

Mestrado em Engenharia Informática

Estágio

Relatório Final

HD Voice & HD Video

(Voz e vídeo de alta definição)

Daniel José Pais Baltasar

djpb@student.dei.uc.pt

Orientador:

Rui Pedro Paiva

Data: 12 de Julho de 2012



FCTUC DEPARTAMENTO
DE ENGENHARIA INFORMÁTICA
FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

Resumo

O presente documento descreve um projecto de engenharia de software que tem, como objectivo, melhorar a qualidade global de funcionalidades relacionadas com voz e vídeo, de uma suite de produtos de telecomunicações. A importância do projecto prende-se com o interesse em destacar um produto em mercados internacionais de operadores móveis, pelo acréscimo de qualidade audiovisual e interactiva inerente às funcionalidades em questão.

Palavras-chave

Alta definição, Codecs, H.264, Long Term Evolution, Rich Communication Suite, RCSs, SILK, *Videoshare*

Índice

1.Introdução	1
1.1. A Empresa	1
1.2. WIT Communications Suite	1
1.3. Metodologia de trabalho	2
1.4. Motivação	3
2.Estado da Arte	4
2.1. Competidores	6
2.2. Áudio	7
2.3. Vídeo	9
2.4. 4G e LTE	11
3.HD Voice & HD Video	13
3.1. Requisitos	13
3.2. Arquitectura	15
3.2.1. Overview	15
3.2.2. Camadas de aplicação	16
3.2.2.1. UI – Cliente Android	18
3.2.2.2. CommLib	24
3.2.2.3. PJSIP	30
3.2.2.4. Externals	30
3.2.3. Protocolos	30
3.3. Trabalho realizado	35
4.Conclusões	37
5.Referências	38

Lista de Figuras

Figura 1 – Crescimento de aplicações, por funcionalidade	4
Figura 2 – Utilização de aplicações.....	6
Figura 3 – Efeito do <i>deblocking filter</i> em H.264.....	10
Figura 4 – Evolução da velocidade móvel.....	12
Figura 5 – Arquitectura global do WCS.....	15
Figura 6 – Protocolos de comunicação entre clientes e servidor.....	16
Figura 7 – Representação dos clientes WCS por camadas de aplicação.....	16
Figura 8 – Ciclo de vida de uma <i>Activity</i> , em Android	18
Figura 9 – Pilha LIFO de <i>Activities</i> , utilizada pelo Android	19
Figura 10 – Hierarquia de <i>layouts</i> e <i>Views</i> , no Android.....	19
Figura 11 – Diagrama de classes – <i>managers</i> (WMC-Android)	21
Figura 12 – Diagrama de sequência – <i>Activities</i> e chamada (WMC-Android)	22
Figura 13 – Diagrama de sequência – <i>Activities</i> e fim de chamada (WMC-Android) ...	23
Figura 14 – Diagrama de classes – Communicator (CommLib)	25
Figura 15 – Diagrama de classes – serviços (CommLib)	26
Figura 16 – Diagrama de classes – <i>CallManager</i> (CommLib)	28
Figura 17 – Diagrama de classes – eventos (CommLib).....	29
Figura 18 – Iniciação de chamada, via SIP	32

Lista de Gráficos

Gráfico 1 – Tendência de circulação de dados, por funcionalidade.....	5
Gráfico 2 – Signal-to-noise ratio a 10Hz, 176x144	11
Gráfico 3 – Signal-to-noise ratio a 30Hz, 352x288	11

Lista de Tabelas

Tabela 1 – Requisitos por cliente WCS.....	14
Tabela 2 – Requisitos por camada de aplicação.....	17
Tabela 3 – Tipos de pedidos do protocolo SIP.....	31
Tabela 4 – Tipos de resposta do protocolo SIP.....	31

Glossário

Termo	Descrição
Android	Sistema operativo para telemóveis, desenvolvido pela Google.
iOS	Sistema operativo para telemóveis, desenvolvido pela Apple.
IPTV	Designação dada a serviços de televisão sobre uma rede IP.
Lossy	Termo que designa um tipo de compressão de dados, onde há perda de informação durante o processo.
LTE	Long Term Evolution. Designação oficial para a 4ª geração de redes móveis.
MOS	Mean Opinion Score. Método quantitativo normalmente usado para medição da qualidade de <i>media</i> que é transmitido.
MTU	Maximum Transmission Unit. Valor definido numa rede para limitar o tamanho dos pacotes transmitidos entre nós.
Product Backlog	Artefacto de metodologias ágeis de desenvolvimento, normalmente indicando os objectivos/funcionalidades centrais do projecto.
RCS	Rich Communication Suite. Especificação que descreve um conjunto de funcionalidades e serviços em telecomunicações móveis.
RCS-e	Rich Communication Suite enhanced. Especificação do RCS.
RFC	Request For Comments. Nome dado à publicação de uma proposta para criação de (ou adenda a) um <i>standard</i> sobre uma determinada tecnologia ou protocolo de comunicação.
RTP	Real-time Transport Protocol. Protocolo de rede sobre IP que descreve o formato de pacotes áudio/vídeo para transmissão entre peers.
RTCP	Real-time Transport Control Protocol. Protocolo de rede sobre IP, paralelo ao RTP, para estatísticas e controlo de fluxo deste.
SCRUM	Metodologia ágil de desenvolvimento de software.
SDP	Session Description Protocol. Protocolo de rede sobre IP para descrever informação de <i>media</i> entre peers.
SIP	Session Initiation Protocol. Protocolo de rede sobre IP para sinalização e controlo de sessões entre peers
Sprint Backlog	Artefacto de metodologias ágeis de desenvolvimento, normalmente descrevendo o trabalho realizado num determinado período do projecto.
VoIP	Voice over IP. Grupo de tecnologias e/ou protocolos para transmissão de voz sobre a rede IP

1. Introdução

1.1. A Empresa

A empresa acolhedora do projecto de estágio chama-se **WIT Software, S.A.**, é portuguesa e foi fundada em 2001. É uma empresa de consultoria e desenvolvimento de aplicações e serviços, maioritariamente para o ramo das telecomunicações. Tem também projectos nas áreas da banca e TV.

1.2. WIT Communications Suite

O “produto” central da WIT Software é um sistema chamado **WCS – WIT Communications Suite** que permite e promove a comunicação entre utilizadores, através de chamadas de voz e vídeo, instant messaging, SMS e MMS, partilha de ficheiros multimédia, gestão de contactos e presença, entre outros. O WCS encontra-se disponível para PCs, telemóveis, tablets e browsers, tendo um servidor aplicacional de suporte, o WCAS.

Tecnicamente, os produtos do WCS inserem-se num conceito designado por **RCS – Rich Communication Suite**^[1] que descreve uma indústria focada no desenvolvimento e oferta de serviços multimédia sobre IP, em redes móveis. Esta especificação técnica é mantida por uma associação de operadores móveis, a GSMA, com o intuito de desenvolver *standards* e tecnologias para este tipo de redes.

Actualmente, o RCS desdobra-se em duas especificações, com as quais os clientes do WCS pretendem ser compatíveis – **RCS** (versão 2.0) e **RCS-e**. Estas especificações diferem essencialmente no conjunto de funcionalidades que os clientes devem oferecer. A especificação RCS-e, por exemplo, não contempla vídeo chamadas nem chamadas de voz sobre IP (VoIP), existindo apenas, neste caso, chamadas pela rede telefónica usual.

Recentemente, em Fevereiro de 2012, a WIT Software foi seleccionada^[2] pela GSMA para ser o fornecedor oficial de aplicações de referência RCS-e para Android e iPhone, que serão posteriormente distribuídas por intermédio de 5 operadores móveis: Orange, Vodafone, Deutsche Telekom, Telefonica e Telecom Italia. Por este motivo, ambos os clientes móveis viram a sua importância e visibilidade aumentar largamente, tendo provocado alterações a nível de planeamento e desenvolvimentos da equipa.

1.3. Metodologia de trabalho

A metodologia base de trabalho é **SCRUM**^[3]. Sendo uma ferramenta de desenvolvimento ágil e iterativa, permitirá um controlo mais adequado e mais acompanhado do trabalho desenvolvido. O estagiário começará por elaborar um Product Backlog inicial – reproduzido no Anexo A –, onde são listados os objectivos de desenvolvimento e as várias tarefas que permitem atingir esses objectivos. São também indicadas estimativas temporais previstas para atingir cada um desses objectivos.

De duas em duas semanas, são elaborados Sprint Backlogs – presentes no mesmo Anexo –, que listam com maior detalhe o que se prevê desenvolver nessas duas semanas, sob a forma de tarefas, as estimativas temporais e o esforço efectuado, no final desse período.

Será feita, sempre que possível, um mínimo de uma reunião por mês, com todas as equipas envolvidas nos produtos, onde é apresentado o progresso dos desenvolvimentos e, quando aplicável, demonstrações práticas das funcionalidades implementadas ou do seu estado actual.

1.4. Motivação

A principal motivação para a criação do presente projecto de estágio prende-se com a necessidade de melhorar um aspecto dos clientes WCS que tem sido descurado: as chamadas. Sendo que o foco da equipa de desenvolvimento do WCS se tem centrado em ter os clientes o mais de acordo possível com as especificações de serviços RCS e RCS-e, tal factor repercutiu-se nas funcionalidades de chamadas de voz e de vídeo ficarem deficientes ou incompletas.

O presente projecto de estágio pretende complementar o WCS, através de melhoramentos na qualidade das chamadas. Esta qualidade não só diz respeito a aspectos mais directos (áudio-visuais), que são inerentes às chamadas – som mais perceptível, imagem mais detalhada e real –, mas também a nível de interactividade, melhorando a experiência de utilização.

2. Estado da Arte

Este capítulo pretende elucidar o leitor sobre o estado actual do produto onde se insere o presente projecto de estágio, bem como de outras ofertas semelhantes e concorrentes, actualmente no mercado. Através dessa informação, é possível averiguar a necessidade e importância de se investir no tema proposto da alta definição para voz e vídeo.

Segundo um estudo^[4] realizado em Julho de 2011, pela Allot Communications, observou-se um aumento de tráfego de dados relativos a voz sobre IP e partilha de vídeo, sendo as aplicações que oferecem este último ponto as mais utilizadas (ver Figura 1).

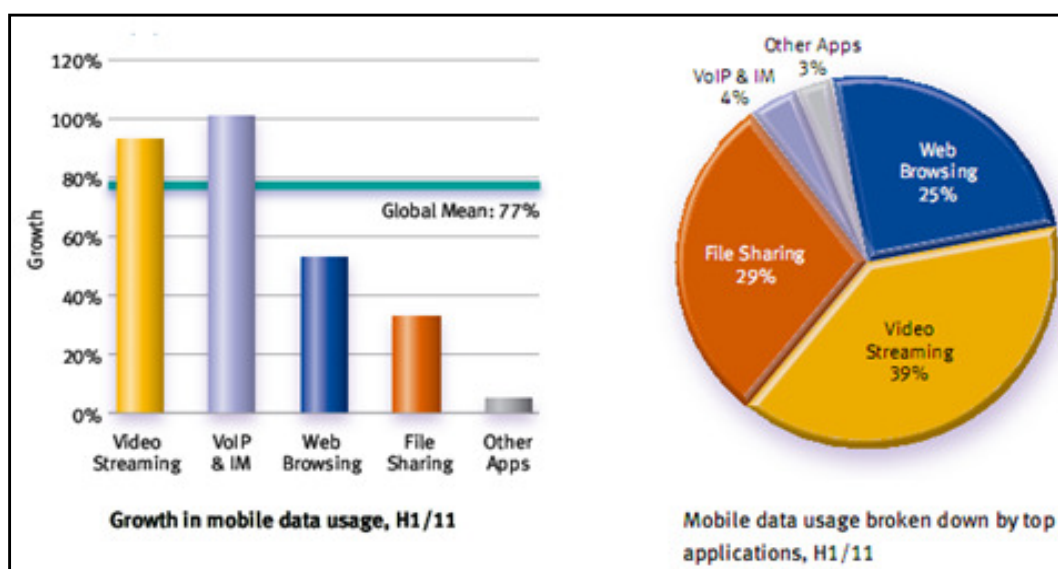


Figura 1 – Crescimento de aplicações, por funcionalidade

O mesmo estudo apresenta ainda a tendência no uso de várias das funcionalidades disponibilizadas pelas aplicações, ao longo dos vários trimestres, sendo de novo claro que são as funcionalidades que incluem vídeo que apresentam o maior crescimento. Pelo Gráfico 1, podemos igualmente ver que, apesar da utilização de funcionalidades que incluem voz ser relativamente baixa quando comparada com

as restantes, demonstra em contraste que a sua utilização é mais constante ao longo do tempo, e portanto igualmente importante.

