *IMS Core*, mais precisamente para o P-CSCF onde a Alice está conectada. Sempre que a mensagem *SIP INVITE* é encaminhada, é enviada uma resposta *100 Trying* para o nó anterior, indicando que está a decorrer a descoberta pelo utilizador com que a Alice pretende comunicar. O P-CSCF encaminha então a mensagem *SIP INVITE* para o S-CSCF.

O S-CSCF irá verificar que este SIP URI contém um endereço IP que desconhece, o IP do OpenBTS (192.168.0.100), e que não pertence a esta máquina. Então irá encaminhar a

mensagem para o *Access Point*, que depois encaminha para o *Asterisk*.

O *Asterisk* recebe o *SIP INVITE* e contacta o *SIPAuthServe* de forma a autenticar o SIP URI de destino e confirmar que este se encontra registado e activo no OpenBTS. Em caso afirmativo, o pedido *SIP INVITE* é encaminhado para o OpenBTS com a indicação do IMSI do terminal de destino. O OpenBTS irá encaminhar o pedido para o terminal, que irá fazer soar o toque de chamada recebida e devolver uma resposta *180 Ringing* com destino à Alice.

Assim que o utilizador do terminal aceite a sessão, é enviado um *200 OK* para a Alice. A Alice reconhece que a sessão foi aceite e responde com uma mensagem de *acknowledgement*. A partir deste momento a ligação está estabelecida e terá início a sessão entre os dois utilizadores.

4.2.2.1 Fim da sessão IMS -> GSM

O fim de sessão é um processo mais simples, como é possível verificar nas figuras 4.8 e 4.9. No anexo deste documento é possível encontrar a captura *wireshark* do processo de fim de sessão entre um cliente IMS e um terminal GSM.

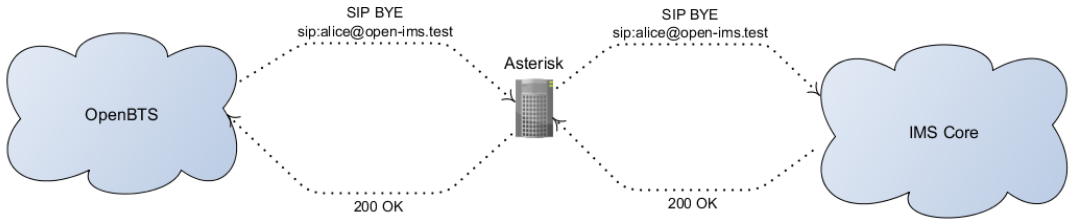

Figura 4.8: Sinalização de fim de sessão *IMS Core - Asterisk*/OpenBTS.

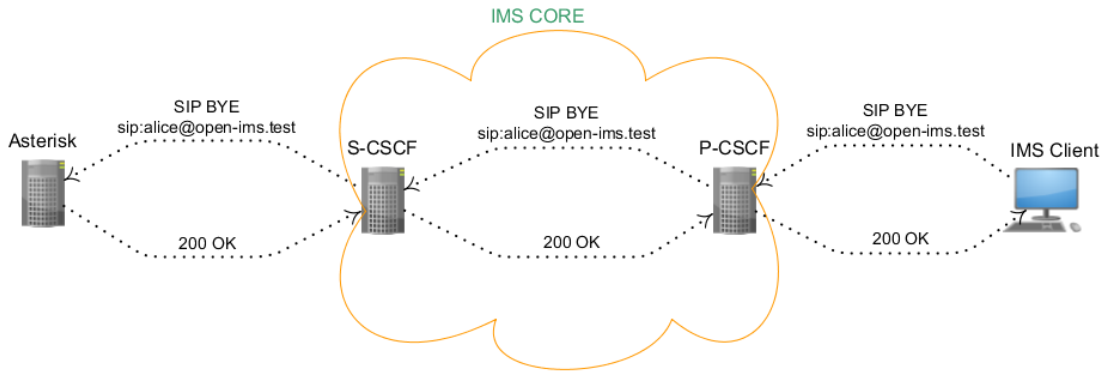

Figura 4.9: Sinalização de fim de sessão *IMS Client - IMS Core*.

Quando o utilizador do terminal móvel decide terminar a sessão (premir o botão de terminar a chamada ou desligar o terminal), é enviada uma mensagem *SIP BYE* ao/s outro/s utilizador/es da sessão. Este/s recebe/m a mensagem e reconhece/m que aquele

utilizador deseja abandonar a sessão e respondem com uma mensagem afirmativa *200 OK*.

A sessão termina assim para os dois utilizadores ou pode continuar, caso seja uma sessão de grupo. No caso de ser o cliente IMS a terminar a sessão o processo é semelhante, dado que a mensagem *SIP BYE* cumpre o mesmo ciclo.

Capítulo 5

Serviços Experimentais

Este capítulo destina-se à introdução de serviços experimentais na plataforma de teste, serviços estes que se destinam a melhorar a experiência do utilizador.

A implementação destes serviços tira proveito da sinalização SIP e da conversão de *SIP Messages* para SMS para garantir que tanto os clientes IMS como os utilizadores OpenBTS possam usufruir da subscrição destes serviços.

Neste sistema foi implementado um serviço experimental de chamadas perdidas. O serviço é lançado neste sistema por um servidor denominado *JBoss*, que é descrito em seguida.

5.1 Servidor JBoss

O *JBoss* é um servidor de aplicações *open source* baseado na plataforma *Java Enterprise Edition* e é totalmente implementado na linguagem *Java*. Assim, é possível usar o *JBoss* em qualquer sistema operativo que suporte a linguagem *Java*.

O JBoss não é mais do que um *container* que faz o *deploy* do serviço de chamadas perdidas neste sistema. Quando é compilado o programa que implementa o serviço, é gerado um ficheiro *war* (*Web application ARchive*). Este ficheiro é um arquivo onde são compactados todos os ficheiros e recursos inerentes ao programa que implementa o serviço.

No *script* de *deploy* do *JBoss*, é adicionado o caminho para o ficheiro *war* gerado pelo programa. Quando o *JBoss* é iniciado, os programas dos serviços são executados no sistema.

5.2 Application Server - AS

Os *Application Servers* são implementados no *core IMS* para adicionar novas funcionalidades à rede sob a forma de serviços multimédia [7]. Os operadores podem disponibilizar estes serviços aos clientes, os quais podem usufruir das suas funcionalidades mediante uma subscrição. Assim, é possível melhorar a experiência do cliente e, ao mesmo tempo, recolher dados estatísticos sobre o funcionamento da rede.

Dependendo do tipo de serviço, o AS pode mesmo comportar-se como um:

- *SIP Proxy* - O AS processa o pedido SIP e encaminha o pedido de volta para o S-CSCF. Enquanto processa o pedido, o AS pode modificar os cabeçalhos do pedido. Este foi o comportamento escolhido para o serviço de chamadas perdidas que foi implementado na rede OpenBTS+IMS;
- *Back-to-back user agent* (B2BUA)- Neste modo o AS gera um novo pedido para uma nova sessão e envia o pedido para o S-CSCF. Este será o comportamento do AS pode ser usado num serviço de desvio de chamadas ou *call forwarding*;
- *Originating user agent* - Neste modo, o AS comporta-se como um utilizador que gera pedidos de sessão SIP como um *SIP INVITE*;
- *Terminating user agent* - Neste modo o AS comporta-se como um utilizador a terminar o diálogo ou a sessão. Pode ser usado num serviço de *voice mail*, por exemplo.

Numa rede IMS cada AS está dedicado a um único serviço e é executado sobre o IMS (nível de aplicação). Um AS pode providenciar serviços no início, durante, ou no fim da chamada, mediante a sua configuração e da configuração dos seus *trigger points* e *Initial Filter Criteria* (IFCs) associados. No anexo deste documento é possível encontrar a configuração do AS, TP e IFC para o serviço de chamadas perdidas implementado neste sistema.

