



INSTITUTO SUPERIOR TÉCNICO  
Universidade Técnica de Lisboa

## **Application layer QoS support for video services**

**Bruno Miguel Marcos Guedes**

Dissertação para obtenção do Grau de Mestre em  
**Engenharia de Redes de Comunicações**

### **Júri**

Presidente: Professor Doutor Rui Jorge Morais Tomaz Valadas

Orientador: Professora Doutora Teresa Maria Sá Ferreira Vazão Vasques

Co-orientador: Professor Doutor António Carlos da Cruz Semedo Varela

Vogais: Professor Doutor Rui António dos Santos Cruz

**Abril de 2009**

## Agradecimentos

É com o pensamento de dever cumprido, que quero agradecer em primeiro lugar aos meus pais, pelo esforço e apoio dispendidos no meu percurso académico. Também lhes quero agradecer, por todo o carinho e amor que têm por mim.

Agradeço a toda a minha família pelo carinho e apoio demonstrados, em especial o meu padrinho e os meus avós pela amizade que nos une e pelo apoio dado, que influenciou significativamente o meu percurso académico.

Uma palavra de gratidão a todos os meus amigos e colegas, que sempre me apoiaram.

Um agradecimento especial à Professora Doutora Teresa Vazão e ao Professor Doutor António Varela, pela orientação concedida e por todo o seu apoio e incentivo.

Agradeço a toda a equipa da empresa *Streambow*, em especial ao Professor Norbert Stöhr por todo o seu apoio, paciência e sabedoria, que em muito contribuíram para a realização desta tese.

Por fim, mas não menos importante, quero agradecer ao Professor Fernando Pereira, pelos seus vastos conhecimentos na área de vídeo.

A todos os referidos, o meu Muito Obrigado!

## Resumo

A explosão da tecnologia IP está na génese de muitos dos serviços actuais, como é o caso do IPTV. A maior parte destes serviços, tem como principal ponto de falha a rede de acesso, onde é frequentemente utilizada a tecnologia xDSL.

O serviço IPTV é muito sensível a perdas e a atrasos, assim o objectivo desta tese, é desenvolver uma solução que adapte os conteúdos de vídeo à rede de acesso do cliente.

Com o intuito de fornecer diferentes qualidades aos clientes de acordo com as suas capacidades de recepção, foi criada uma solução num MANE. Contudo esta solução não foi adequada, devido ao grande impacto visual aquando de pequenos cortes no vídeo. Deste modo, foi criada uma segunda técnica, onde o servidor envia diferentes qualidades do mesmo conteúdo. A qualidade que o cliente irá receber é decidida pelo MANE, antes dos conteúdos chegarem à rede de acesso. Esta solução demonstrou ser melhor que a anterior, permitindo maiores reduções de débito sem um elevado impacto visual.

Para obter informação dinâmica relativamente à capacidade de recepção dos clientes, foi utilizada a solução *Axess*, baseada no TR-069.

A solução desenvolvida nesta tese baseou-se num MANE que gere as diferentes qualidades enviadas pelo servidor de vídeo, recorrendo à solução *Axess* para decidir qual dos fluxos melhor se adequa à qualidade de recepção de cada cliente. Isto permite uma adaptação às características actuais do cliente e não a uma capacidade de recepção predefinida de acordo com a sua localização geográfica ou qualquer outro critério.

**Palavras-chave:** IPTV, xDSL, dinâmica, adaptação, rede de acesso.

## **Abstract**

The IP technology explosion is the starting-point to a lot of current services, such as the IPTV. Most of these services have the bottleneck in the access network.

The IPTV service is very sensitive to packet losses and delays, so the goal of this thesis is to present a solution that adapts video contents to the last mile of the DSL networks.

In order to provide to the clients different qualities according with their reception quality, it was created the solution in the MANE. However this solution wasn't adequate, due to the huge visual impact even when a small number of video packets were dropped. In this way, it was built a second technique, where the server sends different qualities of the same content. The quality retrieved to the client is decided by MANE before the contents arrive to the access network. It was demonstrated that this technique is better than the last one, allowing more bitrate reduction without a huge visual impact.

To get dynamically information, relatively to the clients reception capacity, was used the *Axess* solution, based on TR-069.

The developed solution in this thesis uses a MANE, which manages different qualities sent by the streamer, using the *Axess* solution to decide which of quality flows fits best to reception capacity quality for each client. This allows an adaptation to the client current characteristics and not to a reception capacity predefined, agreeing with geographic location of the client or any other criteria.