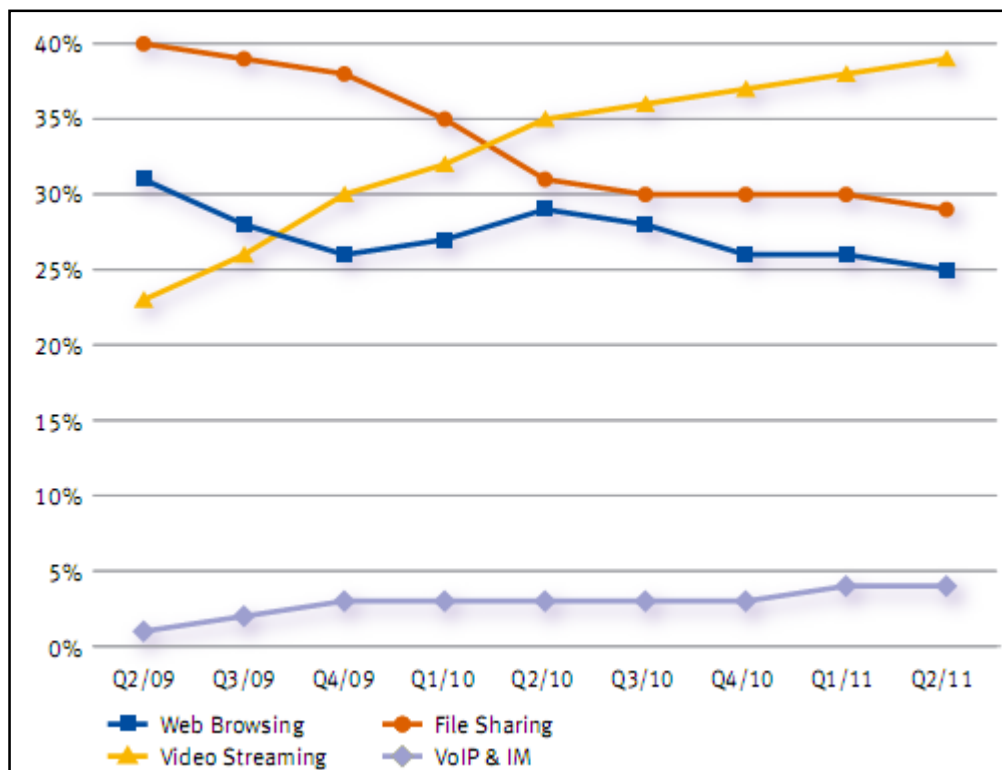


Gráfico 1 – Tendência de circulação de dados, por funcionalidade

Algumas das razões que o estudo aponta para explicar este crescimento incluem o facto de cada vez mais os dispositivos físicos usados na comunicação permitirem o uso de câmaras frontais – enriquecendo assim, por exemplo, chamadas de vídeo – e ainda ser cada vez mais fácil comunicar entre diferentes dispositivos – telemóveis, aplicações para PCs, páginas Web, *tablets*, etc.

No entanto, existe também um factor fundamental que pode explicar o porquê deste crescimento e deste interesse na voz e no vídeo. Prende-se com o facto de existir cada vez mais um aumento na qualidade de som e imagem oferecidas. A experiência de utilização é algo que provoca no utilizador o interesse em querer usar e continuar a usar determinada aplicação. Uma aplicação que transmita som com cortes ou interrupções, ou com uma qualidade baixa que impeça o utilizador de perceber o que a outra pessoa diz poderá não ser tão provável que volte a ser utilizada. Da

mesma forma, uma aplicação que transmita vídeo com baixa qualidade, com artefactos na imagem, pixelizada ou desfocada, ou com cores alteradas, não terão o impacto desejado junto do utilizador.

2.1. Competidores

Existem inúmeras soluções e aplicações que oferecem experiências de utilização completas e de qualidade. Produtos como AOL Instant Messenger, eyeBeam, Google Talk, Yahoo Messenger, etc., facilmente permitem aos utilizadores a possibilidade de comunicação através da voz e vídeo, mensagens de texto, partilha de ficheiros multimédia e inclusivamente integrações com outros serviços, como por exemplo serviços de e-mail. No entanto, nem

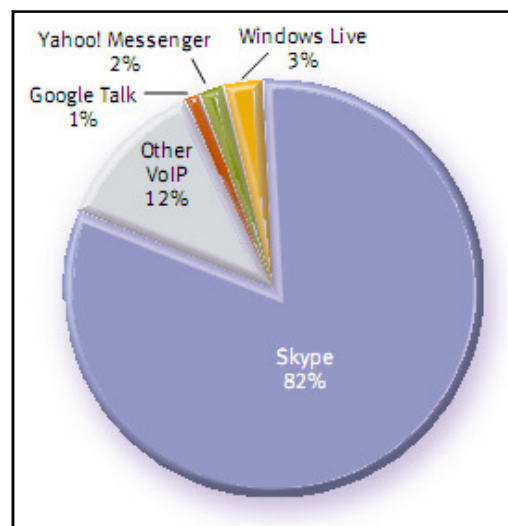


Figura 2 – Utilização de aplicações

todos conseguem ainda atingir um patamar de qualidade sonora e visual que é muitas vezes desejada pelos utilizadores. É aqui que surgem soluções como o **Skype** que, também como o referido estudo indica, continua a ser a aplicação mais utilizada.

O Skype é uma aplicação de VoIP e Video over IP que, para além de todas as funcionalidades habituais de *instant messaging*, oferece nas suas chamadas de voz e vídeo uma elevada qualidade e desempenho, sendo por isso um dos produtos de comunicação mais usados no mundo. Para atingir este patamar, o Skype usa codecs de áudio designados como *wideband* (como o SILK) e codecs de vídeo de última geração (como o H.264 ou VP8).

O Skype, no entanto, é uma aplicação bastante mais complexa, e vai muito para além das especificações RCS, que são o objectivo principal do produto WCS. Assim sendo, o WCS compete directamente com outras soluções RCS/RCS-e actualmente

em comercialização por todo o mundo. Existe uma vasta lista, cada um com a sua visibilidade, da qual se destacam a SummitTech, Huawei, Movial e Jibe Mobile, todas elas distribuindo o mesmo tipo de funcionalidades às operadoras móveis, e procurando que os seus serviços e soluções sejam as referências no mercado.

2.2. Áudio

Um dos aspectos determinantes para obter melhores resultados a nível de som e imagem centra-se nos já referenciados codecs. Este tipo de software permite comprimir os dados áudio e/ou vídeo para depois serem armazenados ou transmitidos sobre a rede.

Existem inúmeros codecs áudio com diferentes características, sendo classificados como *narrowband* ou *wideband*. Existe ainda a categoria *ultra-wideband* embora, de uma maneira geral, estes estejam pensados para uma utilização diferente, nomeadamente comunicações de curto alcance como periféricos de PCs. A maioria dos codecs é do tipo *narrowband*; esta designação indica que o espectro de som disponível para transmissão não é muito abrangente (rondando os 300Hz a 3.4kHz), e é muitas vezes limitado pelos próprios meios físicos usados, como é o caso da rede telefónica. Tais limitações resultam invariavelmente numa qualidade de som insuficiente e que, para o utilizador, tornam a experiência do seu uso desagradável.

É aqui que surge a necessidade de se usar os codecs de gama superior, designados como *wideband*. Em contraste com os anteriores, estes permitem alargar o espectro de sinal para 50Hz a 7kHz, valores que se aproximam mais que os anteriores do diferencial da voz humana (80Hz a 14kHz). Aliando o uso destes codecs em meios diferentes de comunicação, como por exemplo sobre a rede de internet, pode melhorar significativamente a qualidade do som.

Para além da frequência usada, é importante também considerar parâmetros como a variação de bitrate disponível – permitindo a mesma qualidade, independentemente

da dimensão ou frequência dos dados –, o tamanho de cada amostra de sinal e, até mesmo o tipo de licença de utilização permitido.

No Anexo B, é apresentada uma tabela elucidativa da actual oferta de codecs existentes nos vários produtos do WCS. Grande parte deles é do tipo *narrowband* ou requerem bitrates demasiado elevados e constantes (como 64kbits) para o meio físico usado, resultando numa qualidade de som que pode não ser inteiramente desejável.

Ainda nessa oferta, podemos observar dois codecs que oferecem uma qualidade potencialmente melhor, como são os casos do G.722.1 (ou AMR-WB) e do Speex. Estes dois codecs podem operar a 16kHz (também a 32kHz no caso do Speex), com abrangente diferencial de bitrate – na ordem dos 20kbits no primeiro caso, e 40kbits no segundo. No Anexo C, é feito um estudo comparativo dos codecs áudio suportados pelos clientes do WCS, através do método de avaliação chamado **MOS – Mean Opinion Score**^[5].

Um codec áudio emergente nos últimos anos chama-se **SILK**^[6], e foi desenvolvido pela Skype Limited, empresa criadora do Skype. Este codec foi recentemente disponibilizado para uso livre por outras aplicações, já tendo sido incorporado, por exemplo, em várias plataformas de *online gaming*, para comunicações entre jogadores^[7]. É capaz de transmitir a uma frequência de 8, 16 e 24kHz, com um bitrate variável entre 8 a 30kbit/s. Para além disso, é de uso livre, acabando por ser uma escolha promissora, tendo em conta também que é o codec usado pelo próprio Skype.

Outros codecs também designados *wideband* e que poderão de igual forma permitir uma qualidade de som superior são, entre outros, **AMR-WB+** e **G.722.1C**.

2.3. Vídeo

O vídeo é também um aspecto muito importante quando se fala em comunicações, pois o impacto causado junto do utilizador é também imediato como o áudio. É um meio de transmissão de informação extremamente visual e interactivo, e por isso é importante que a sua qualidade não seja também ela descurada.

Os produtos actuais do WCS já oferecem algumas funcionalidades de transmissão de vídeo, no entanto com algumas deficiências: visualmente, nota-se com facilidade a existência de “artefactos” (ou “quadriculado”) nas imagens, tornando o vídeo em alguns casos imperceptível e de fraca qualidade. O Anexo E contém algumas imagens que demonstram essa qualidade. A razão para tal acontecer deve-se ao facto das capacidades do codec de vídeo que está a ser usado, neste caso o H.263, serem insuficientes e inadequadas.

Assim, há uma necessidade paralela de se actualizar também a oferta de codecs de vídeo no WCS, adicionando-se *standards* mais actuais e que potenciem significativas melhorias visuais. O codec vídeo mais utilizado nos dias de hoje é o **H.264**, e é usado não só em telecomunicações mas também em *broadcasting* – por exemplo serviços de alta-definição de TV – e armazenamento – como os discos Blu-ray.

O H.264 apresenta-se como sendo bastante mais flexível que os seus antecessores e com um algoritmo preditivo mais configurável. Permite, por exemplo, usar 16 frames como referência na codificação preditiva das frames seguintes – ao contrário do seu antecessor H.263, que só permitia um máximo de 2 frames de referência. Permite também que a divisão da imagem em macro-blocos seja de tamanho variável – diversas combinações de 4x4, 8x8, 16x16, 8x4, etc, oferecendo uma melhor sub-divisão das frames, e inerentemente maior detalhe. É possível também codificar os planos de cor individualmente tendo, cada um, as suas próprias

estruturas de dados (macro-blocos, vectores de movimento, etc). No final, obtém-se uma clara melhoria de imagem, quando comparado com outros codecs.

A Figura 3 ilustra o resultado que se obtém ao se usar uma das funcionalidades existentes no *standard* H.264: o *deblocking filter*. O uso dos chamados macro-blocos acaba por causar o efeito de se notarem as suas “bordas”, como acontece no H.263. O *deblocking filter* permite reduzir este efeito.



Figura 3 – Efeito do *deblocking filter* em H.264

Um estudo^[8] introdutório da Universidade de Arizona demonstra que o H.264 é capaz de produzir a mesma – ou ainda melhor – qualidade até um bitrate de quase metade de outros codecs. Nos Gráficos 2 e 3 é possível ver uma comparação feita entre o H.264 e os codecs MPEG-4, MPEG-2 e H.263, onde se mostra que, a bitrates muito inferiores, o ratio signal-to-noise obtido (que traduz, no fundo, o grau de fidelidade de compressões *lossy*), para duas resoluções diferentes (176x144 e 352x288) é maior, descrevendo uma qualidade de imagem superior.

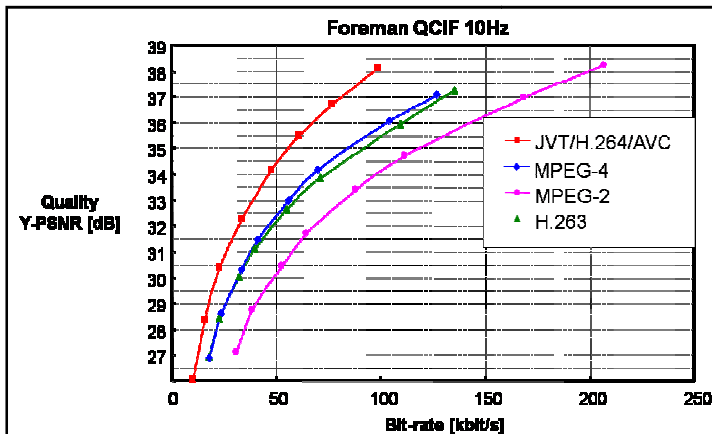


Gráfico 2 – Signal-to-noise ratio a 10Hz, 176x144

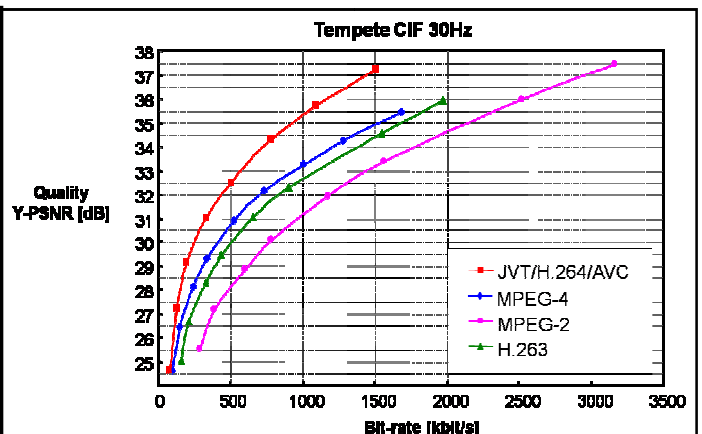


Gráfico 3 – Signal-to-noise ratio a 30Hz, 352x288

Um outro codec emergente, embora ainda não se vejam muitas aplicações para além do Skype que o usem, é o **VP8**^[9]. Recentemente, a Google anunciou que os seus produtos iriam passar a suportar também este formato, nomeadamente o browser Chrome, e no futuro o seu sistema operativo Android. Daí que este codec seja também uma solução a considerar, embora com menor prioridade. O facto de se disponibilizar diferentes soluções permite também aos clientes WCS poderem ser inter-operáveis com outras aplicações no mercado, tornando-se numa mais-valia para o produto.