Os *triggers TP* são configurados para reportarem o acontecimento de eventos que o AS pretende receber (métodos SIP, sessão do terminal que origina a sessão, etc). Em seguida a associação do *triggers TP* ao AS é feita através do *Initial Filter Criteria*. A sua função é indicar ao S-CSCF quais eventos, definidos pelo TP, que devem ser encaminhados aos ASs correctos.

Considere-se o exemplo de um serviço de chamadas perdidas. Quando o serviço é iniciado, o AS faz um pedido ao HSS para ativar notificações para um conjunto de utilizadores que subscreveram o serviço de chamadas perdidas.

O *trigger point* configurado define que o AS do serviço de chamadas perdidas deve ser notificado quando ocorrerem eventos relacionados com um *SIP INVITE* ou com a sessão do utilizador que origina a sessão. Assim, se um *SIP INVITE* não chegar ao destino ou o utilizador não conseguir originar a sessão, o AS tem essa informação devido a estar a funcionar como um *proxy*, e pode ser indício de que o utilizador de destino não conseguiu estabelecer a sessão (ou seja, perdeu a chamada).

O IFC configurado associa o *trigger point* ao AS do serviço de chamadas perdidas e, diz ao S-CSCF qual o AS que deve notificar da ocorrência de um dos eventos descritos.

Assim, o S-CSCF sabe que todos os eventos relacionados com o *SIP INVITE* ou com a sessão relacionados com a sessão do terminal que origina a sessão, devem ser encaminhados para o AS (interface ISC (SIP)) que está dedicado ao serviço de chamadas perdidas.

A figura 5.1 demonstra as interfaces entre os blocos AS, HSS, S-CSCF, I-CSCF e P-CSCF numa arquitectura IMS, dentro de um possível cenário de interligação entre um terminal móvel e um cliente IMS.

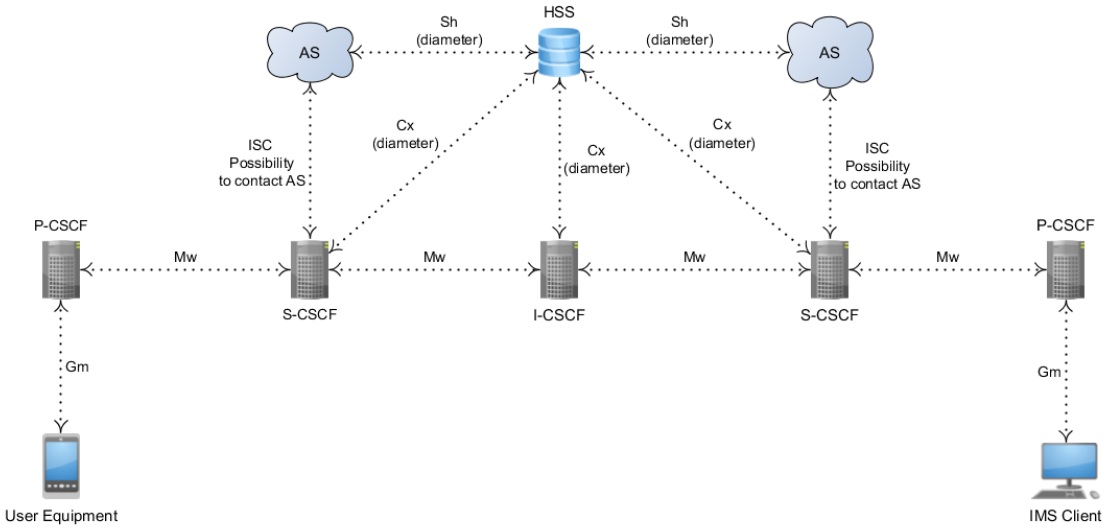

Figura 5.1: Diagrama da arquitectura IMS com a interação com um AS no estabelecimento de sessão entre dois utilizadores.

A interface Gm transporta todas as mensagens SIP entre o utilizador e a arquitectura IMS (P-CSCF). Já a interface Mw transporta as mensagens SIP entre diferentes CSCFs. Em ambas as interfaces ocorrem procedimentos como registo de utilizador e controlo de sessão.

A interface ISC (*IMS Service Control*) transporta todas as mensagens SIP entre o AS e o S-CSCF. Estas mensagens poderão ser os pedidos SIP com destino ao AS ou os pedidos SIP gerados pelo AS (no caso de estar em modo *Originating User Agent*).

Quando o processo de registo de utilizador termina com sucesso, o S-CSCF atribuído irá pedir ao HSS o perfil do utilizador através da interface Cx. O perfil indica quando é que as mensagens SIP devem ser encaminhadas para o respectivo AS (do serviço/s que o utilizador subscreveu).

A interface Cx transporta dados relativos ao utilizador entre o HSS e o S-CSCF ou I-CSCF. Os dados podem ser a localização do utilizador, bem como o perfil do utilizador, entre outros. Por fim, a interface Sh constitui o ponto de comunicação entre o AS e o HSS. O AS comunica com o HSS para informar dos utilizadores que subscreveram o seu serviço e também para saber informações sobre estes utilizadores (S-CSCF, localização, taxação,

entre outras).

5.3 Serviço de chamadas perdidas

O serviço de chamadas perdidas consiste num serviço subscrito pelos utilizadores que gera notificações quando um utilizador perde ou falha um pedido de estabelecimento de sessão. Considere-se o seguinte exemplo.

O utilizador Alice quer ligar para o Bob, mas o Bob não se encontra conectado ou já se encontra noutra sessão. A Alice recebe a notificação de que o Bob não se encontra disponível no momento e decide tentar mais tarde. No entanto, quando o Bob se conectar novamente à rede, irá receber uma notificação de que a Alice o tentou contactar.

A figura 5.1 demonstra o funcionamento do sistema com este exemplo com mais detalhe. No anexo deste documento está algum código relevante e capturas *wireshark* do funcionamento deste serviço.

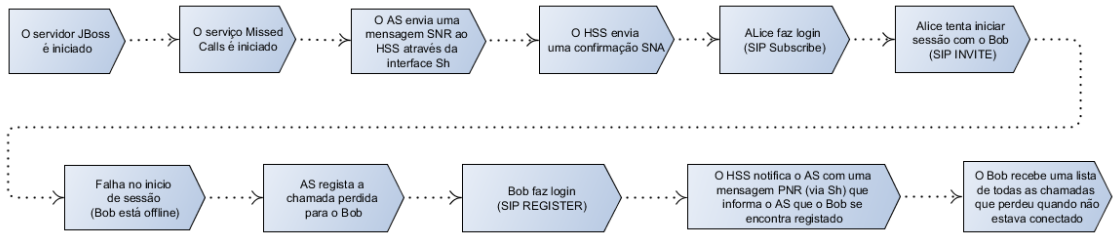

Figura 5.2: Exemplo do funcionamento do serviço de chamadas perdidas entre a Alice e o Bob.

O servidor *JBoss* é iniciado e implementa todos os serviços que estão associados, incluindo o serviço de chamadas perdidas. Quando este serviço é inicializado, o AS envia um pedido de subscrição de notificações (*Subscribe Notifications Request* - SNR) ao HSS, através da interface Sh (interface de comunicação AS-HSS).

Este pedido subscreve os utilizadores cujos SIP URIs estão incluídos no ficheiro de configuração do serviço (diameter-openims.properties, incluído em Anexo I). O HSS envia uma confirmação do pedido (*Subscribe Notifications Answer* - SNA). A partir deste momento estão activadas as notificações para estes utilizadores.

A Alice conecta-se à rede e tenta iniciar uma sessão com o Bob. A Alice envia um *SIP INVITE* ao seu P-CSCF, que encaminha o mesmo para S-CSCF que serve a Alice. Como está a funcionar como SIP Proxy, o AS recebe este *SIP INVITE*, processa o pedido, e encaminha de volta para o S-CSCF. Desta forma, o AS coloca-se no diálogo SIP entre a Alice e o Bob, de forma a que possa detectar quando uma chamada não é atendida.