**Keywords:** IPTV, xDSL, dynamic, adaptation, access network.

## **Lista de Acrónimos**

**AAC** – *Advanced Audio Coding*

**ACS** – *Auto Configuration Server*

**ADSL** – *Asymmetric Digital Subscriber Line*

**ADTE** – *Adaptation Decision Taking Engine*

**API** – *Application Programming Interface*

**ARP** – *Address Resolution Protocol*

**ASTRALS** – *Audio-visual Streaming plAtaform for domestic Leisure and Security*

**AVC** – *Advanced Video Coding*

**B-NT** – *Broadband Network Termination*

**CATV** – *Cable Television*

**CD** – *Compact Disk*

**CGS** – *Coarse Grain Scalability*

**CIF** - *Common Intermediate Format*

**CPE** – *Customer Premises Equipment*

**CPU** – *Central Processing Unit*

**CWMP** – *CPE WAN Management Protocol*

**DANAE** – *Dynamic and distributed Adaptation of scalable multimedia coNtent in a context-Aware Environment*

**DIA** – *Digital Item Adaptation*

**DVB** – *Digital Video Broadcasting*

**DSL** – *Digital Subscriber Line*

**DSLAM** – *Digital Subscriber Line Access Multiplexer*

**DSS** – *Darwin Streaming Server*

**FGS** – *Fine Granularity Scalability*

**FPS** – *Frame Per Second*

**FTTH** – *Fiber To The Home*

**GOP** – *Group of Pictures*

**ICC** – *Instant Channel Change*

**IDR** – *Instantaneous Decoding Refresh*

**IGMP** – *Internet Group Management Protocol*

**IP** – *Internet Protocol*

**IPTV** – *Internet Protocol Television*

**ITU-T** – *International Telecom Union*

**JAPI**– *Java API*

**JVT** – *Joint Video Team*

**LAN** – *Local Area Network*

**MAC** – *Media Access Control*

**MANE** – *Media-Aware Network Element*

**MGS** – *Medium Grain Scalability*

**MPEG** – *Moving Picture Experts Group*

**MOS** – *Mean Opinion Score*

**MP3** – *MPEG-1 Audio Layer 3*

**NAL** – *Network Abstraction Layer*

**NAT** – *Network Address Translator*

**NMM** – *Network-Integrated Multimedia Middleware*

**PC** – *Personal Computer*

**PDA** – *Personal Digital Assistants*

**PES** – *Packetized Elementary Stream*

**PID** – *Packet Identifier*

**PLC** – *Power Line Communications*

**PS** – *Program Stream*

**PSNR** – *Peak SNR*

**PSTN** – *Public Switched Telephone Network*

**PT** – *Portugal Telecom*

**PVC** – *Permanent Virtual Circuit*

**QoE** – *Quality of Experience*

**QoS** – *Quality of Service*

**QCIF** – *Quarter CIF*

**RADIUS** – *Remote Authentication Dial In User Service*

**RAM** – *Random Access Memory*

**RFC** – *Request for Comments*

**RPC** – *Remote Procedure Call*

**RTP** – *Real-time Transport Protocol*

**RTCP** – *Real-Time Transport Control Protocol*

**RTSP** – *Real Time Streaming Protocol*

**SAP** – *Session Announcement Protocol*

**SDP** – *Session Description Protocol*

**SIP** – *Session