2.4. 4G e LTE

Desde há cerca de 3 décadas que o ramo das comunicações tem vindo a evoluir, ao ritmo de uma grande actualização a cada 10 anos. Desde a telefonia analógica de 1ª Geração (1981/1982) até aos dias de hoje, se tem observado um crescendo na oferta de serviços de maior qualidade a cada vez mais utilizadores, por diferentes meios e a maiores velocidades. Até há pouco tempo, esta tecnologia situava-se na chamada 3G/3.5G (3ª Geração). Esta geração descreve um conjunto de standards que permitem a comunicação de voz, vídeo e internet em ambientes sem fios, despoletando também o acesso a serviços que até então não estavam tão disponíveis em clientes móveis, como por exemplo televisão (mobile TV). Estes standards declaravam um limite mínimo de 21.6Mbit/s de download e 5.8Mbit/s de upload de dados.

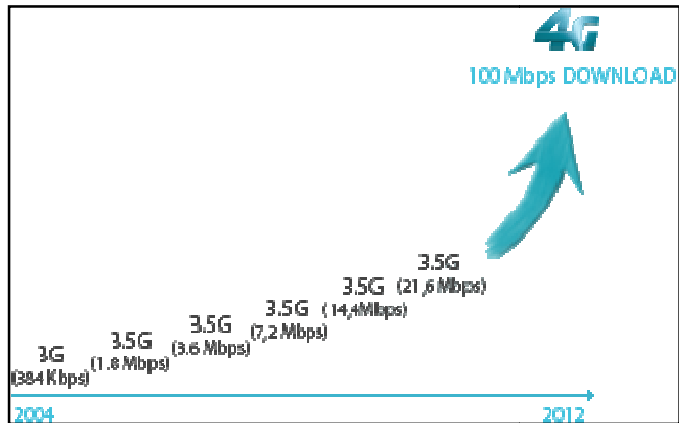


Figura 4 – Evolução da velocidade móvel

A evolução natural da tecnologia originou recentemente a publicação da chamada 4G, aumentando os limites teóricos de transmissão de dados para 100Mbit/s (alta mobilidade dos dispositivos) e 1Gbit/s (baixa mobilidade). Ou seja, comparando com a anterior geração, dispositivos que usem serviços por 4G conseguem transmitir e receber dados cerca de 5x mais rapidamente.

O sistema 4G tem como principais objectivos permitir que toda a oferta de serviços seja baseada em protocolos IP, a altas velocidades, a frequências mais reduzidas, de forma mais escalável e dinâmica, para permitir a existência de mais utilizadores por área de cobertura^[10]. A “implementação” desta tecnologia tem o nome de LTE – Long Term Evolution. Actualmente, já existem versões comerciais do LTE em alguns países, embora em formato experimental, não tendo ainda os débitos publicitados pelo *standard*, como são os casos da Noruega e Suécia – fornecido pela TeliaSonera –, e os Estados Unidos da América – fornecida pela MetroPCS. Estima-se que durante 2012/2013, cerca de 55 redes móveis um pouco por todo o mundo disponibilizem a 4G^[11].

A entidade reguladora da tecnologia por detrás do RCS (a GSMA) tem vindo gradualmente a actualizar as suas especificações deste serviço, para ir ao encontro do LTE, estando previsto para 2012 o lançamento da nova versão oficial – o RCS 5.0. Como já tem sido habitual, é objectivo fundamental do WCS, enquanto produto, de ir ao encontro desta especificação e de disponibilizar clientes que a suportem, justificando-se assim um interesse em contemplar também desenvolvimentos nesse sentido, no projecto de estágio.

3. **HD Voice & HD Video**

O presente capítulo tem como objectivo fazer uma descrição de conhecimentos introdutórios que servem de base aos Anexos D, E, F, G, e H. Cada um destes anexos descreve com mais detalhe um módulo de desenvolvimentos do projecto de estágio. Neste capítulo abordam-se os requisitos de mais alto nível de todo o estágio e a arquitectura geral dos clientes do WCS, sob a forma de camadas, seguindo-se uma breve explicação de cada uma.

3.1. **Requisitos**

Os requisitos do estágio pretendem responder à necessidade base de melhorar a qualidade global das chamadas de voz e vídeo dos produtos do WCS. Tendo em conta o seu estado actual, melhorar a qualidade global das chamadas descreve-se através dos seguintes objectivos principais:

- Aumentar a oferta de codecs de áudio dos produtos, adicionando determinados codecs do tipo *wideband*, capazes de transmitir em frequências de sinal mais elevadas. Os codecs a adicionar serão, por esta ordem de prioridade e necessidade, o **SILK**, o **G.722.1C** e o **AMR-WB+**;
- Aumentar a oferta de codecs de vídeo dos produtos, adicionando codecs *standard* de última geração, capazes de transmitir em alta definição e/ou resoluções elevadas. Os codecs a adicionar serão, por esta ordem de prioridade e necessidade, o **H.264** e o **VP8**. Uma vez que o cliente PC já dispõe de uma implementação do codec H.264, este será apenas adicionado aos clientes móveis (Android e iPhone). Em termos de *user interface*, apenas são contemplados desenvolvimentos no cliente Android;
- Adicionar partilha de vídeo *live*, em chamadas de voz, e também a partilha de ficheiros multimédia que contenham vídeo, nessas chamadas. A partilha

de vídeo está dependente da implementação dos codecs vídeo dos pontos anteriores e deve estar de acordo com a especificação RCS-e mais actual;

- Fazer uma implementação inicial de voz sobre LTE, como preparação do cliente móvel do WCS para a 4G, de acordo com a especificação RCS 5.0.

Para além dos descritos, consideram-se como objectivos secundários e/ou opcionais o seguinte:

- Adicionar a apresentação de informação referente à qualidade relativa das chamadas num dado instante. Traduz-se, especificamente, em notificar o utilizador se o microfone e/ou as colunas estão desligadas, a funcionar incorrectamente, se o som se encontra demasiado baixo, etc.

Uma vez que o WCS engloba diversos clientes, é necessário determinar também quais deles precisam de responder aos objectivos enumerados anteriores. A Tabela 1 faz o mapeamento dos requisitos que devem ser implementados por cliente:

Tabela 1 – Requisitos por cliente WCS

#	Requisitos	WPC	WMC – Android	WMC – iPhone
1	Aumentar a oferta de codecs áudio			
1.1	SILK	✓	✓	✓
1.2	G.7221.C	✓		
1.3	AMR-WB+	✓		
2	Aumentar a oferta de codecs vídeo			
2.1	H.264		✓	✓
2.2	VP8	✓	✓	✓
3	Partilha de vídeo			
3.1	Partilha de vídeo <i>live</i> (em H.264)		✓	✓
3.2	Partilha de ficheiros vídeo pré-gravados (em H.263 ou H.264)		✓	✓
4	4G / LTE			
4.1	Funcionalidades de voz sobre LTE – RCS 5.0		✓	✓
#	Outros requisitos	WPC	WMC – Android	WMC – iPhone
1	Notificações			
1.1	Adicionar notificações indicativas do estado actual dos equipamentos de som	✓		

3.2. Arquitectura

A arquitectura dos produtos é bastante extensa e grande parte desta está fora do espectro do projecto de estágio. Neste capítulo ilustram-se e descrevem-se os pontos mais relevantes da arquitectura do WCS já existente, e indica-se onde e de que forma se inserem os novos desenvolvimentos.

3.2.1. Overview

Como referido anteriormente, o WCS dispõe de um servidor (WIT Communications Application Server) ao qual se ligam os clientes. Este servidor faz toda a gestão de sessões entre utilizadores, que incluem chamadas, sessões de *instant messaging*, SMS/MMS, transferências de ficheiros, etc. Este sistema foi pensado de modo a poder integrar-se com diversos outros sub-sistemas, como servidores de e-mail, gateways para *media*, sistemas de autenticação, billing, etc. A Figura 5 ilustra de uma forma geral como os clientes, o servidor e os sub-sistemas se interligam.

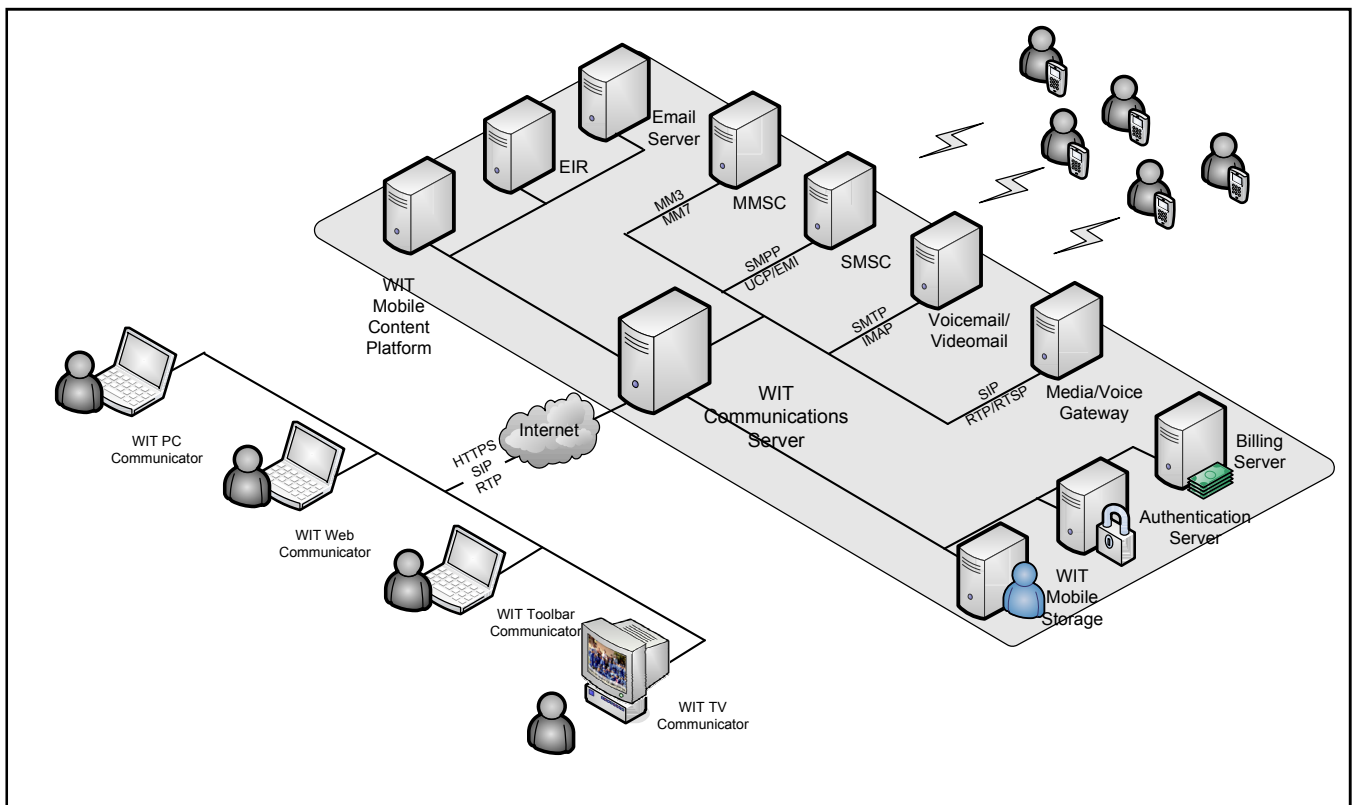


Figura 5 – Arquitectura global do WCS

Os clientes do WCS comunicam com o servidor preferencialmente através de UDP e usando os protocolos SIP (para sinalização), RTP/RTCP (para transmissões voz/vídeo e controlo de fluxo) e HTTP/HTTPS (para as restantes comunicações, como gestão de contactos, perfil de utilizador, etc.). Quando tal não for possível, é usado TCP com encapsulamento do tráfego.