O S-CSCF da Alice interroga o I-CSCF do Bob pelo S-CSCF do Bob. O I-CSCF do Bob interroga o HSS a fim de receber a localização do seu S-CSCF. Porém o Bob não está registado e o HSS responde com uma mensagem de erro, indicando que o Bob não está activo e não pode ser iniciada a sessão.

A Alice é informada que houve uma falha no início de sessão e decide tentar mais tarde. O AS, como está no diálogo SIP, também fica a saber que ocorreu o erro no estabelecimento de sessão. E é neste momento que o AS acrescenta informação na lista do Bob de que este recebeu uma chamada numa determinada data.

Mais tarde o Bob regista-se na rede e envia um *Push Notification Request*, PNR, ao AS, para verificar se alguém tentou contactar quando esteve inactivo. O AS processa esta informação como é sugerido na figura 5.2.

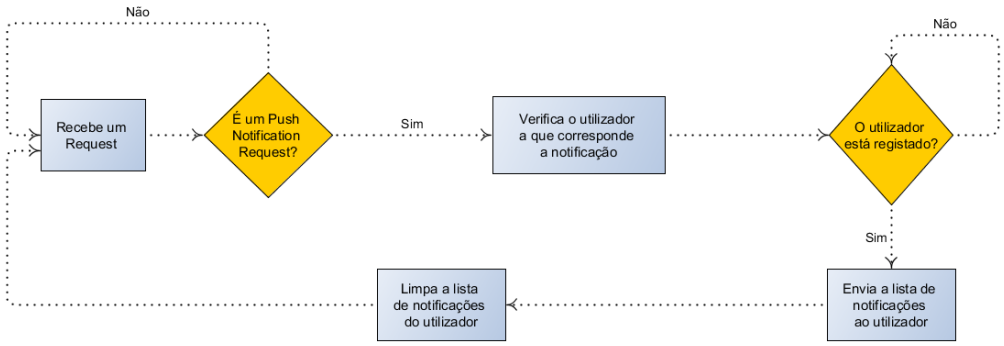

Figura 5.3: Fluxograma que explica o processo que ocorre quando um utilizador perde uma chamada.

A mensagem PNR é recebida e o AS verifica que foi uma alteração no estado do Bob que originou a mensagem. Em seguida, o AS pede ao HSS para confirmar se o Bob está registado na rede, de forma a poder enviar a lista de chamadas perdidas que o Bob tem para receber. O AS só irá enviar a lista ao Bob se este estiver registado na rede.

Assim, como o Bob está registado, irá receber do AS a lista com informação de todas as chamadas que recebeu enquanto esteve de-registado da rede. Neste exemplo, irá verificar que recebeu uma chamada da Alice enquanto este inactivo.

Capítulo 6

Conclusões

Nesta dissertação foi desenvolvido um protótipo experimental que integra o nível rádio GSM com uma rede *core IMS* que oferece serviços de chamadas de voz e um serviço de chamadas perdidas. Para criar a rede GSM foi usada uma BTS implementada em dispositivo SDR através do *software* OpenBTS, que implementa as camadas inferiores da pilha de protocolos GSM.

Este *software* também oferece a funcionalidade de um terminal GSM poder ser visto como um terminal SIP por uma rede externa através do módulo *Asterisk*. Este módulo funciona como um *Private Branch Exchange*, PBX, e realiza funções de controlo de chamadas, que normalmente seriam da responsabilidade de um *Mobile Switching Center*, MSC, numa rede convencional GSM. Através da configuração do *Asterisk*, foi possível integrar a rede GSM com uma rede *core IMS* e estabelecer sessões de voz entre um terminal GSM, registado no sistema OpenBTS, e um cliente IMS, registado no *core IMS*.

Foi ainda implementado um serviço experimental que oferece uma nova funcionalidade: um serviço de chamadas perdidas. Para o utilizador que subscreve este serviço, é oferecida a funcionalidade de notificação de pedidos de estabelecimento de sessão que foram recebidos enquanto este esteve de-registado da rede.

A implementação e testes destes serviços revelou a possibilidade de implementar outros exemplos de serviços. Um exemplo seria um serviço de desvio de chamadas ou *call forwarding*, que consistiria na possibilidade de encaminhar uma chamada, que seria destinada ao seu terminal, para outro terminal distinto que esteja na sua posse.

Não obstante dos serviços implementados, e de outros que se poderiam implementar, também se poderiam realizar outro tipo de testes que não foram possíveis em tempo útil. Nomeadamente, a utilização de simuladores de carga para simular tráfego na rede e estabelecer sessões de teste nestas condições. O objectivo seria medir a qualidade do serviço de voz num cenário mais próximo da realidade, em que não existem apenas dois

utilizadores na rede a estabelecer uma sessão de voz entre os mesmos.

A integração das múltiplas tecnologias num protótipo funcional permite a análise dos diferentes protocolos envolvidos, constituíndo uma plataforma para desenvolvimento de outros trabalhos, nomeadamente ao nível dos serviços.

Bibliografia

[1] S. Barve, A. Akotkar, A. Chavan, A. Kumar e M. Dhaigude. "Open Source Software Defined Radio Using GNU Radio And USRP'". Em: *International Journal of Scientific Technology Research* 3.5 (2014).

[2] R. Bryant, L. Madsen e J. V. Meggelen. *Asterisk. The Definitive Guide*. 4. O'Reilly Media, Inc., 2013.

[3] J. R. Machado-Fernández. "Software Defined Radio: Basic Principles and Applications". Em: *Facultad de Ingeniería* 24.38 (2015), pp. 79–96.

[4] R. Networks. *OpenBTS Application Suite. User Manual*. 4. Range Networks, 2014.

[5] R. Newton. *Asterisk Project Wiki*. Ago. de 2014. URL: https://wiki.asterisk.org/wiki/display/AST/Asterisk+Architecture%2C+The+Big+Picture.

[6] A. S. Pawar e A. S. Pawar. "STUDY OF THE GSM NETWORK". Em: *International Journal of Advanced Engineering Research and Studies* 1.38 (2012), pp. 287–292.

[7] M. Poikselka, G. Mayer, H. Khartabil e A. Niemi. *The IMS. IP Multimedia Concepts and Services in the Mobile Domain*. 4. John Wiley & Sons Ltd, 2004.

[8] M. Rahnema. "Overview of the GSM system and protocol architecture". Em: *IEEE Communications magazine* 31.4 (1993), pp. 92–100.

[9] A. S. TANENBAUM e D. J. WETHERALL. *Computer Networks*. 4. PRENTICE HALL, 2011.

[10] T. Turletti, H. J. Bentzen e D. Tennenhouse. "Toward the software realization of a GSM base station". Em: *IEEE Journal on selected areas in communications* 17.4 (1999), pp. 603–612.

[11] T. Ulversoy. "Software defined radio: Challenges and opportunities". Em: *IEEE Communications Surveys & Tutorials* 12.4 (2010), pp. 531–550.

ANEXO I

ANEXO

A imagem I.1 é a captura *wireshark* do processo de registo de um terminal GSM na rede OpenBTS (com um filtro ao protocolo SIP).