Initiation Protocol*

**SLA** – *Service Level Agreement*

**SNR** – *Signal to Noise Ratio*

**SOAP** - *Simple Object Access Protocol*

**SSH** – *Secure Shell*

**SRTP** – *Secure RTP*

**STB** – *Set-top Box*

**TS** – *Transport Stream*

**UDP** – *User Datagram Protocol System*

**UCD** – *Universal Constraints Descriptor*

**UED** – *Usage Environment Description*

**UMTS** – *Universal Mobile Telecommunications*

**URI** – *Uniform Resource Identifier*

**VLC** – *VideoLan Client*

**VoD** – *Video on Demand*

**VoIP** – *Voice over IP*

**WAN** – *Wide Area Network*

**4CIF** - *4 times CIF*

# Índice

<b>AGRADECIMENTOS</b> .....	<b>1</b>
<b>RESUMO</b> .....	<b>2</b>
<b>ABSTRACT</b> .....	<b>3</b>
<b>LISTA DE ACRÓNIMOS</b> .....	<b>4</b>
<b>LISTA DE FIGURAS</b> .....	<b>9</b>
<b>LISTA DE TABELAS</b> .....	<b>10</b>
<b>INTRODUÇÃO</b> .....	<b>11</b>
1.1 ENQUADRAMENTO .....	11
1.2 OBJECTIVOS E CONTRIBUIÇÕES .....	12
1.3 ESTRUTURA DO DOCUMENTO.....	14
<b>2 ESTADO DA ARTE</b> .....	<b>15</b>
2.1 ADAPTAÇÃO DE VÍDEO .....	16
2.1.1 <i>Escalabilidade na codificação</i> .....	18
2.1.2 <i>H.264/SVC</i> .....	19
2.1.3 <i>MPEG-J</i> .....	21
2.2 TAXIONOMIA DE GESTÃO DA <i>QUALITY OF SERVICE (QoS)</i> .....	22
2.3 TÉCNICAS DE ADAPTAÇÃO .....	23
2.3.1 <i>Headend</i> .....	23
2.3.2 <i>Núcleo da rede</i> .....	27
2.3.3 <i>Rede doméstica</i> .....	30
2.3.4 <i>Ponto genérico na rede</i> .....	31
<b>3 ESCOLHA DA TÉCNICA DE ADAPTAÇÃO</b> .....	<b>33</b>
3.1 REQUISITOS .....	33
3.2 TÉCNICAS DE ADAPTAÇÃO .....	34
3.2.1 <i>Degradação de conteúdos no MANE</i> .....	34
3.2.2 <i>Degradação de conteúdos no servidor</i> .....	45
3.2.3 <i>Síntese Final</i> .....	51
<b>4 ADAPTAÇÃO DINÂMICA</b> .....	<b>52</b>
4.1 CONSIDERAÇÕES INICIAIS .....	52
4.2 SOLUÇÃO DINÂMICA.....	52
<b>5 SOLUÇÃO PROPOSTA PARA ADAPTAÇÃO DO VÍDEO NA REDE DE ACESSO</b> .....	<b>60</b>
5.1 DESCRIÇÃO GERAL .....	60
5.1.1 <i>Pressupostos</i> .....	61
5.2 ARQUITECTURA DA SOLUÇÃO.....	61
5.2.1 <i>Descrição geral</i> .....	61
5.2.2 <i>Modo de funcionamento</i> .....	62
5.2.3 <i>Descrição detalhada dos blocos funcionais</i> .....	63
5.3 AMBIENTE DE TESTE .....	68
5.4 RESULTADOS.....	70



5.5	CONDICIONALISMOS DA IMPLEMENTAÇÃO NUM AMBIENTE REAL .....	72
5.5.1	<i>Solução no DSLAM</i> .....	73
5.5.2	<i>Solução no TR-135</i> .....	74
<b>6</b>	<b>CONCLUSÃO</b> .....	<b>76</b>
6.1	TRABALHO FUTURO .....	78
	<b>REFERÊNCIAS</b> .....	<b>80</b>
	<b>ANEXOS</b> .....	<b>82</b>
	ANEXO A – TECNOLOGIAS DE SUPORTE .....	82
	<i>Netfilter e Iptables</i> .....	82
	<i>Tipos de comunicação</i> .....	84
	ANEXO B.....	87
	ANEXO C – CWMP.....	88
	<i>Especificação CWMP</i> .....	88
	<i>Axess – Plataforma de gestão de CPEs</i> .....	91
	ANEXO D – ESTRUTURA DA UNIDADE NAL DO H.264/SVC.....	93