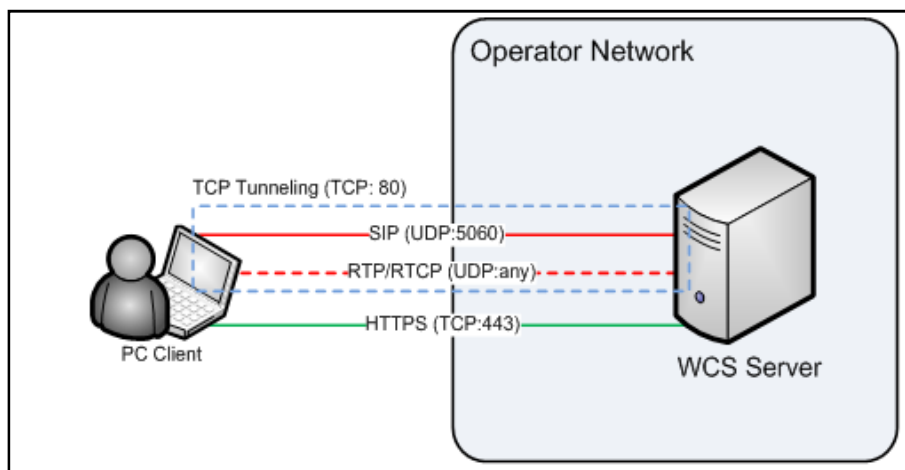


Figura 6 – Protocolos de comunicação entre clientes e servidor

3.2.2. Camadas de aplicação

A arquitectura geral dos clientes PC e móvel (Android / iPhone) do WCS encontra-se ilustrada na Figura 7.

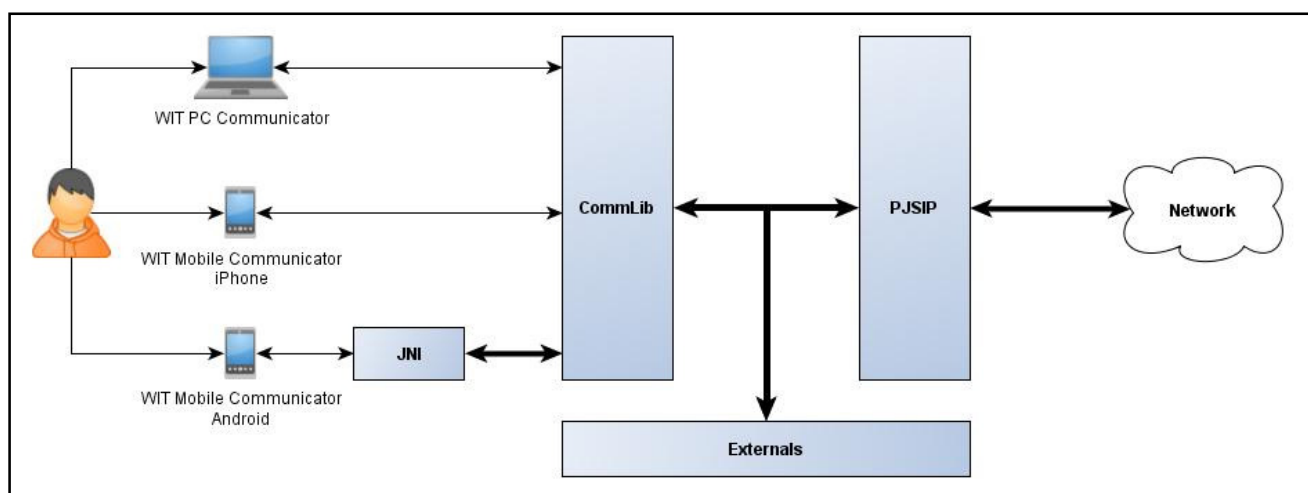


Figura 7 – Representação dos clientes WCS por camadas de aplicação

Através da figura, podem ser observados os módulos, ou camadas, que constituem os clientes, sendo eles a camada de UI (por cliente), CommLib, bibliotecas externas (designadas por *Externals*) e PJSIP, que funciona como camada de transporte e conexão com a rede. À excepção da camada de UI do cliente Android, todas elas usam a linguagem C/C++ ou variantes. No caso do cliente Android, a camada de UI usa a linguagem Java e por isso é necessária a existência de uma interface de integração Java – C, denominada na figura por *JNI*. A camada de UI da versão iPhone do cliente móvel não está contemplada no projecto de estágio.

A Tabela 2 faz o mapeamento entre os requisitos enumerados no ponto 3.1 e as camadas da arquitectura do produto onde serão feitos os desenvolvimentos:

Tabela 2 – Requisitos por camada de aplicação

#	Requisitos	UI	CommLib	PJSIP	Externals
1	Aumentar a oferta de codecs áudio				
1.1	SILK			✓	✓
1.2	G.7221.C			✓	✓
1.3	AMR-WB+			✓	✓
2	Aumentar a oferta de codecs vídeo				
2.1	H.264			✓	✓
2.2	VP8			✓	✓
3	Partilha de vídeo				
3.1	Partilha de vídeo <i>live</i> (em H.264)	✓			
3.2	Partilha de ficheiros vídeo pré-gravados (em H.263 ou H.264)	✓	✓	✓	✓
4	4G / LTE				
4.1	Funcionalidades de voz sobre LTE – RCS 5.0			✓	✓
#	Outros requisitos	UI	CommLib	PJSIP	Externals
1	Notificações				
1.1	Adicionar notificações indicativas do estado actual dos equipamentos de som	✓	✓		

3.2.2.1. UI – Cliente Android

Uma aplicação Android é construída por intermédio de vários elementos chamados *Activity*^[12]. Estas “actividades” representam um ecrã com determinadas funcionalidades, com o qual o utilizador pode interagir, e que se podem inicializar umas às outras. Por exemplo, ao se iniciar a aplicação Android, é criada uma *Activity*, considerada a “principal”. O utilizador, ao interagir com a aplicação, faz com que essa *Activity* possa iniciar outra *Activity*, com outras funcionalidades, e assim sucessivamente. Na Figura 8, está representado o ciclo de vida de uma *Activity*.

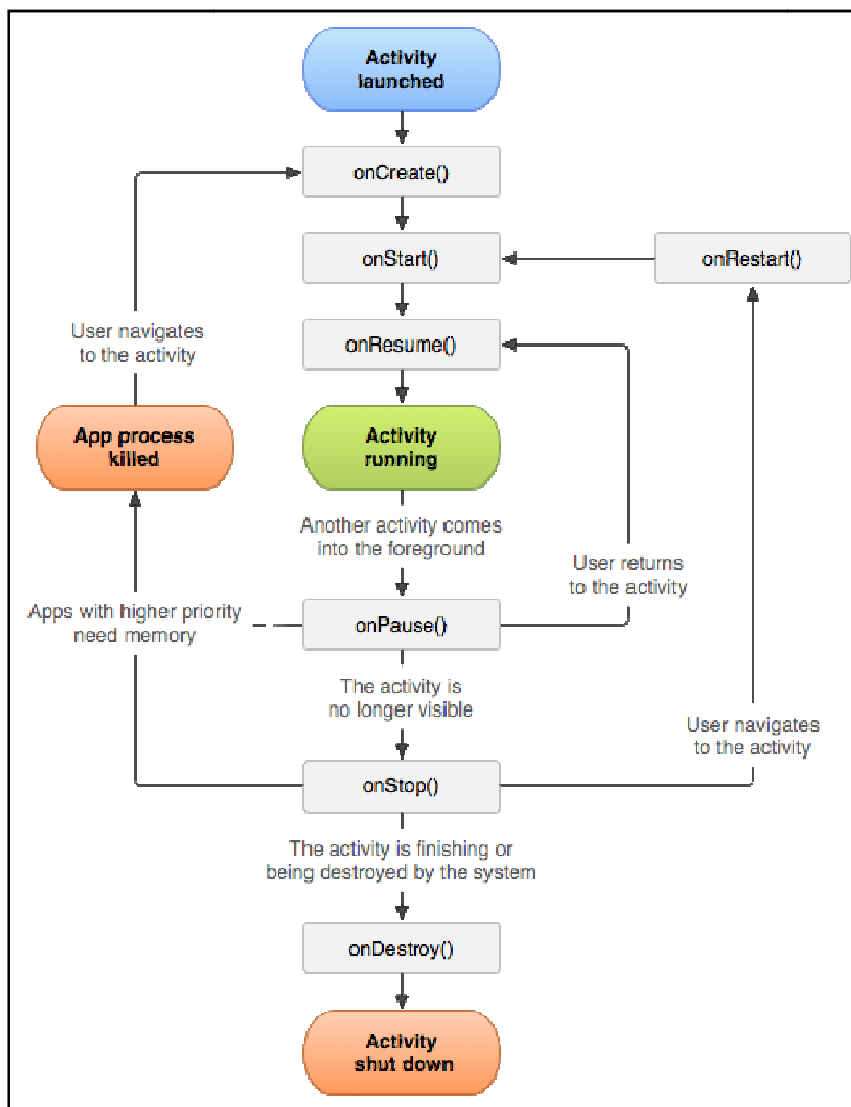


Figura 8 – Ciclo de vida de uma *Activity*, em Android

O sistema operativo Android faz a gestão das Activities sob a forma de uma pilha LIFO (Last In, First Out)^[13]. De cada vez que uma Activity é substituída por outra – por exemplo, devido a uma interacção do utilizador – essa Activity é colocada no topo da pilha. Quando a nova Activity termina – por exemplo, o utilizador clicou no botão “Back” do telemóvel, ou a Activity terminou o seu processamento –, esta é destruída e é retomada a Activity anterior, a partir do ponto onde foi deixada quando colocada na pilha.

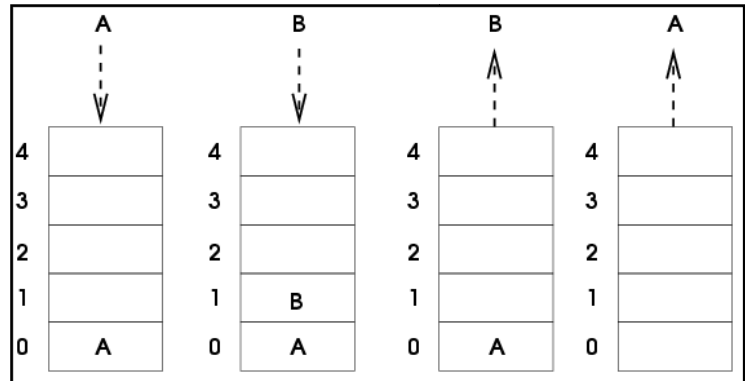


Figura 9 – Pilha LIFO de Activities, utilizada pelo Android

Sendo no fundo a representação de um ecrã, cada Activity tem associado um *layout* geral, podendo este ter um ou mais *layouts* internos, e de um conjunto de elementos do tipo *View*, dispostos nesses *layouts*. Uma *View*, ou vista, representa um elemento básico de formulário, como por exemplo uma caixa de texto, uma lista de elementos, uma *label*, ou uma superfície para desenho de imagens, etc. Na Figura 10, está ilustrado um exemplo deste tipo de hierarquia^[14].

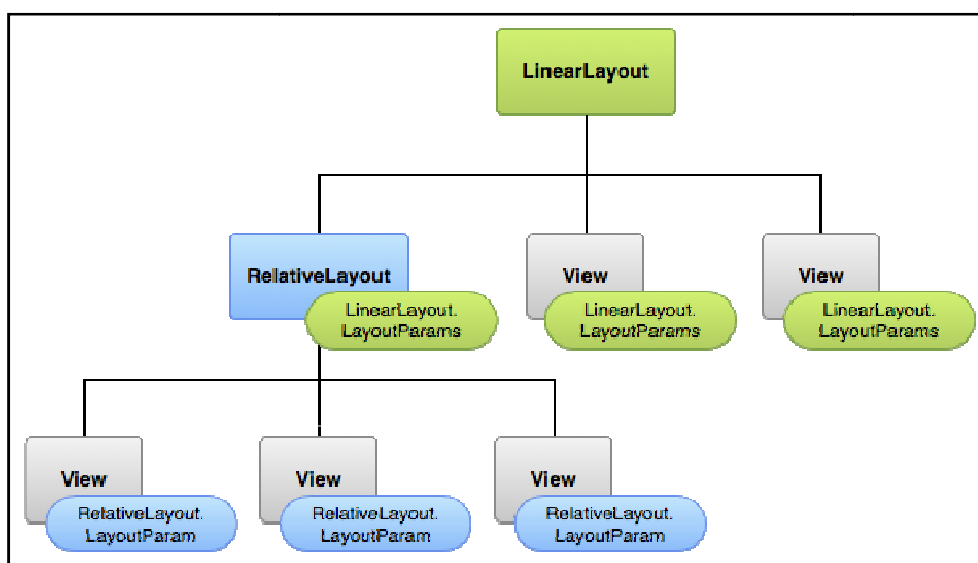


Figura 10 – Hierarquia de layouts e Views, no Android

Ainda nas vistas, existe um outro elemento, denominado *Dialog*, que representa um diálogo ou janela mais pequena e que pode ser sobreposta a toda a Activity. Este Dialog é, normalmente, uma caixa com uma mensagem de alerta, uma janela de progresso (por exemplo, uma transferência), ou até mesmo um menu temporário de opções^[15].