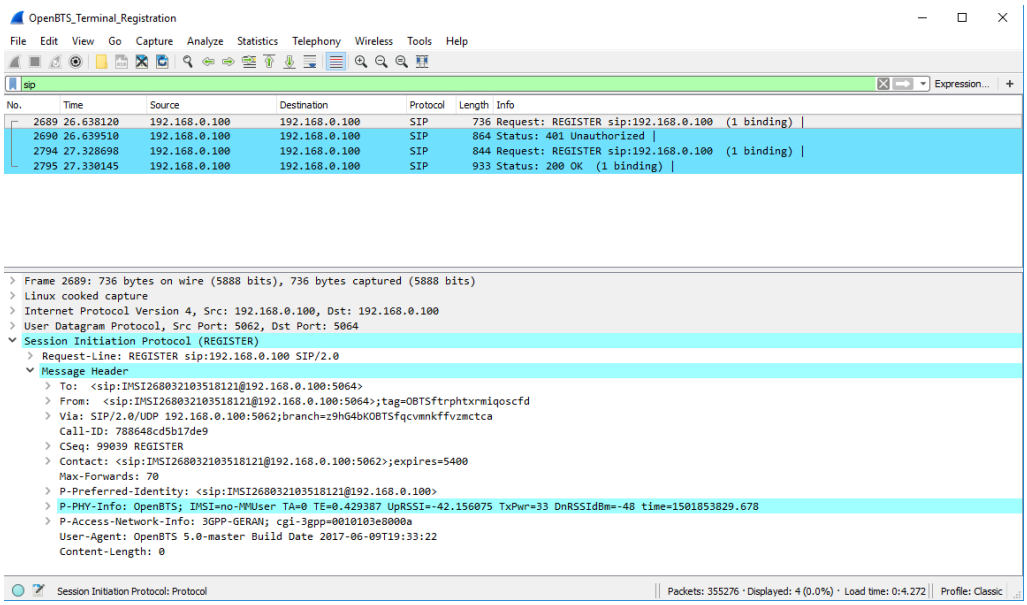

Figura I.1: Captura *wireshark* do registo de um terminal GSM na rede OpenBTS, na perspectiva do protocolo SIP.

A imagem I.2 é a captura *wireshark*, recolhida na máquina que corre o *software* OpenBTS, do processo de estabelecimento de uma ligação GSM-GSM (com um filtro ao protocolo SIP).

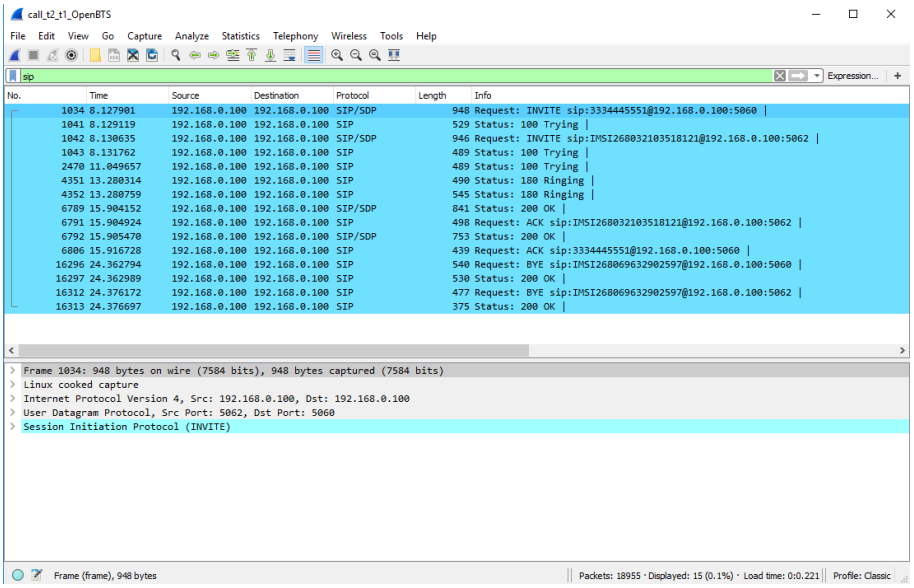

Figura I.2: Captura *wireshark*, na máquina OpenBTS, de uma sessão de voz entre dois terminais GSM.

A imagem I.3 é a captura *wireshark*, recolhida na máquina que corre o *software* OpenBTS, do processo de estabelecimento de uma ligação IMS-GSM (com um filtro ao protocolo SIP).

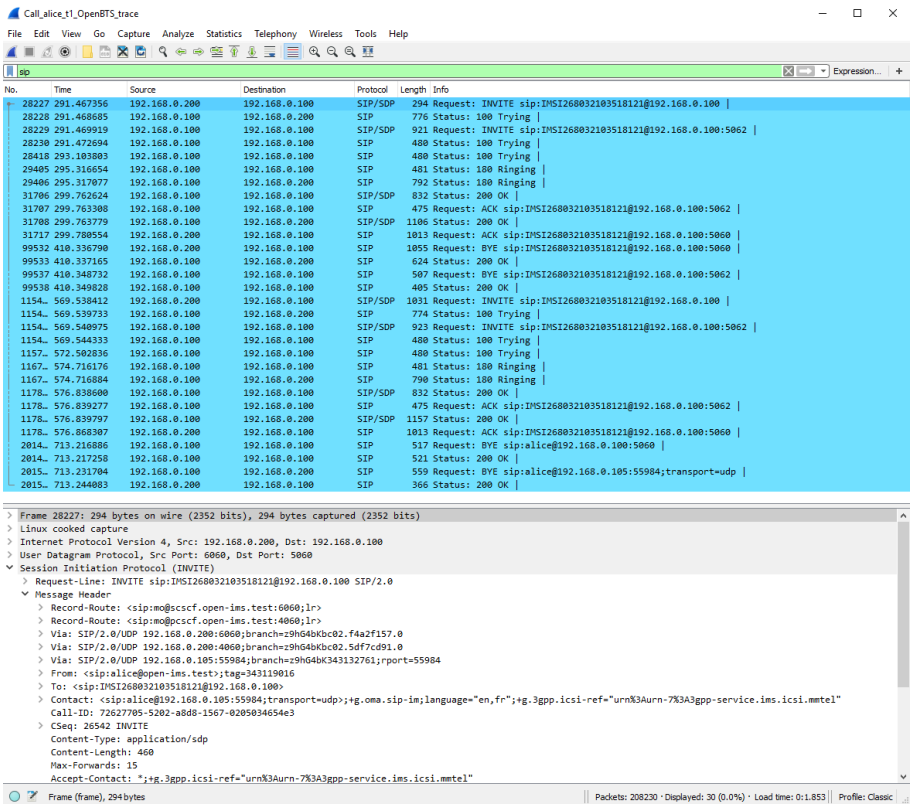

Figura I.3: Captura *wireshark*, na máquina OpenBTS, de uma sessão entre o cliente IMS Alice e um terminal móvel GSM.

A imagem I.4 é a captura *wireshark*, recolhida na máquina que corre o *core IMS*, do processo de estabelecimento de uma ligação IMS-GSM (com um filtro ao protocolo SIP).

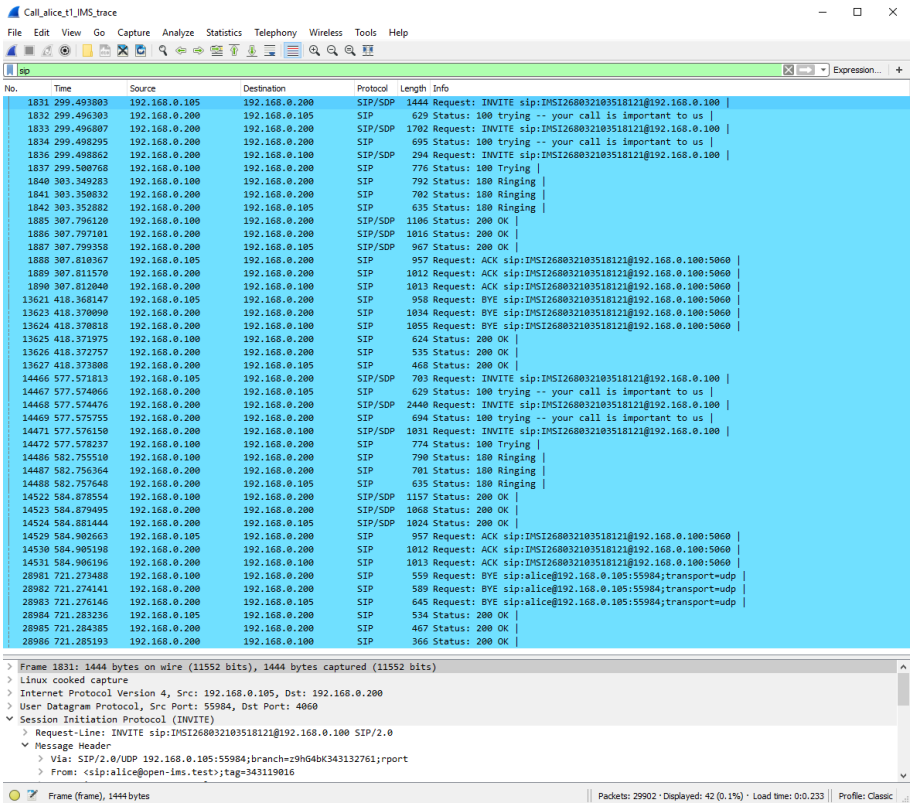

Figura I.4: Captura *wireshark*, na máquina *IMS core*, de uma sessão entre o cliente IMS Alice e um terminal móvel GSM.