## Lista de Figuras

FIGURA 1 – PERSPECTIVA DE CRESCIMENTO DOS UTILIZADORES DE IPTV EM TODO O MUNDO. ....	11
FIGURA 2 – LOCALIZAÇÃO DOS PROBLEMAS DE QUALIDADE NUM SERVIÇO DE IPTV.....	12
FIGURA 3 – ARQUITECTURA DE IPTV.....	15
FIGURA 4 – DISTRIBUIÇÃO DAS CAMADAS PELOS TRÊS FACTORES. ....	19
FIGURA 5 – TAXIONOMIA DAS SOLUÇÕES PARA A GESTÃO DE QOS. ....	22
FIGURA 6 – ARQUITECTURA DA MÁQUINA DE ADAPTAÇÃO.....	25
FIGURA 7 – CENÁRIO DE <i>STREAMING</i> ADAPTATIVO <i>SVC</i> .....	25
FIGURA 8 – CENÁRIO DE ADAPTAÇÃO DISTRIBUÍDA.....	28
FIGURA 9 – SISTEMA <i>NMM</i> .....	31
FIGURA 10 – GRAFO DE FLUXO DE CONTEÚDO MULTIMÉDIA. ....	32
FIGURA 11 – EXEMPLO DA MÉTRICA DO ÚLTIMO PACOTE <i>IP</i> , COM DESCARTE NAS TRAMAS MAIS PEQUENAS.....	35
FIGURA 12 – EXEMPLO DA MÉTRICA DE TRAMA COM DESCARTES NO FINAL DO <i>GOP</i> .....	35
FIGURA 13 – AMBIENTE DE TESTE.....	38
FIGURA 14 – GRÁFICO DA COMPARAÇÃO DAS MÉTRICAS ANALISADAS NO TESTE 1.....	41
FIGURA 15 – QUALIDADES OBTIDAS (DEVIDAMENTE IDENTIFICADAS) PARA O VÍDEO CONSIDERADO. ....	48
FIGURA 16 – COMPARAÇÃO ENTRE A MELHOR MÉTRICA DE DEGRADAÇÃO NO <i>MANE</i> E A DEGRADAÇÃO NO SERVIDOR.....	50
FIGURA 17 – RITMOS DE MAIOR E MENOR PERÍODO.....	54
FIGURA 18 – FLUXOGRAMA COM O ALGORITMO DE ADAPTAÇÃO.....	55
FIGURA 19 – EXEMPLO DO EFEITO DE <i>PING PONG</i> .....	56
FIGURA 20 – FLUXOGRAMA COM O MÉTODO UTILIZADO PARA REDUZIR O <i>PING PONG</i> .....	58
FIGURA 21 – UTILIZAÇÃO DO MECANISMO DE REDUÇÃO DO EFEITO DE <i>PING PONG</i> .....	59
FIGURA 22 – AMBIENTE GERAL DA SOLUÇÃO DE REENCAMINHAMENTO DE PORTAS.....	60
FIGURA 23 – ARQUITECTURA DA SOLUÇÃO.....	62
FIGURA 24 – MODELO RELACIONAL DA BASE DE DADOS.....	66
FIGURA 25 – AMBIENTE DE TESTE DA SOLUÇÃO.....	68
FIGURA 26 – IMAGENS COM A DIFERENÇA DE QUALIDADE ANTES E APÓS A ADAPTAÇÃO.....	72
FIGURA 27 – IMPLEMENTAÇÃO NUM AMBIENTE REAL.....	73
FIGURA 28 – DIAGRAMA DO <i>iptables</i> PARA AS TABELAS DE <i>NAT</i> E <i>FILTER</i> .....	83
FIGURA 29 – EXEMPLO DE <i>BROADCAST</i> , <i>UNICAST</i> E <i>MULTICAST</i> .....	84
FIGURA 30 – DIFERENÇAS ENTRE <i>IGMPv2</i> E <i>IGMPv3</i> .....	86
FIGURA 31 – <i>CWMP</i> E <i>ACS</i> .....	88
FIGURA 32 – COMPONENTES DO <i>CWMP</i> .....	89
FIGURA 33 – EXEMPLO DE UMA SESSÃO ENTRE <i>ACS</i> E UM <i>CPE</i> .....	91
FIGURA 34 – PLATAFORMA <i>AXESS</i> .....	92
FIGURA 35 – ESTRUTURA DA UNIDADE <i>NAL</i> EM <i>H.264/SVC</i> .....	93

## Lista de Tabelas

TABELA 1 – DESCRIÇÃO DOS TESTES REALIZADOS.....	36
TABELA 2 – CLASSIFICAÇÃO DOS VALORES DE QoE.....	37
TABELA 3 – RESULTADOS DO PRIMEIRO TESTE.....	40
TABELA 4 – RESULTADOS OBTIDOS DO SEGUNDO TESTE.....	43
TABELA 5 – RESULTADOS DO TERCEIRO TESTE.....	43
TABELA 6 – TABELA COM OS RESULTADOS OBTIDOS DA ADAPTAÇÃO REALIZADA.....	49
TABELA 7 – CAMPOS DA BASE DE DADOS LOCAL DO MANE.....	67
TABELA 8 – RESULTADOS DOS TESTES DA SOLUÇÃO, RELATIVAMENTE AO TEMPO DE ADAPTAÇÃO.....	70
TABELA 9 – ANÁLISE DE QoE ANTES E APÓS A ADAPTAÇÃO.....	71

## Introdução

### 1.1 Enquadramento

A explosão tecnológica das redes de comunicações suportadas no *Internet Protocol* (IP), está na génese de muitos dos serviços actuais, entre os quais do *Internet Protocol Television* (IPTV). O IPTV define-se pela transmissão de conteúdos digitais, dentro de uma rede privada que utiliza o protocolo de comunicação IP ao nível da camada de rede [1].