No âmbito do cliente Android do WCS, o início das Activities é feito por intermédio de uma classe *manager*, correspondente à funcionalidade associada à Activity. Como ilustra o diagrama de classes da Figura 11, existem vários *managers*, responsáveis pela gestão, por exemplo, de chamadas, serviços em chamada de voz (partilha de vídeo, imagem, localização), transferências de ficheiros, conversas de *instant messaging*, histórico, perfil de utilizador, contactos, etc. No caso das chamadas, a classe **CallManager** é a responsável pelo início de Activities relativas a chamadas bidireccionais – voz e vídeo – e a classe **CallServiceManager** é a responsável pelo início de Activities relativas a serviços numa chamada de voz.

Na Figura 12, está ilustrada a sequência de interações entre a camada de UI (por intermédio dos *managers*) com a CommLib (através da camada JNI), tomando como exemplo o início de uma **CallActivity** (ecrã de uma chamada). Como descrito anteriormente, independentemente da Activity actualmente a executar (designada na Figura por *Activity X*), esta é substituída por uma CallActivity, quando chega um pedido de chamada, vinda da rede para as camadas PJSIP e CommLib. Na Figura 13, é ilustrada ainda a forma como essa CallActivity é terminada e o Android retoma a Activity anterior (*Activity X*). Neste caso, esta acção pode ser provocada, por exemplo, pela terminação explícita da chamada, através de uma opção no ecrã. O clicar do botão “Back” (ou “Home”) do telemóvel faz com que essa CallActivity não seja terminada – sendo colocada em *standby*, na pilha do Android – e a Activity X seja retomada.

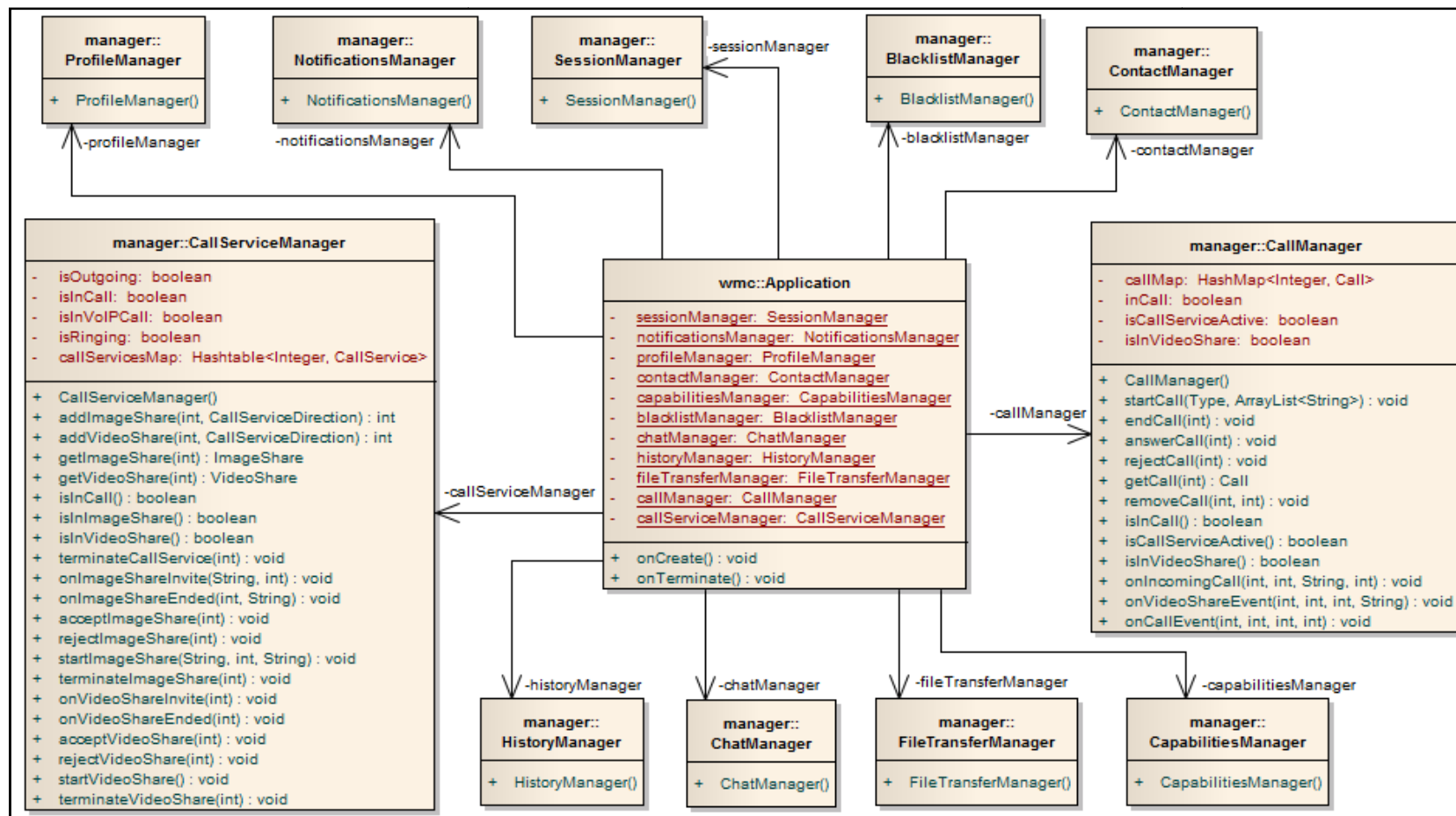


Figura 11 – Diagrama de classes – managers (WMC-Android)

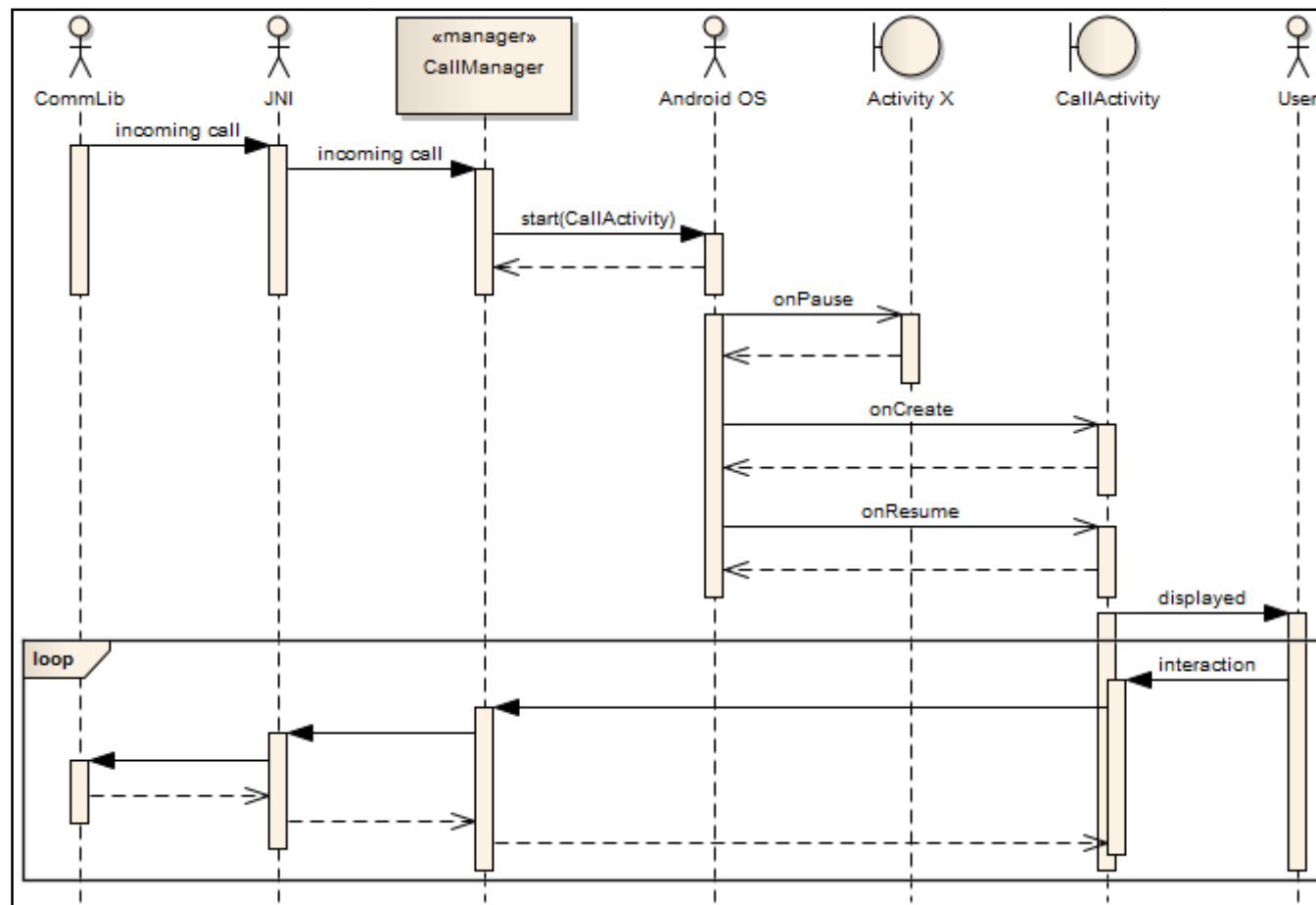


Figura 12 – Diagrama de sequência – *Activities* e chamada (WMC-Android)

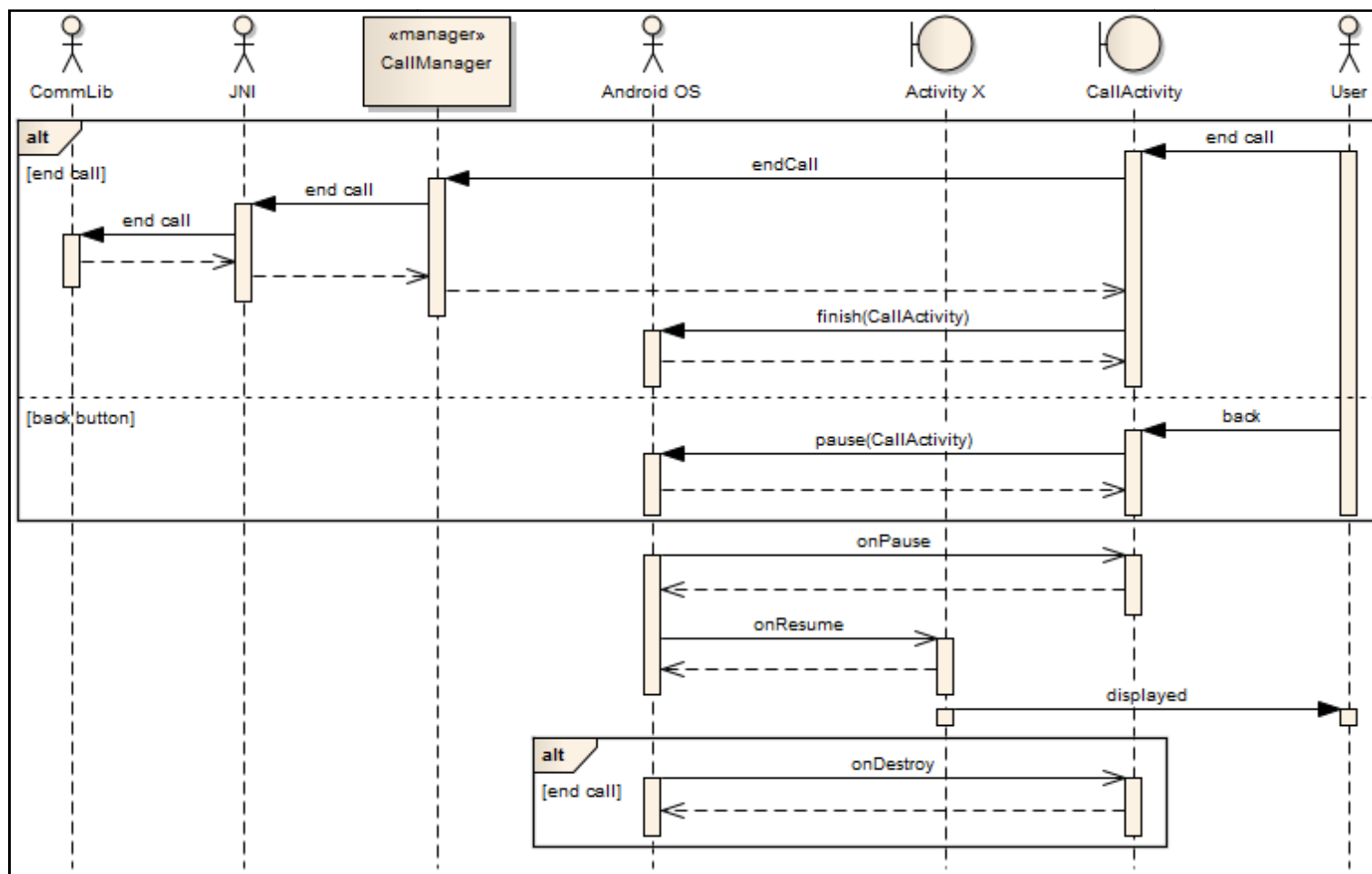


Figura 13 – Diagrama de sequência – *Activities* e fim de chamada (WMC-Android)

3.2.2.2. **CommLib**

A CommLib é um dos pontos centrais dos clientes WCS. Esta camada foi implementada de forma a poder ser integrada em qualquer cliente actual ou futuro do WCS, ou de outro projecto com necessidades semelhantes, sendo assim apenas necessários desenvolvimentos a nível das camadas de apresentação.