A imagem seguinte demonstra o conteúdo do ficheiro que contém a configuração do plano de chamadas do *Asterisk*.

```
1  [phones]
2
3  exten => 3334445551,1,NoOp(Calling Terminal 1)
4  same => n,Dial(SIP/terminal1)
5  same => n,Hangup
6
7  exten => 3334445552,1,NoOp(Calling Terminal 2)
8  same => n,Dial(SIP/terminal2)
9  same => n,Hangup
10
11 exten => 100,1,NoOp(Calling ...)
12 ;same => n,Wait(10)
13 same => n,Dial(SIP/alice,30)
14 same => n,Hangup
15
16 exten => IMSI268032103518121,1,NoOp(Call Received)
17 ;same => n,Wait(10)
18 same => n,Dial(SIP/openbts)
19 same => n,Hangup
20
21
22
23
24
25
```

Figura I.5: Ficheiro de configuração do plano de chamadas do Asterisk.

As imagens seguintes demonstram o conteúdo do ficheiro que contém a configuração principal do *Asterisk*.

```
[general]
context=phones
allowoverlap=no
udpbinaddr=0.0.0.0
bindaddr=192.168.0.100
bindport=5060
tcpenable=no
tcpbinaddr=0.0.0.0
transport=udp
srvlookup=yes
nat=no
insecure=invite,port
canreinvite=no
qualify=no
fromdomain=192.168.0.100
allowguest=yes

[authentication]
[basic-options](!)
    dtmfmode=rfc2833
    context=from-office
    type=friend
[natted-phone](!,basic-options)
    directmedia=no
    host=dynamic
[public-phone](!,basic-options)
    directmedia=yes
[my-codecs](!)
    disallow=all
    allow=ilbc
    allow=g729
    allow=gsm
    allow=g723
    allow=ulaw
[ulaw-phone](!)
    disallow=all
    allow=ulaw
    allow=gsm

;-------- Peers ------------------------
[terminal1]
    type=friend
    username=IMSI268032103518121
    context=phones
    allow=ulaw,alaw,gsm
    secret=12345678
    host=192.168.0.100
    port=5062
```

Figura I.6: Ficheiro de configuração das comunicações SIP do *Asterisk*.

```

;-------- Peers ------------------------
[terminal1]
    type=friend
    username=IMSI268032103518121
    context=phones
    allow=ulaw,alaw,gsm
    secret=12345678
    host=192.168.0.100
    port=5062

[terminal2]
    type=friend
    username=IMSI268069632902597
    context=phones
    allow=ulaw,alaw,gsm
    secret=12345678
    host=192.168.0.100
    port=5062

[alice]
    type=friend
    username=bob
    context=phones
    allow=ulaw,alaw,gsm
    secret=87654321
    host=open-ims.test
    ;fromuser=IMSI268032103518121
    canreinvite=no

; For incoming calls
;[outside]
;   type=friend
;   username=alice@open-ims.test
;   context=incoming
;   allow=ulaw,alaw,gsm
;   secret=12345678
;   host=open-ims.test


```

Figura I.7: Ficheiro de configuração das comunicações SIP do *Asterisk*.

Entramos agora na configuração dos serviços experimentais. As imagens que se seguem mostram a configuração, na consola do HSS, do *Application Server*, AS, *Trigger Point* e *Initial Filter Criteria* associados ao serviço de chamadas perdidas.

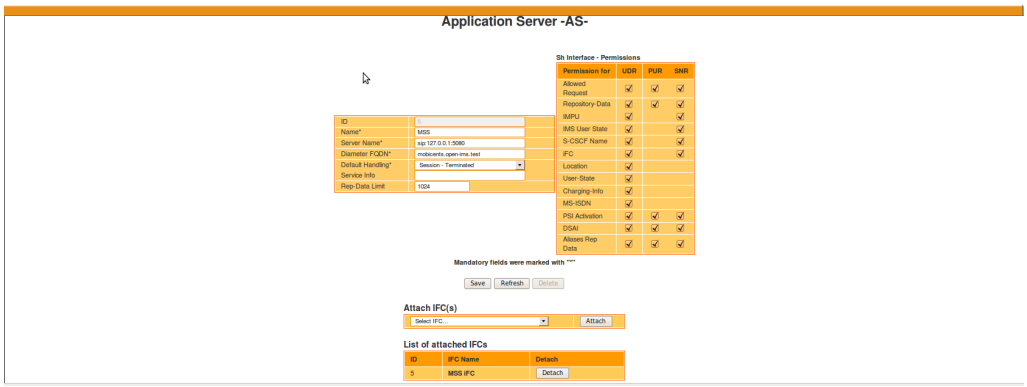

Figura I.8: Configuração do AS do serviço de chamadas perdidas, na consola do HSS.

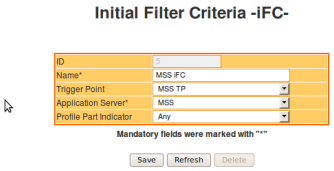

Figura I.9: Configuração do *Initial Filter Criteria* associado ao AS do serviço de chamadas perdidas, na consola do HSS.

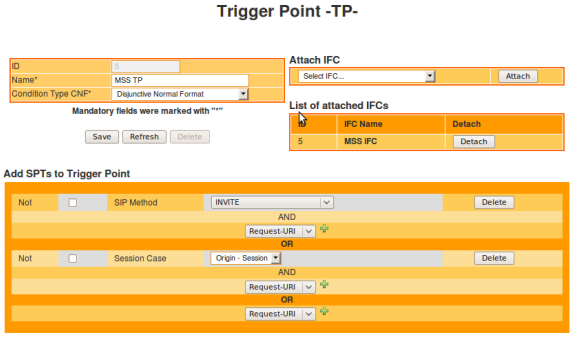

Figura I.10: Configuração do *trigger point* associado ao AS do serviço de chamadas perdidas, na consola do HSS.

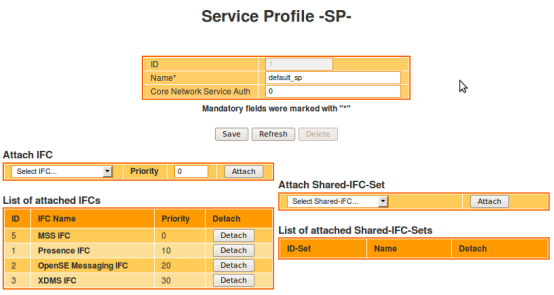

Figura I.11: Configuração do perfil do serviço de chamadas perdidas, na consola do HSS.

As imagens seguintes demonstram as capturas *wireshark* do funcionamento do serviço de chamadas perdidas. A notificação de chamada perdida é uma mensagem SIP no formato *text/html* (no final da segunda imagem).