A grande vantagem de um serviço de IPTV relativamente à televisão tradicional tem a ver com a maior interactividade que oferece ao utilizador, permitindo um controlo personalizado sobre os conteúdos. Enquanto que, na televisão tradicional todos os canais são difundidos por todos os clientes, no IPTV apenas é enviado para o cliente, o canal que este pediu. Uma outra vantagem desta solução está relacionada com a tão pretendida convergência, constituindo uma das principais razões da migração dos serviços tradicionais, como o caso de vídeo, voz e dados para o IP, oferecendo aos utilizadores um único serviço que engloba todas estas vertentes, usualmente designado por “*triple play*” [2]. O objectivo é fornecer numa única ligação de banda larga os diferentes serviços, recorrendo a um ambiente comum assente no protocolo IP. Actualmente já se fala inclusive no “*quadplay*” que envolve também as redes móveis, para além dos serviços anteriormente referidos.

O serviço de IPTV tem ganho cada vez mais adeptos, conforme se ilustra na Figura 1, com as previsões do crescimento do número de utilizadores de IPTV em todo o mundo. Neste estudo [2], prevê-se que o número de utilizadores passe 13,5 milhões em 2007 para 72,6 milhões em 2011, correspondendo a um crescimento anual de cerca de 40%.

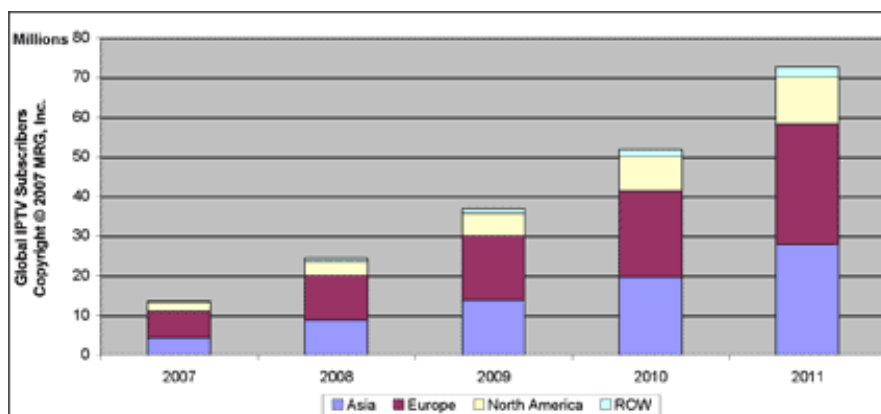


Figura 1 – Perspectiva de crescimento dos utilizadores de IPTV em todo o mundo.<sup>1</sup>

<sup>1</sup> Extraído de [2].

Em Portugal, desde Abril de 2006 que existe serviço de IPTV da Clix, designado de “*smart TV*”. Em Junho de 2007 a *Portugal Telecom* (PT) lançou o seu serviço de IPTV, o MEO [3].

## 1.2 Objectivos e contribuições

Actualmente com o elevado crescimento de serviços baseados em IP (como o IPTV), são feitas maiores exigências (por exemplo de largura de banda) às redes que os suportam. A tecnologia *Digital Subscriber Line* (DSL) tem sido amplamente utilizada para fornecimento de serviços IP, possuindo um vasto conjunto de tipos, sendo o mais comum o *Asymmetric Digital Subscriber Line* (ADSL). Estas redes caracterizam-se pela transmissão de informação digital por um par entrançado de cabos de cobre (aproveitam a linha telefónica) e pelo facto de terem ligações dedicadas entre os sistemas *Digital Subscriber Line Access Multiplexer* (DSLAM) e o *modem* dos clientes de DSL. Uma rede ADSL tem como principal ponto crítico a rede de acesso, não só pelos problemas inerentes ao próprio meio de transmissão (por exemplo, interferências no cabo), mas também pelo próprio comprimento do cabo, que influencia directamente a atenuação e conseqüentemente a largura de banda disponível para o cliente.

Num ambiente de IPTV, o vídeo é extremamente sensível a atrasos e perdas. Assim é necessário adequar o vídeo, para que o cliente receba a melhor qualidade possível, dadas as suas capacidades de recepção. Deste modo, para evitar a degradação do vídeo, este é enviado com o ritmo mais elevado possível, contudo receber o vídeo a um ritmo muito superior à capacidade da rede provoca grandes perdas. Na Figura 2, estão ilustradas as partes da rede que introduzem maiores problemas de qualidade, sendo que a rede de acesso é o segundo sítio com maiores problemas.