As principais funções desta são a gestão de eventos e dados/objectos durante a execução das aplicações e ainda armazenamento persistente (base de dados em ficheiros). A gestão de dados/objectos é feita por intermédio de classes designadas por *managers*, semelhante à camada de UI.

Os eventos gerados permitem a comunicação entre a *user interface* e a camada de transporte (PJSIP). Por exemplo, quando uma chamada chega à camada de transporte, vinda da rede, é gerada uma notificação na CommLib que, por sua vez, origina a criação de um determinado evento. Este evento traduz-se posteriormente, na camada da UI, sob a forma de uma janela com controlos para atender ou rejeitar essa chamada.

Nas Figuras 14 e 15, está representado o aspecto geral da CommLib. Existe uma classe principal, **Communicator**, responsável pela gestão e configuração de cada *manager*. Estes managers são tratados como sendo um “serviço”, que pode ser activado ou desactivado em qualquer instante. Ainda na Figura 15, é possível ver que alguns desses *managers* contêm uma referência para uma classe UserAgent. Esta classe representa o *interface* entre a CommLib e o PJSIP, sendo através desta que são feitas e recebidas as interacções com a camada de transporte.

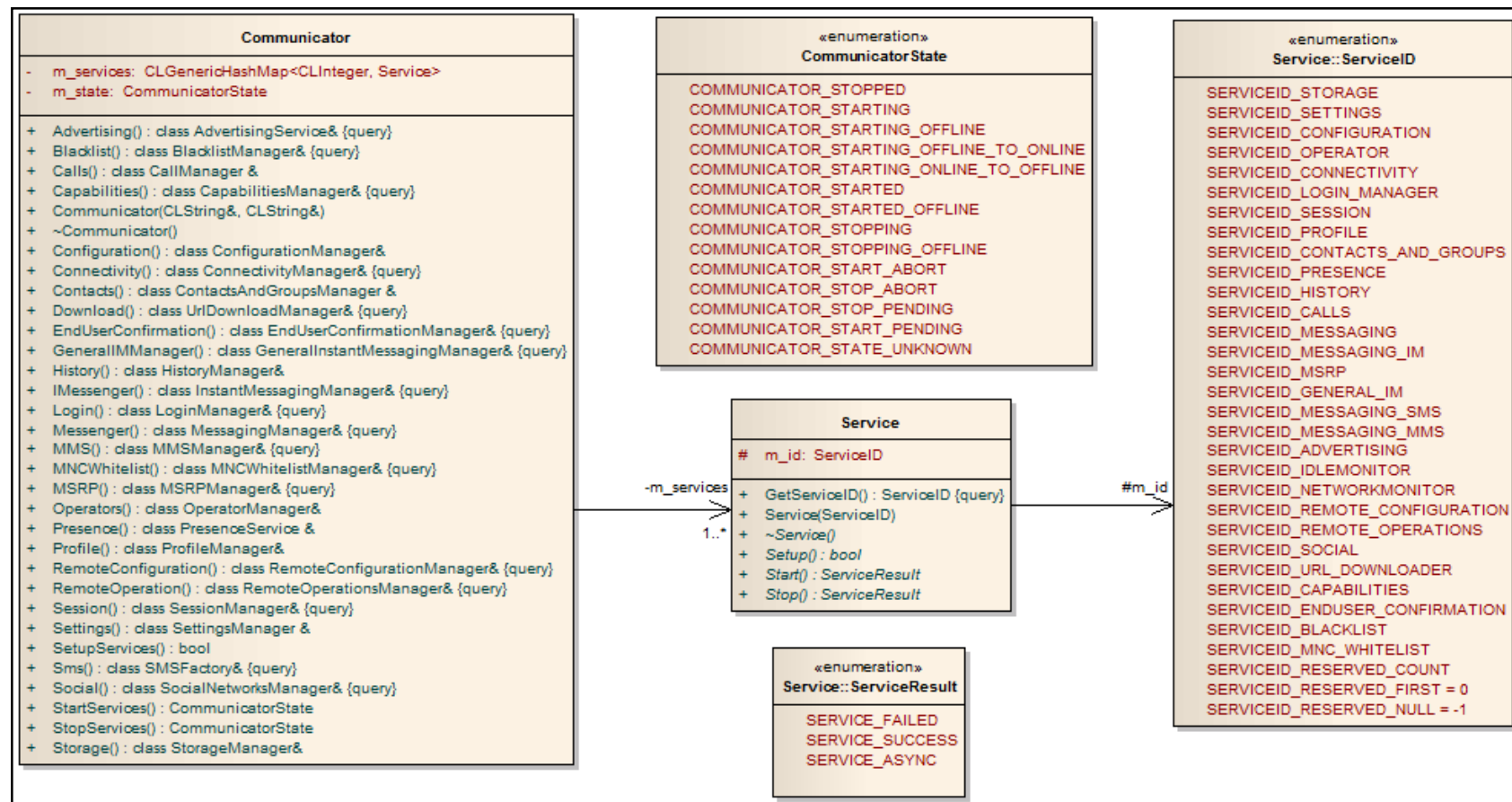


Figura 14 – Diagrama de classes – Communicator (CommLib)

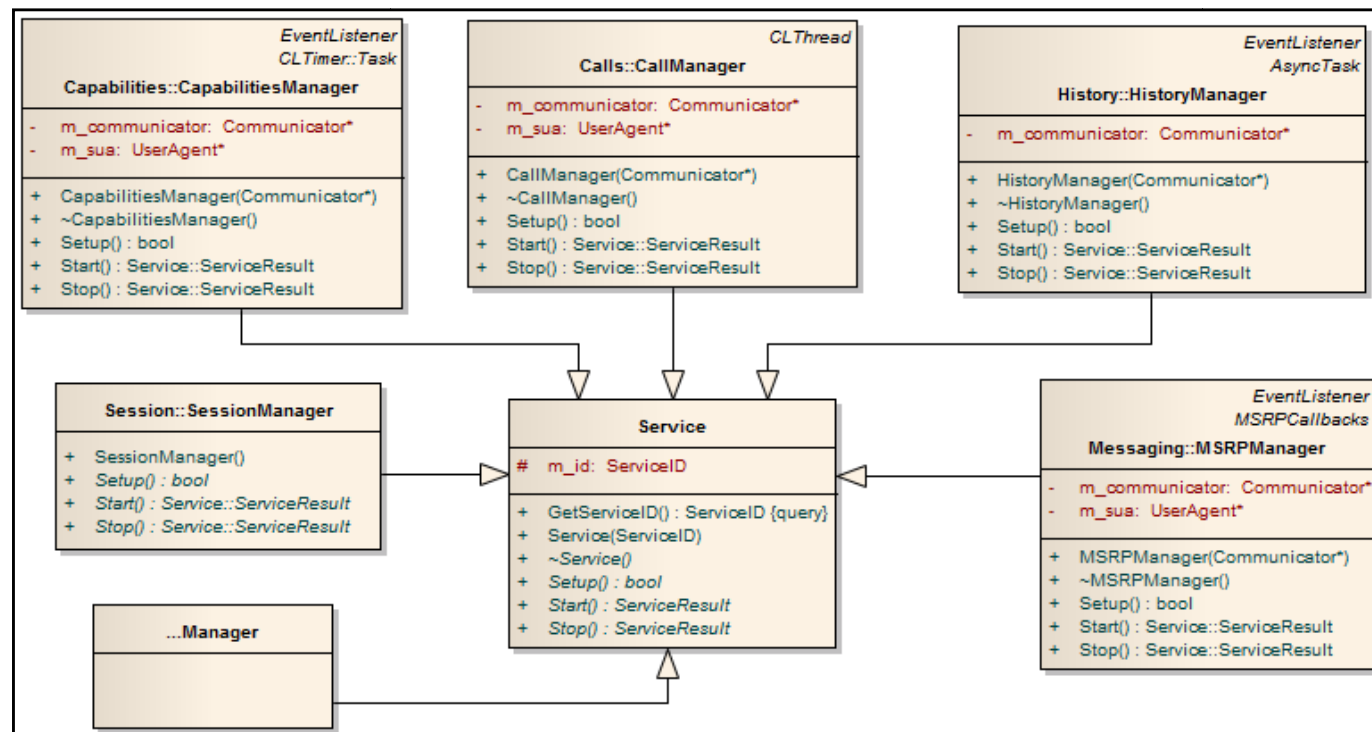


Figura 15 – Diagrama de classes – serviços (CommLib)

Na Figura 16 é possível ver a representação da classe **CallManager**, e dos seus dois atributos, **AudioManager** e **VideoManager**, responsáveis pela transmissão de frames áudio e vídeo entre a UI e o PJSIP, respectivamente. Do ponto de vista lógico, uma **Call** pode ser uma chamada de voz ou vídeo (bidireccional) ou até uma partilha de vídeo (unidireccional). Uma partilha de vídeo bidireccional é também possível com esta estruturação – e, do ponto de vista lógico, é diferente de uma chamada de vídeo bidireccional.

Finalmente, na Figura 17, é possível ver uma representação dos eventos da CommLib. Todos os eventos descendem da classe **CommunicatorEvent**, e têm um tipo, designado por **CommunicatorEventType**. Alguns exemplos de eventos são o **CallEvent**, para comunicar um determinado *estado* da chamada (a tocar no destinatário, em espera activa, atendida, terminada, etc); **IncomingCallEvent**, que indica um pedido de chamada que chegou ao cliente; **VideoShareStatusEvent**, semelhante ao **CallEvent**, mas referente a partilha de vídeo; e também **VideoResolutionUpdatedEvent**, que indica uma upgrade ou downgrade da resolução actual do vídeo.

Os *managers* devem ainda implementar o *interface* **EventListener**, e utilizar a classe **EventDispatcher** para registar um evento. O registo de um evento descreve-se como sendo um associar de uma determinada função a esse evento. Quando esse evento é capturado pelo *manager* que o registou, a função que lhe foi associada no registo é executada.

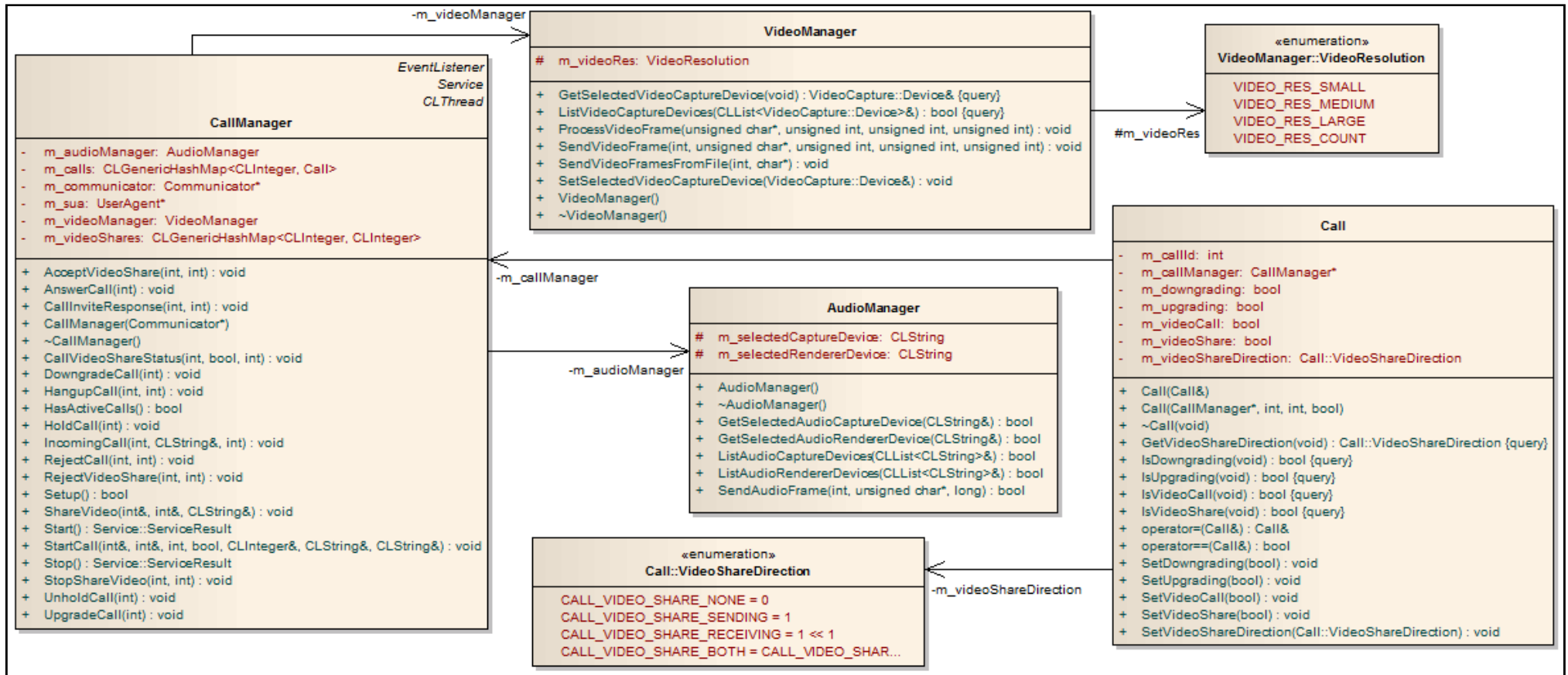


Figura 16 – Diagrama de classes – *CallManager* (CommLib)