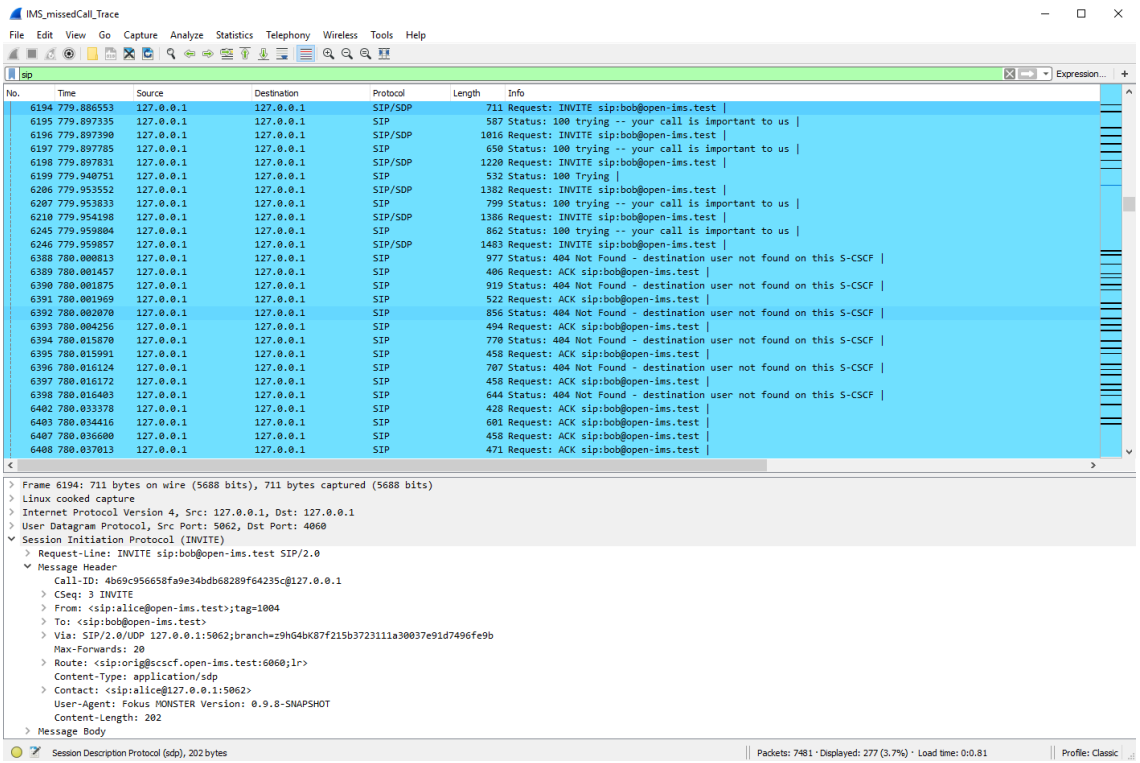

Figura I.12: Captura *wireshark* do funcionamento do serviço de chamadas perdidas.

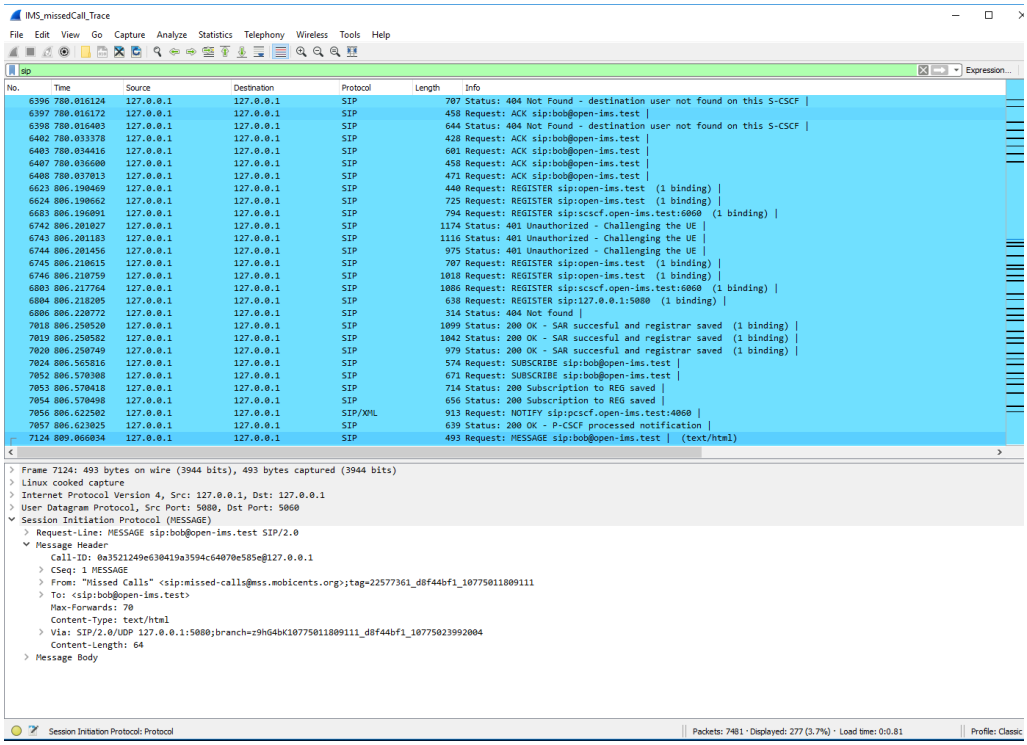

Figura I.13: Continuação da captura *wireshark* do funcionamento do serviço de chamadas perdidas.

Os anexos seguintes são respectivos ao serviço de chamadas perdidas. A imagem I.14 demonstra o ficheiro da configuração (endereço IP, porto e *realm*) do *Application Server* e do *Home Subscriber Server*. Também neste ficheiro encontram-se os SIP URIs dos utilizadores que subscrevem este serviço.

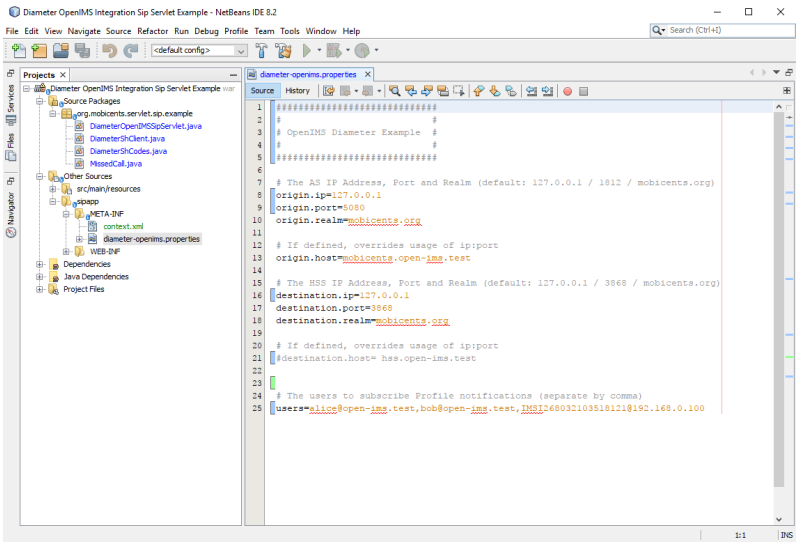

Figura I.14: Ficheiro com a configuração do AS, HSS e dos utilizadores que subscrevem o serviço de chamadas perdidas.

As imagens I.15 e I.16 ilustram o método que inicializa os parâmetros para o AS, HSS

e os SIP URIs dos utilizadores (acedendo ao ficheiro referido anteriormente). Além disso, este método cria e envia um pedido de subscrição de notificações de chamadas perdidas aos utilizadores e recebe as respostas a estes pedidos.