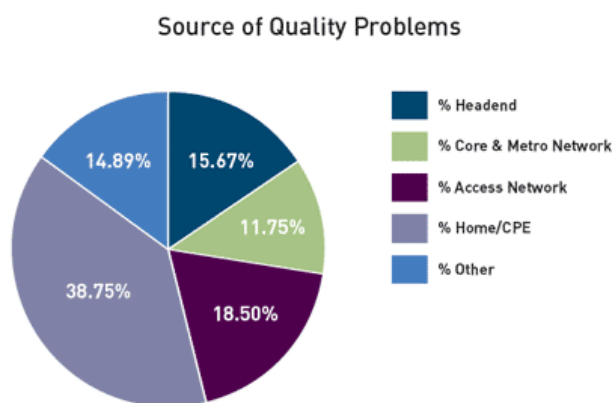


Figura 2 – Localização dos problemas de qualidade num serviço de IPTV.<sup>2</sup>

Tendo em conta os problemas anteriormente referidos na rede de acesso das redes DSL, esta tecnologia encontra-se actualmente a ser preterida pela fibra óptica, como é o caso de *Fiber To The*

<sup>2</sup> Figura extraída de [32].

*Home* (FTTH), que permite disponibilizar maior largura de banda ao cliente. Contudo, esta nova tecnologia ainda não está massificada, principalmente devido ao seu elevado custo. Mesmo que o seu uso se generalize, será sempre usada para zonas densamente povoadas, uma vez que o custo/benefício neste tipo de tecnologia assim o exige. Deste modo, as redes DSL, com todos os seus problemas, têm de ser consideradas, na actualidade, como uma solução viável. Todavia, e uma vez que, no futuro, o seu uso será mais comum em zonas poucas povoadas, é importante ter em consideração que estas redes perdem eficiência nestes cenários, uma vez que devido à pouca densidade populacional, é frequente o DSLAM não estar “à porta” do cliente. Assim, com a maior distância entre o DSLAM e o cliente, maior terá de ser o comprimento do cabo, o que aumenta a atenuação, provocando uma redução no débito binário.

Assim, serão estudadas soluções para a degradação do vídeo com menor impacto visual, de modo a poder fornecer diferentes qualidades a clientes com capacidades de recepção distintas. Será analisada a melhor opção para obter informação dos clientes e, por fim, apresentada uma solução que permita adaptar dinamicamente o vídeo à capacidade de recepção dos clientes. Serão ainda apresentados resultados do modelo criado, como prova de conceito.

Os princípios para a construção da solução foram os seguintes:

1. Capacidade de adaptação dinâmica às características dos clientes, isto é, que evolua com o tempo não se restringindo a uma adaptação inicial.
2. Capacidade de adaptação à rede de acesso do cliente; sendo esta uma rede analógica a adaptação tem de ser feita antes, no servidor ou num elemento intermédio entre o servidor e o cliente.
3. Capacidade de suportar um número elevado de clientes de IPTV, que se consegue através do desenvolvimento duma solução escalável.

Deste modo, a principal contribuição desta tese foi o desenvolvimento de uma nova solução de adaptação de vídeo à capacidade de recepção dos clientes. Contudo, para obtenção desta solução, foram dadas outras contribuições no que respeita a análise e realização de testes de duas técnicas de adaptação da qualidade do vídeo. As referidas técnicas tinham como principal referência a recente norma de codificação de vídeo escalável, o H.264/SVC, que foi desenhada com o objectivo de fornecer diferentes níveis de qualidade aos clientes. Esta solução é também inovadora no modo como se obtém a informação do cliente, utilizando para isso a especificação do *Broadband Forum* de gestão remota de *Customer Premises Equipment* (CPE). Esta tese contribui ainda com uma visão de uma futura implementação da solução, num ambiente real de IPTV.

### 1.3 Estrutura do documento

Nesta tese serão apresentadas duas técnicas de adaptação dos conteúdos de vídeo à rede de acesso e um mecanismo para obter dinamicamente informação das condições de recepção dos clientes. Será também apresentada uma solução que unifica a melhor das duas técnicas de adaptação analisadas com a técnica de adaptação dinâmica. Assim, nesta secção pretende-se dar um mapa global dos capítulos que compõe a tese.

Esta tese encontra-se estruturada em seis capítulos, sendo que o primeiro corresponde ao capítulo actual, onde se pretende dar um enquadramento do âmbito da tese, incluindo o problema que se pretende resolver e a consequente solução encontrada.

No segundo capítulo serão explicados os conceitos básicos para a compreensão e desenvolvimento da tese, assim como algumas soluções já existentes.

No terceiro capítulo serão apresentados e analisados os resultados dos testes de dois caminhos para a adaptação dos conteúdos, de modo a determinar aquele que obtém maior redução de débito com menor impacto visual possível.

No capítulo quatro, será proposto um mecanismo para obter dinamicamente informação da qualidade de recepção dos clientes.