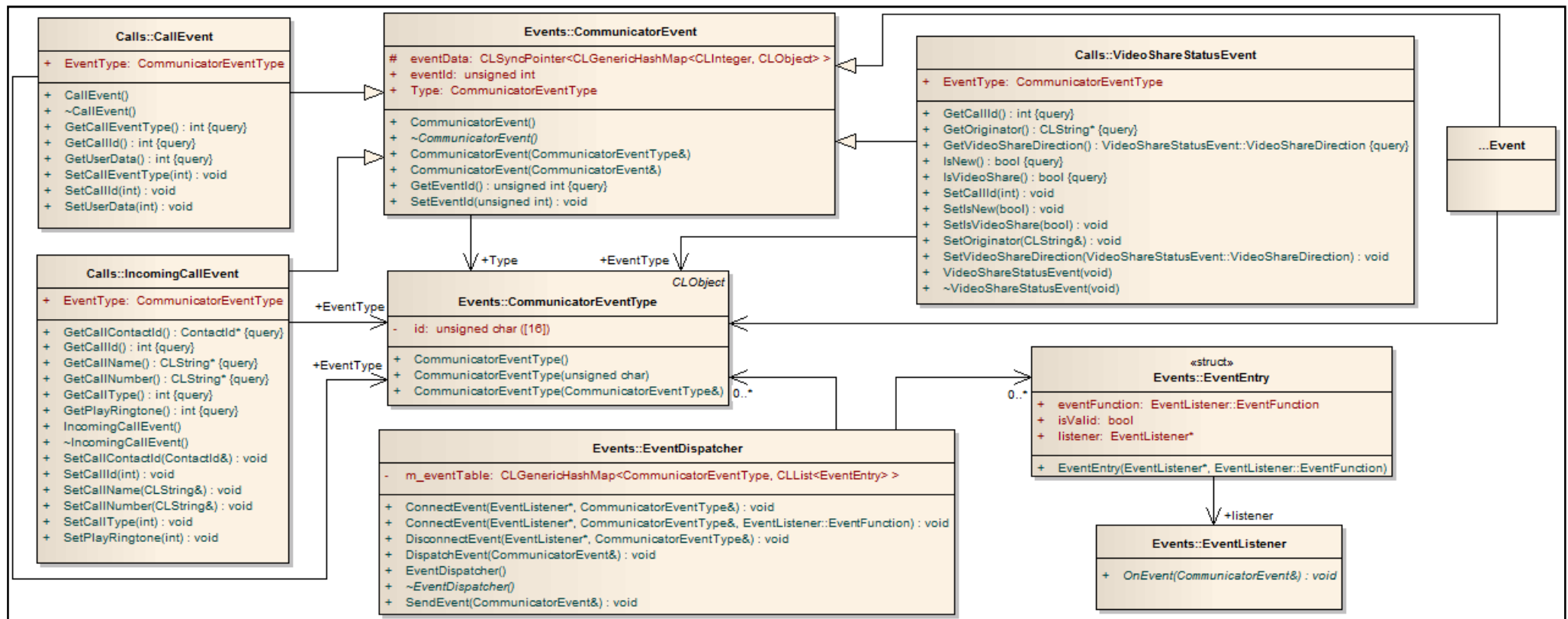


Figura 17 – Diagrama de classes – eventos (CommLib)

3.2.2.3. **PJSIP**

O outro módulo central de um cliente WCS é o PJSIP^[16]. É uma *framework* *opensource* e contém implementações dos protocolos SIP, SDP e RTP/RTCP – fundamentais às funcionalidades das chamadas –, bem como outros protocolos suplementares, para as restantes funcionalidades dos clientes (como o protocolo MSRP para *instant messaging* e transferência de ficheiros). O PJSIP é também a camada responsável pela gestão e uso dos codecs disponíveis – criando instâncias e configurações destes para serem usados durante as chamadas – e ainda servir de camada de transporte/acesso à rede.

3.2.2.4. **Externals**

Finalmente, as bibliotecas externas representam pequenos módulos que oferecem um tipo de funcionalidade mais específica. Destacam-se, como exemplos, a biblioteca **openssl** (uma implementação de protocolos de segurança SSL/TLS), **expat** (parsing de XML) e **ffmpeg** (que contém já algumas implementações de codecs, tendo sido até então usada para a transmissão de vídeo no formato H.263).

3.2.3. **Protocolos**

Os protocolos de rede a ser usados no projecto são:

- **SIP** – Session Initiation Protocol (descrito no RFC 3261^[17]);
- **SDP** – Session Description Protocol (descrito nos RFCs 2327^[18] e 3264^[19]);
- **RTP** – Real-time Transport Protocol (descrito no RFC 3550^[20]) e
- **RTCP** – Real-time Transport Control Protocol (descrito no mesmo RFC que o anterior).

SIP

O protocolo SIP representa um standard para criação, alteração ou término de sessões entre clientes de telecomunicações. O tipo de sessões possíveis varia desde chamadas, transferências de ficheiros, sessões de chat instantâneo, etc. O uso do

protocolo traduz-se na troca de mensagens pedido-resposta, semelhante ao protocolo HTTP, com um código numérico. Os tipos de pedido existentes no protocolo distinguem-se pelo parâmetro “Method” da mensagem e estão descritos na Tabela 3.

Tabela 3 – Tipos de pedidos do protocolo SIP

Método	Descrição
REGISTER	Serve para registar determinada informação de localização ou contacto do cliente (por exemplo endereço IP).
INVITE	Para estabelecimento de uma sessão (uma chamada, por exemplo), o cliente deve criar uma mensagem deste tipo.
ACK	Confirmação de determinada mensagem.
CANCEL	Para cancelar um pedido anterior, ainda em processamento.
BYE	Termina a sessão que foi estabelecida entre dois clientes.
OPTIONS	Refere-se a um pedido de informações sobre o tipo de funcionalidades que um determinado cliente suporta. Por exemplo, uma mensagem deste tipo indica se um cliente suporta ou não partilha de vídeo.

As mensagens de resposta mais relevantes do protocolo estão enumeradas na Tabela 4, com uma breve descrição do que representam no âmbito do projecto:

Tabela 4 – Tipos de resposta do protocolo SIP

Código	Mensagem	Descrição
1xx	Em Processamento	
180	Ringin	O pedido foi recebido e está a ser processado pelo receptor.
2xx	Sucesso	
200	OK	O pedido foi aceite. No caso das chamadas, a sessão é estabelecida e a transmissão RTP começa.
4xx	Falha no Cliente	
400	Bad Request	Pedido contém erros de sintaxe.
403	Forbidden	
404	Not Found	O receptor não foi encontrado ou destino inválido.
408	Request Timeout	Pedido expirou.
481	Call/Transaction Does Not Exist	A sessão referenciada no pedido já não existe.
486	Busy Here	O receptor não pode responder a pedidos (por exemplo, se o cliente não suportar multi-chamadas e já estiver numa).
488	Not Acceptable Here	Descrição da sessão não foi aceite. Por exemplo, quando a negociação de codecs áudio e/ou vídeo falha porque o receptor não suporta nenhum codec anunciado, deve ser usada uma resposta deste tipo.
5xx	Falha do Servidor	
500	Server Internal Error	Falha geral no servidor.
6xx	Falha Geral	
603	Decline	Recusa explícita de sessão.

A Figura 18 ilustra o conjunto de passos executado para o estabelecimento de uma chamada e consequente transmissão RTP de voz e/ou vídeo^[21], do ponto de vista protocolar.

As chamadas são iniciadas por uma mensagem SIP do tipo INVITE, e é através desta mensagem que se procede a uma “negociação”, ou anúncio, da lista de codecs que cada cliente suporta.

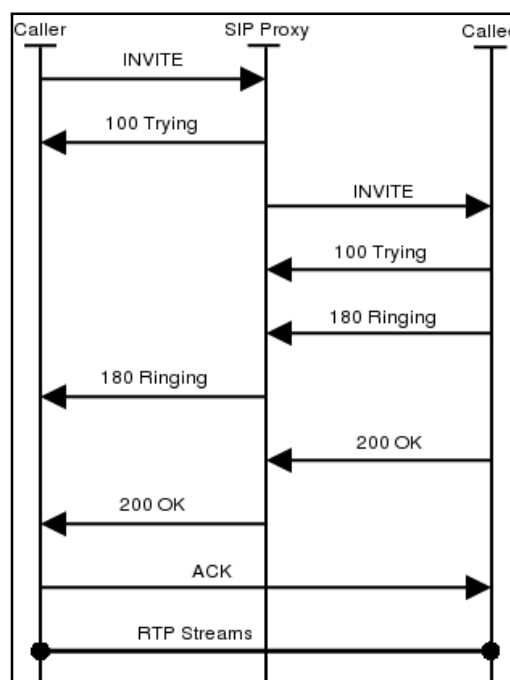


Figura 18 – Iniciação de chamada, via SIP

O iniciador da chamada identifica a sua lista de codecs nessa mensagem e, por sua vez, o receptor verifica se tem suporte para algum deles. Envia depois a sua própria lista, na mensagem OK de resposta. O emissor verifica que consegue codificar/descodificar os dados do receptor de acordo com a lista recebida e dá-se a aceitação ou rejeição da chamada.

A partir deste ponto, a transmissão de voz e vídeo é iniciada, até que a chamada termine, com uma mensagem do tipo BYE. Só após se encontrar um ponto em comum – em que ambos os clientes anunciam um codec e formatos que lhe permitam codificar/descodificar dados que o outro cliente “compreenda” – se consegue prosseguir com a chamada e consequente transmissão dos dados via RTP.

SDP

Conforme descrito, a iniciação de chamada é feita usando uma mensagem do tipo INVITE, do protocolo SIP. Paralelamente a este, é usado ainda um segundo protocolo, o SDP, que serve para “descrever” uma sessão. Esta descrição, no caso de uma chamada, traduz-se pelo anúncio dos codecs áudio e/ou vídeo que cada cliente suporta, e pela “negociação” destes.

O SDP descreve a sessão através de um conjunto de parâmetros, diferenciados por um carácter. O tipo de *media* a usar (áudio ou vídeo) é definido no parâmetro **m** e os atributos relativos ao codec são definidos no parâmetro **a**. Ainda neste parâmetro, existem dois atributos importantes a usar: **rtpmap** e **fntp**. O **rtpmap** designa o codec através de um número identificador, o nome comum do codec e a frequência; já o **fntp** indica pares de valores que são específicos ao codec (por exemplo *bitrates*, *resoluções*, *framerates*, etc.)

No quadro seguinte é apresentado um exemplo de um INVITE que anuncia a lista de codecs suportados por um dado cliente:

```
INVITE sip:+351910000002@wcs.wit-software.com SIP/2.0
Via: SIP/2.0/UDP 10.0.0.17:6060;rport;branch=z9hG4bKPjf862fd7626cb40c18f44a1844bd08ef3
Max-Forwards: 70
From: <sip:+351910000001@wcs.wit-software.com>;tag=d337c1b7ed004b3f9b1d0e4315ac221f
To: <sip:+351910000002@wcs.wit-software.com>
Contact: <sip:10.0.0.17:6060>
Call-ID: b0f4dfcd203b418bae9aac3887ff2bf7
CSeq: 491 INVITE
Route: <sip:213.30.86.106:5062;transport=UDP;lr>
User-Agent: WITPCCommunicator/2.0 (English;)
Content-Type: application/sdp
Content-Length: 279

v=0
o=- 3424462141 3424462141 IN IP4 10.0.0.17
s=pjmedia
c=IN IP4 10.0.0.17
t=0 0
m=audio 4000 RTP/AVP 0 8 4 101
a=rtpmap:4001 IN IP4 10.0.0.17
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:4 G723/8000
a=sendrecv
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15
m=video 31008 RTP/AVP 120 118 119 34 31
a=rtpmap:120 H264/90000
a=fmtp:120 profile-level-id=424014; packetization-mode=0; sprop-parameter-sets=Z0IACpZTBmI,aMljiA==
a=rtpmap:118 H263-1998/90000
a=fmtp:118 CIF=1; QCIF=1; SQCIF=1; F=1; I=1; J=1; T=1
a=rtpmap:119 H263-2000/90000
a=fmtp:119 PROFILE=2; PROFILE=0; LEVEL=40; LEVEL=45
a=rtpmap:34 H263/90000
a=fmtp:34 CIF=1; QCIF=1; SQCIF=1; D=1; F=1
a=rtpmap:31 H261/90000
a=fmtp:31 CIF=1; QCIF=1; D=0
```

Segundo o SDP, cada codec tem atribuído um número, designado por *payload type*, sendo estes enumerados no parâmetro **m=audio** / **m=video**. No exemplo podemos observar os codecs áudio 0, 8 e 4 (o 101 refere-se a um tipo especial de

eventos que não faz parte do espectro do estágio). A ordem pela qual estes números aparecem indica a ordem de preferência a ter em conta pelo receptor da mensagem. Para cada codec, é depois apresentada a lista de parâmetros disponíveis para codificar/descodificar (quando relevante), sendo usado o *payload type* como referência a esse codec. O exemplo mostra o início da negociação dos codecs áudio PCMU (0), PCMA (8) e G.723 (4), e dos codecs vídeo H.264 (120), 3 versões do H.263 (118, 119 e 34) e H.261 (31).

```
...
m=audio 4000 RTP/AVP 0 8 4 101
a=rtcp:4001 IN IP4 10.0.0.17
a=rtptime:0 PCMU/8000
a=rtptime:8 PCMA/8000
a=rtptime:4 G723/8000
...
m=video 31008 RTP/AVP 120 118 119 34 31
a=rtptime:120 H264/90000
a=rtptime:120 profile-level-id=424014; packetization-mode=0; sprop-parameter-
sets=Z0IACpZTBmI,aMljiA==
a=rtptime:118 H263-1998/90000
a=rtptime:118 CIF=1; QCIF=1; SQCIF=1; F=1; I=1; J=1; T=1
a=rtptime:119 H263-2000/90000
a=rtptime:119 PROFILE=2; PROFILE=0; LEVEL=40; LEVEL=45
a=rtptime:34 H263/90000
a=rtptime:34 CIF=1; QCIF=1; SQCIF=1; D=1; F=1
a=rtptime:31 H261/90000
a=rtptime:31 CIF=1; QCIF=1; D=0
...
```

A adição de novos codecs aos clientes WCS obriga a que as mensagens SIP até então usadas devam ser actualizadas, segundo o formato descrito, de modo a contemplar esta nova informação, referente a esses novos codecs.