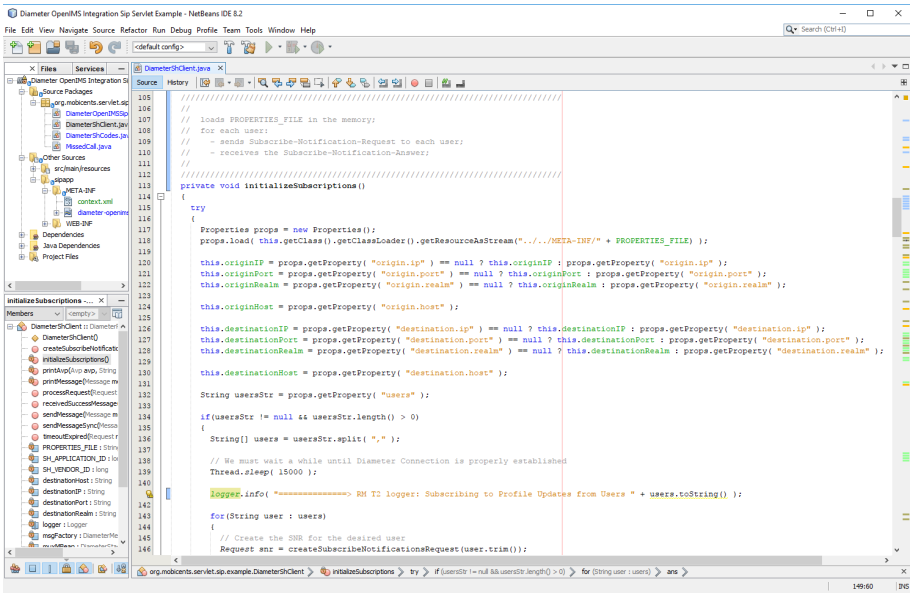

Figura I.15: Método de inicialização dos parâmetros para o *Subscribe Notification Request - SNR*.

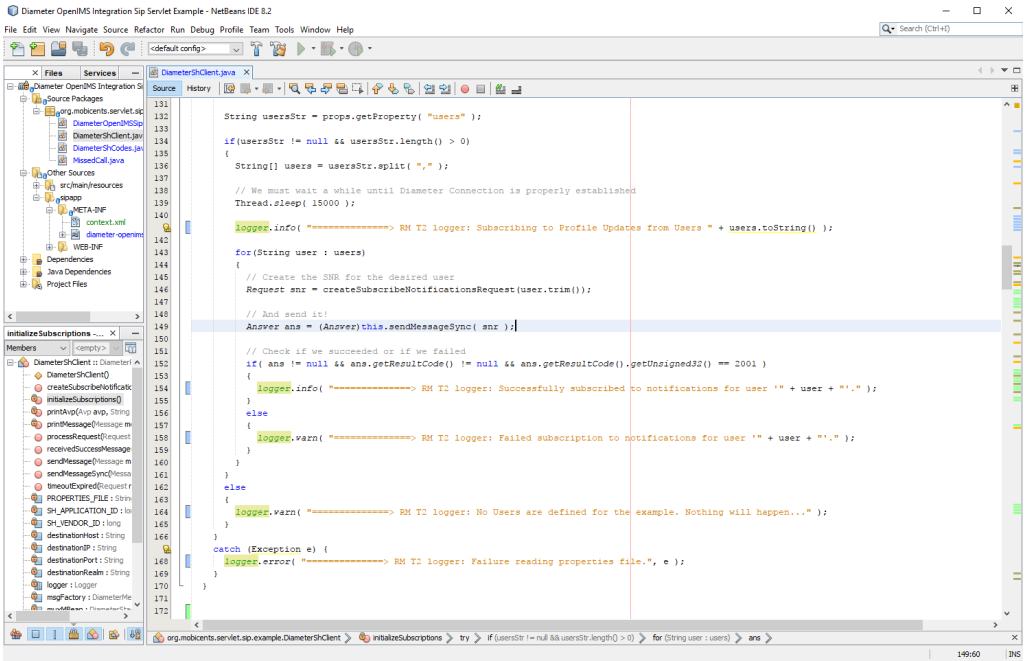

Figura I.16: Continuação do método de inicialização dos parâmetros para o *Subscribe Notification Request - SNR*.

As imagens I.17 e I.18 contém o método responsável por criar um pedido de subscrição de notificações de chamadas perdidas, ou SNR.

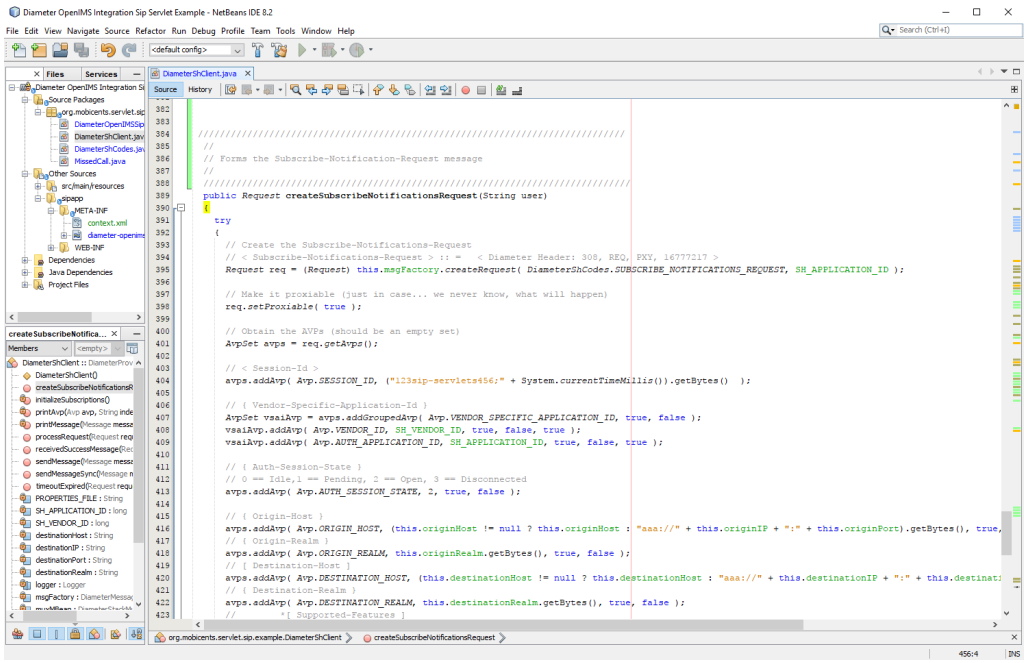

Figura I.17: Método que cria uma mensagem SNR.

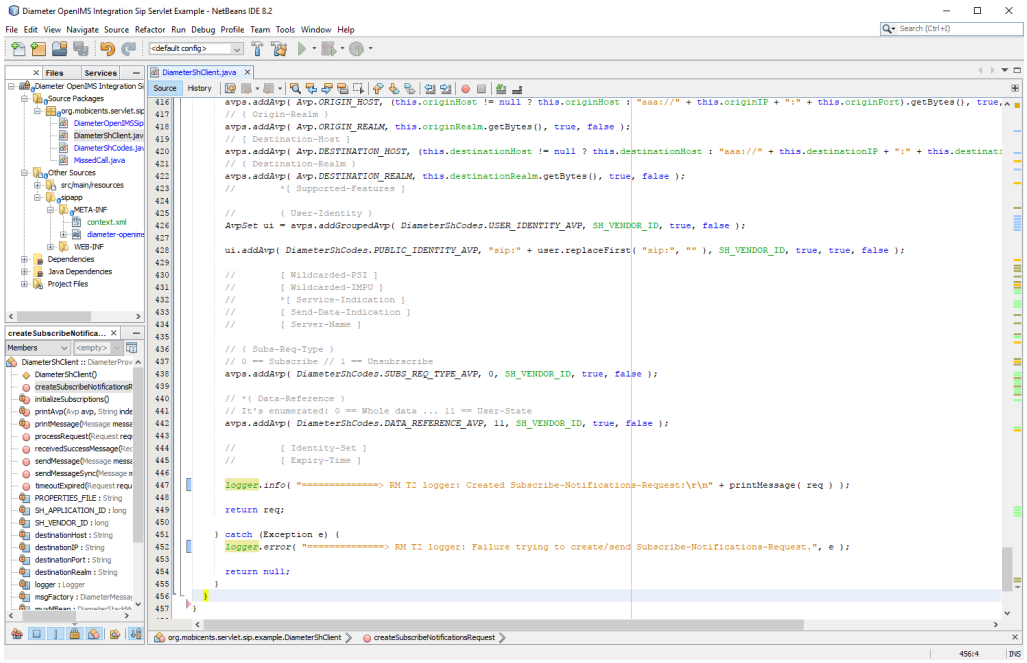

Figura I.18: Continuação do método que cria uma mensagem SNR.