No capítulo cinco, será apresentada uma arquitectura de testes, onde se utilizam as melhores opções obtidas do capítulo três e quatro. Serão apresentados os resultados relativos à implementação dessa mesma arquitectura. Neste capítulo serão também explicados os problemas e as diferenças entre a prova de conceito apresentada e a sua implementação num ambiente real.

No capítulo seis, são apresentadas as conclusões finais, assim como o trabalho de desenvolvimento futuro.

Na secção Anexos, o Anexo A, descreve conceitos básicos como os tipos de comunicação como: o *multicast*, *unicast* e *broadcast*. Também é explicado como funcionam as regras de *iptables*. No Anexo B é detalhado o modo como foram implementadas as regras de *iptables* para o descarte selectivo de pacotes IP (ver 3.2.1). No Anexo C, é descrito detalhadamente o funcionamento da especificação do TR-069 e da solução usada – Axess. No Anexo D, descreve a estrutura (bytes) da norma de codificação H.264/SVC.

## 2 Estado da arte

Uma arquitectura IPTV pode ser hierarquicamente decomposta em três partes fundamentais: *Headend*, distribuição e doméstica – conforme se ilustra na Figura 3.

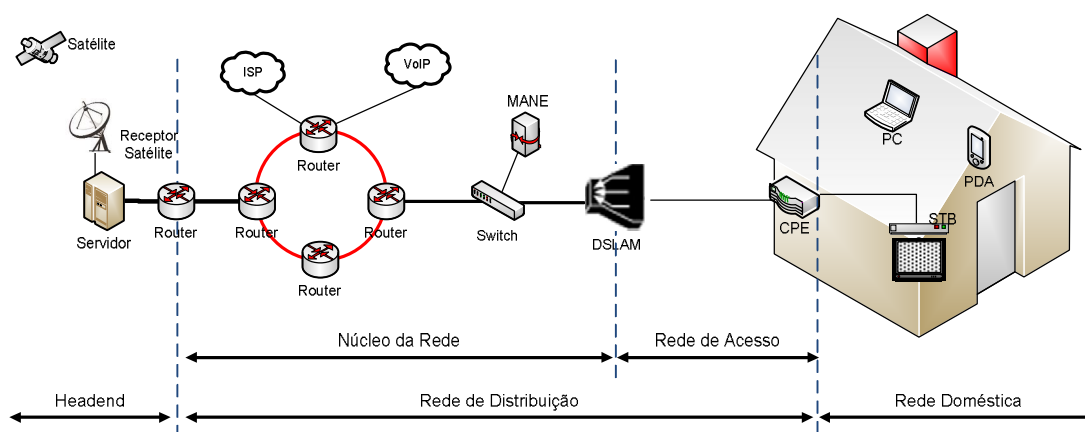


Figura 3 – Arquitectura de IPTV.

O *Headend* é o centro de distribuição num sistema de IPTV [1]. Este é constituído por vários componentes, dos quais se destaca: o *streamer* que transmite os conteúdos; o codificador e o servidor de vídeos a pedido. Os elementos referidos estão representados na Figura 3 por um servidor genérico. Os sinais de vídeo provenientes de uma fonte (por exemplo: satélite e estúdios locais) são recebidos no *Headend*, codificados e convertidos para pacotes IP, para posteriormente serem enviados para a rede de distribuição (núcleo da rede) [4].

Em relação à rede de distribuição, considerar-se-á que esta se encontra dividida em duas partes: núcleo e acesso.

O núcleo corresponde à parte de transporte dos conteúdos, permitindo a interligação entre áreas de distribuição distintas. Tipicamente é usada fibra em várias topologias, sendo mais frequente, a topologia em anel. Esta parte da rede permite a interligação (lógica, ao nível dos serviços) a outros “mundos”, como são os casos do *Voice over IP* (VoIP), *Public Switched Telephone Network* (PSTN), entre outros. É no núcleo da rede que se encontra o *Media-Aware Network Element* (MANE<sup>3</sup>) que poderá ser o responsável por manipular os pacotes de vídeo, com o intuito de adaptar os conteúdos à capacidade do cliente.

<sup>3</sup> Elemento introduzido no *Request For Comments* (RFC) 3984.



A rede de acesso é a zona que permite a interligação entre a rede doméstica e o núcleo da rede. Os tipos de redes de acesso que podem ser usados num serviço IPTV são [1]: DSL<sup>4</sup>, televisão por cabo – *Cable Television* (CATV), banda larga sem fios, rede eléctrica – *Power Line Communications* (PLC), e redes ópticas, sendo que a Figura 3 apenas se foca na tecnologia DSL, pois é a tecnologia que será utilizada neste trabalho.