O indicador numérico do codec, o *payload type*, segue as regras definidas no RFC 3551^[22]. Nele encontram-se definidas tabelas de indicadores para um conjunto de codecs “conhecidos”.

RTP

Relativamente ao protocolo RTP, este não é mais do que o encapsulamento dos dados gerados pelos vários encoders. A cada frame gerada por um encoder é adicionado um cabeçalho RTP, contendo informação de sincronização como

timestamps, ordem numérica do pacote na *stream* RTP (número de sequência), identificação do codec usado, etc.

De notar adicionalmente que um pacote RTP tem um tamanho máximo associado, definido pela rede, designado por MTU^[23], e que, regra geral, uma frame cabe num pacote RTP. Quando tal não acontece – o encoder gera uma frame maior que o MTU disponível – deve existir um processamento adicional, que consiste na divisão lógica da frame em vários blocos, sendo cada um dos blocos encapsulado no seu próprio pacote RTP. Estes “blocos” devem ter o mesmo *timestamp* (para indicar que pertencem à mesma frame) e números de sequência seguidos. No caso dos codecs vídeo, este aspecto é mais notório, uma vez que facilmente as frames vídeo são demasiado grandes para um pacote RTP. Esta “divisão” das frames em vários pacotes RTP é mais conhecida por “pacotização” ou *packetization*.

RTCP

O protocolo RTCP é usado para controlo da *stream* RTP associada. Serve para fornecer a cada *peer* da transmissão determinada informação, que é depois utilizada para calcular, por exemplo, perdas de pacotes ou largura de banda disponível. Esta informação pode depois ser utilizada pelos encoders e decoders para condicionar a sua transmissão RTP, por exemplo baixando ou aumentando o bitrate usado, ou o framerate, ou outros parâmetros conforme necessário.

3.3. Trabalho realizado

Nos Anexos D a H, é feita uma análise de requisitos e descrição mais detalhada da arquitectura e desenvolvimentos que foram realizados no decorrer do estágio, especificamente para cada módulo de funcionalidades.

No Anexo D, é feita uma abordagem à inclusão do codec SILK no WCS, sendo analisado o aspecto geral da *framework* PJSIP no que diz respeito a codecs de áudio

e de como o SILK foi integrado nela, servindo também de exemplo para como outros codecs podem ser integrados futuramente.

No Anexo E, é feita uma análise semelhante à anterior, mas relativamente a codecs vídeo, sendo fornecidas também algumas comparações de imagem que demonstram as melhorias resultantes dos desenvolvimentos.

O Anexo F descreve a nova funcionalidade de partilha de vídeo (vídeo share), na versão Android do cliente móvel do WCS.

O Anexo G descreve o módulo de desenvolvimentos relativos ao suporte para áudio sobre LTE no WIT Mobile Communicator, ficando assim os clientes móveis do WCS em concordância com a versão 5.0 da especificação RCS.

Finalmente, no Anexo H é feita uma descrição dos desenvolvimentos do módulo adicional de funcionalidades, denominado como *Voice Activity Detection*, referente às notificações de estado dos dispositivos de captura e reprodução de áudio, no cliente PC.

4. Conclusões

Finalizado o projecto de estágio, faz-se agora um balanço de todo o processo.

Em termos de produto e empresa, os objectivos iniciais do projecto foram de uma maneira geral atingidos. As melhorias a nível de áudio e vídeo inseridas nos clientes do WCS foram evidentes, dispondo agora o WCS de uma oferta abrangente, actual e de elevada qualidade. Estes melhoramentos permitiram, no decorrer do estágio, e vão continuar a permitir dar ao produto e à empresa uma importante visibilidade, nomeadamente junto do mercado internacional. Tal facto foi comprovado pela decisão da GSMA em escolher a WIT Software como fornecedor oficial de soluções RCS/RCS-e, onde as funcionalidades de voz e vídeo entretanto demonstradas terão feito o seu contributo.

Em termos de aprendizagem do estagiário, houve também evolução. Ainda que num ou noutro momento tenha havido um atraso face ao planeamento de determinada tarefa, tal proporcionou a identificação de problemas a corrigir eventualmente no trabalho futuro. Paralelamente, graças a esse período adicional de assimilação da tecnologia, esse trabalho futuro de integração de outras soluções áudio/vídeo torna-se agora mais acessível, rápido e directo.

Para além disso, dentro da empresa, existia até então apenas um colaborador que era alvo de procura para consulta e apoio, na área do áudio e vídeo do WCS. Os conhecimentos que o estagiário adquiriu nesta área permitiram-no tornar-se num segundo ponto de suporte. Graças a isso, no decorrer do estágio, o estagiário teve oportunidade de fornecer e contribuir de facto com este conhecimento adquirido, no âmbito de outros projectos, com requisitos idênticos, para outros clientes da empresa.

5. Referências

[1] 2012; *Rich Communications*; GSMA

<http://www.gsma.com/rcs/>

[2] 28, Fevereiro 2012; *GSMA Selects WIT Software to Provide RCS-e Application*; GSMA

<http://www.gsma.com/membership/gsma-selects-wit-software-to-provide-rcs-e-application/gsma.com/rcs/>

[3] *The Scrum Guide – the official rulebook*; Scrum.org

<http://www.scrum.org/scrumguides/>

[4] 26, Julho 2011; *Skype, Viber and WhatsApp: The Last Nail in the Operators' Legacy Services Coffin?*; Jonathon Gordon, Director of Marketing; Allot Communications

http://www.mobiletrendsblog.com/MobileTrends_Report_H12011.html

[5] *Mean Opinion Score (MOS) – A Measure of Voice Quality*; Nadeem Unuth; About.com Guide

<http://voip.about.com/od/voipbasics/a/MOS.htm>

[6] *SILK: Super Wideband Audio Codec*; Skype Limited

<http://developer.skype.com/silk>

[7] 22, Março 2011; *Now We're Talking! Steam Voice with SILK*; Valve

<http://store.steampowered.com/news/5100/>

[8] 26, Agosto 2009; *Introduction to H.264/AVC*; Yang Song; The University of Arizona; Tucson, Arizona

<http://www2.engr.arizona.edu/~yangsong/h264.htm>

[9] 27, Fevereiro 2009; *The VP8 video codec: High compression+low complexity*; Jim Bankoski; On2

<http://eetimes.com/design/signal-processing-dsp/4017748/The-VP8-video-codec-High-compression-low-complexity>

[10] 8, Fevereiro 2011; *10 Things You Need to Know About LTE-Advanced*; Stacey Higginbotham; Gigaom

<http://gigaom.com/broadband/lte-advanced/>

[11] GSA, *Information Papers*

http://www.gsacom.com/gsm_3g/info_papers.php4

[12] 2, Julho 2012; *Android Developers – Activity*; Google Inc

<http://developer.android.com/reference/android/app/Activity.html>

[13] 2, Julho 2012; *Android Developers – Tasks and Back Stack*; Google Inc

<http://developer.android.com/guide/components/tasks-and-back-stack.html>

[14] 2, Julho 2012; *Android Developers – Layouts*; Google Inc

<http://developer.android.com/guide/topics/ui/declaring-layout.html>

[15] 2, Julho 2012; *Android Developers – Dialogs*; Google Inc

<http://developer.android.com/guide/topics/ui/dialogs.html>

[16] *Open source SIP stack and media stack for presence, im/instant messaging, and multimedia communication*; Benny Prijono.

<http://www.pjsip.org/>

[17] Junho 2002, *SIP: Session Initiation Protocol*; J. Rosenberg (dynamicsoft) et al.

<http://www.ietf.org/rfc/rfc3261.txt>

[18] Abril 1998, *SDP: Session Description Protocol*; M. Hadley, V. Jacobson; ISI/LBNL.

<http://www.ietf.org/rfc/rfc2327.txt>

[19] Junho 2002, *An Offer/Answer Model with the Session Description Protocol (SDP)*; J. Rosenberg (dynamicsoft), H. Schulzrinne (Columbia University).

<http://www.ietf.org/rfc/rfc3264.txt>

[20] Julho 2003, *RTP: A Transport Protocol for Real-Time Applications*; H. Schulzrinne (Columbia University) et al.

<http://www.ietf.org/rfc/rfc3550.txt>

[21] 2003; *SIP Introduction – A brief overview of SIP describing all important aspects of the Session Initiation Protocol*; Jan Janak; FhG FOKUS

http://ftp.iptel.org/pub/ser/0.8.14/doc/html/sip_introduction.html

[22] Julho 2003, *RTP Profile for Audio and Video Conferences with Minimal Control*; H. Schulzrinne (Columbia University), S. Casner (Packet Design).

<http://www.ietf.org/rfc/rfc3551.txt>

[23] *MTU*; Bradley Mitchell; About.com.

<http://compnetworking.about.com/od/networkprotocols/g/mtu-maximum.htm>

[24] 19, Abril 2012, *Rich Communication Suite 5.0 Advanced Communications Services and Client Specification*; GSMA.

http://www.gsma.com/rcs/wp-content/uploads/2012/03/RCS5.0_Specifications_PDF.zip

[25] Setembro 1997, *RTP Payload Format for H.263 Video Streams*; C. Zhu (Intel Corp.).

<http://www.ietf.org/rfc/rfc2190.txt>

[26] Fevereiro 2005, *RTP Payload Format for H.264 Video*; S. Wenger et al.

<http://www.ietf.org/rfc/rfc3984.txt>

[27] 16, Dezembro 2011, *RCS-e – Advanced Communications: Services and Client Specification*; GSMA.

http://www.gsma.com/rcs/wp-content/uploads/2012/03/rcs-e_advanced_comms_specification_v1.2.1.pdf

[28] 22, Março 2011, *IMS Profile for Voice and SMS*; GSMA.

<http://www.gsma.com/newsroom/wp-content/uploads/2012/03/ir9240.pdf>

[29] Abril 2007, *RTP Payload Format and File Storage Format for the Adaptive Multi-Rate (AMR) and Adaptive Multi-Rate Wideband (AMR-WB) Audio Codecs*; J. Sjöberg (Ericsson) et al.

<http://www.ietf.org/rfc/rfc4867.txt>

[30] 23, Março 2011, *ANSI-C code for the Adaptive Multi-Rate (AMR) speech codec*; 3GPP.

<http://www.3gpp.org/ftp/Specs/html-info/26073.htm>

[31] 23, Março 2011, *ANSI-C code for the Adaptive Multi-Rate – Wideband (AMR-WB) speech codec*; 3GPP.

<http://www.3gpp.org/ftp/Specs/html-info/26173.htm>

[32] 19, Maio 2009, *SDP Capability Negotiation*; F. Andreassen (Cisco Systems).

<http://tools.ietf.org/html/draft-ietf-mmusic-sdp-capability-negotiation-10>

[33] Julho 2003, *Session Description Protocol (SDP) Bandwidth Modifiers for RTP Control Protocol (RTCP) Bandwidth*; S. Casner (Packet Design).

<http://www.ietf.org/rfc/rfc3556.txt>

Outros:

- Wikipedia, The Free Encyclopedia – <http://en.wikipedia.org>
- 6, Agosto 2007; *How video compression works*; BDTI; <http://eetimes.com/design/signal-processing-dsp/4017518/How-video-compression-works>
- 14, Março 2005; *Inside DSP on Digital Video: H.264: the Video codec to watch*; Georgi Beloev e Jennifer Eyre; <http://eetimes.com/design/signal-processing-dsp/4010965/Inside-DSP-on-Digital-Video-H-264-the-Video-codec-to-watch>