As imagens I.19 e I.20 ilustram o método responsável por processar as mensagens SNR. Quando um utilizador perde uma chamada, este método cria uma notificação e adiciona a uma lista de notificações destinadas a este utilizador. Quando este fica *online*, recebe todas as notificações que tiver para receber.

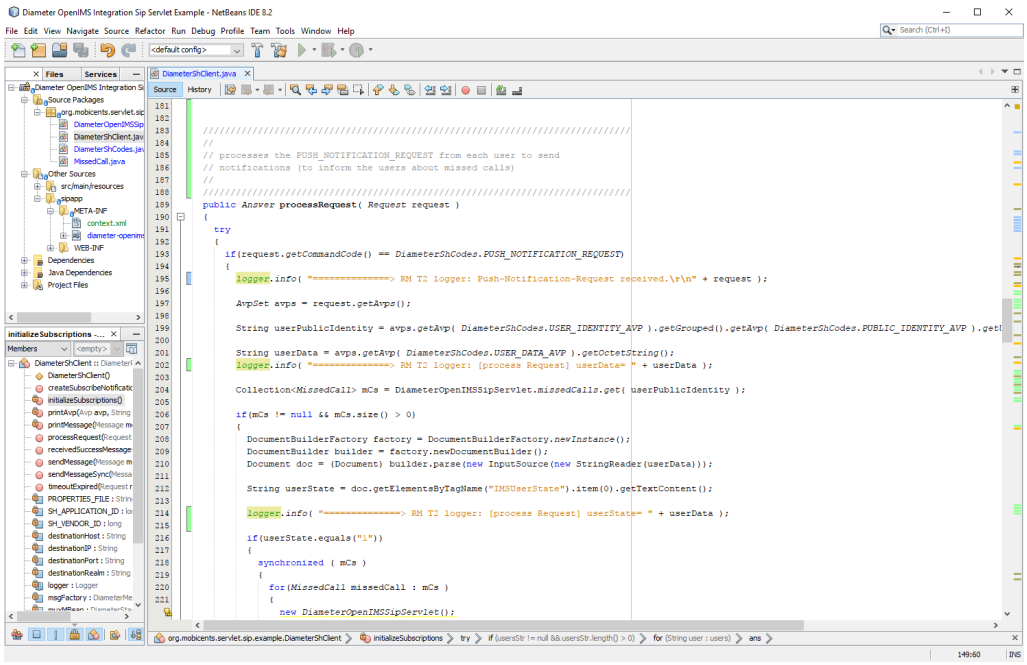

Figura I.19: Método que processa uma mensagem *Push Notification Request - PNR*.

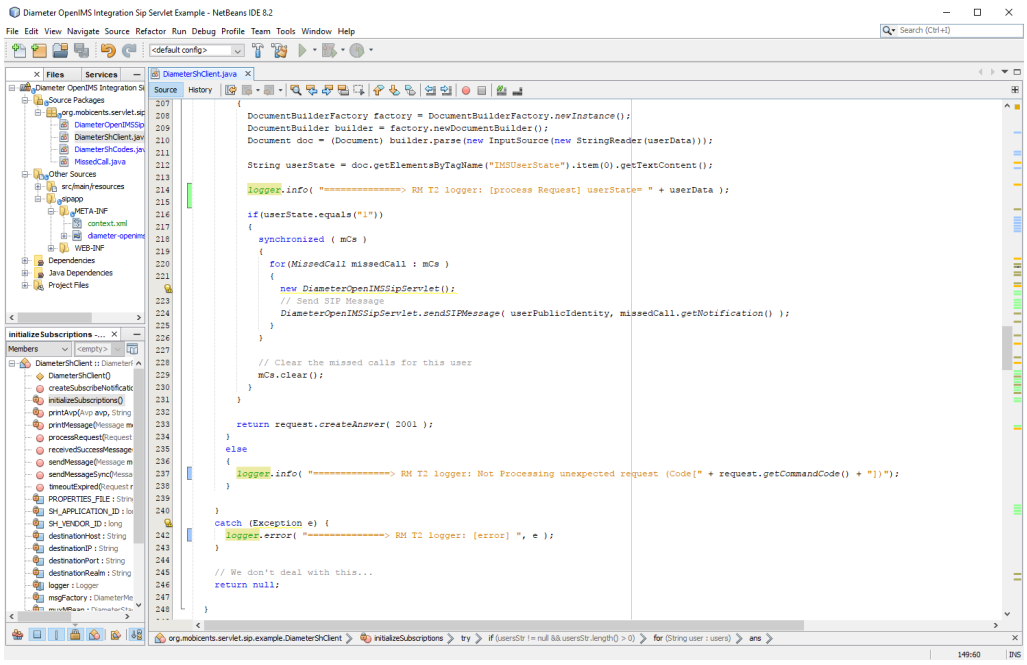

Figura I.20: Continuação do método que processa uma mensagem *Push Notification Request - PNR*.

A imagem I.21 demonstra a rotina responsável por detectar quando existe uma chamada que não foi iniciada por algum erro ocorrido. É nesta rotina que são geradas as notificações de chamada perdida a enviar ao utilizador que perdeu a chamada.

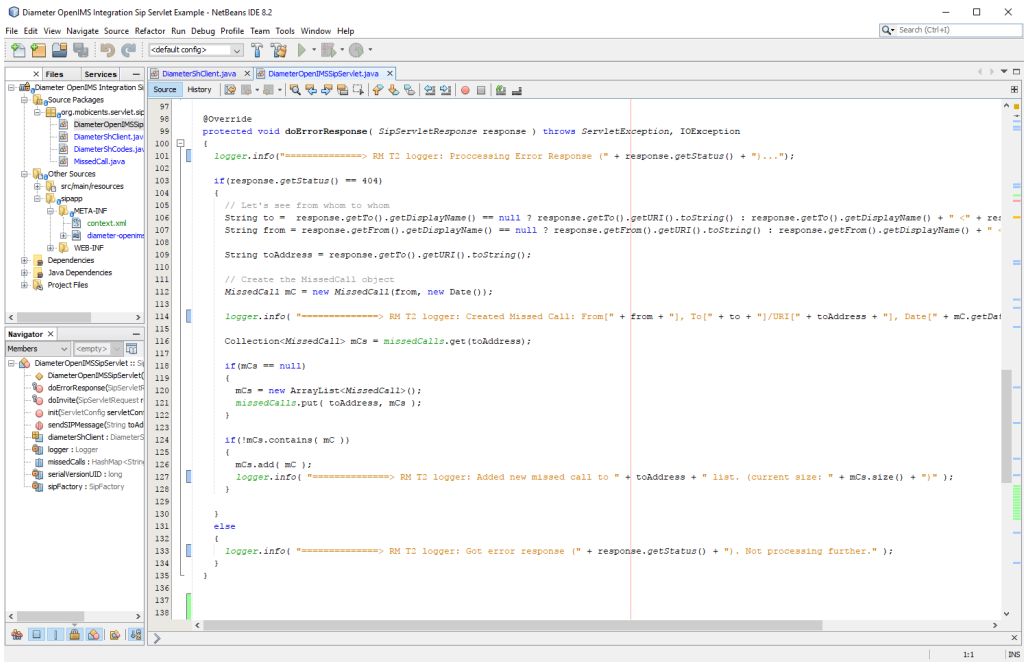

Figura I.21: Rotina onde é adicionada a informação de uma chamada perdida à lista de chamadas perdidas do utilizador.