Para terminar, a rede doméstica vai desde o CPE até ao dispositivo do cliente, como: a televisão, o portátil, *Set-top Box* (STB), entre outros.

Várias soluções têm surgido para a adaptação do vídeo em diferentes pontos da rede IP. Nos próximos capítulos, para além da introdução de algumas normas de codificação de conteúdos, ir-se-á identificar as diversas soluções de qualidade serviço e agrupá-las nas diferentes partes da rede, assim como os seus mecanismos subjacentes. Distinguir-se-á entre codificação dos conteúdos adaptativos e as técnicas de adaptação já existentes.

## 2.1 Adaptação de vídeo

A codificação de vídeo digital tenta explorar dois conceitos: redundância e irrelevância [6]. A redundância está relacionada com as semelhanças, correlação e previsibilidade entre amostras de uma imagem, não envolvendo perdas. A irrelevância tem a ver com informação imperceptível ao olho humano, constituindo um processo irreversível, uma vez que envolve perdas.

O *Moving Picture Experts Group* (MPEG) é a entidade que tem desenvolvido os padrões de codificação de áudio e vídeo. O primeiro grande passo na codificação digital de vídeo foi entre 1988 e 1990 com o MPEG-1<sup>5</sup>, foi criada com o intuito de gravar conteúdos de multimédia digital em *Compact Disk* (CD). A norma MPEG-1 encontra-se dividida em cinco partes, das quais se destaca a parte um Sistema, a parte dois Vídeo, e a parte três Áudio. A parte Sistema especifica a multiplexagem de vários fluxos codificados de áudio e vídeo num único fluxo binário com sincronização. Os fluxos elementares (áudio e vídeo, separados) são divididos em pacotes de comprimento variável, que se designam por *Packetized Elementary Stream* (PES). Estes pacotes, por sua vez, são multiplexados pelo MPEG-1 Sistema. A parte de Vídeo, especifica a codificação do vídeo, tendo como objectivo atingir uma qualidade mínima aceitável para ritmos de cerca de 1,2 Mb/s. O vídeo apenas é suportado no modo progressivo (todas as linhas da imagem são “desenhadas” em sequência). A parte de Áudio especifica a codificação do áudio, com um ritmo que pode variar entre 32 kb/s a 448 kb/s por canal (pode ter no máximo dois canais, estéreo). No Áudio são definidas três camadas, que se diferenciam por três parâmetros: atraso,

---

<sup>4</sup> Para mais informação sobre a tecnologia DSL, consultar [5].

<sup>5</sup> Definido em ISO/IEC 11172.

débito e complexidade. As camadas mais altas têm mais atraso, complexidade e eficiência na codificação, sendo que a camada três corresponde ao famoso *MPEG-1 Audio Layer 3* (MP3).

O segundo grande passo na codificação, foi o surgimento do *Digital Video Broadcasting* (DVB)<sup>6</sup> (em 1993) e com a consequente especificação de uma nova norma (MPEG-2<sup>7</sup>) para a codificação dos conteúdos a difundir. As normas MPEG-1 e MPEG-2 são compatíveis, sendo que o MPEG-2 se demarca do seu antecessor pelo aumento na capacidade de codificação, podendo mesmo codificar conteúdos de alta definição. É constituído por dez partes, das quais se destacam as mesmas três observadas no MPEG-1. A parte Sistema introduziu dois novos conceitos na multiplexagem: o *Program Stream* (PS) mais orientado para armazenamento de vídeo com pacotes de tamanho variável; o *Transport Stream* (TS), também designado de MPEG-TS, é utilizado para transmissão, tendo os pacotes, um tamanho fixo de 188 *bytes*. Deste modo, os PES de áudio e vídeo são multiplexados num único fluxo TS ou PS. Na parte de Vídeo, existem duas grandes diferenças, a codificação de conteúdos entrelaçados (permitindo uma codificação mais eficiente, uma vez que as linhas pares e ímpares são codificadas em separado) e a possibilidade de criar fluxos escaláveis: escalabilidade espacial, isto é, com diferentes resoluções e escalabilidade na qualidade *Signal to Noise Ratio* (SNR). Na parte de Áudio, o MPEG-2 tem duas especificações: a parte dois, que é compatível com o MPEG-1, mas que permite codificar mais do que dois canais; a parte sete, *Advanced Audio Coding* (AAC), que não é compatível com o MPEG-1, tendo maior eficiência (relação débito e qualidade).

O surgimento do MPEG-4<sup>8</sup>, por volta de 1998, veio criar um novo paradigma: codificação por objectos audiovisuais. O MPEG-4 é constituído por diversas partes, onde se destaca a parte Sistema, que representa de forma independente os vários objectos na cena, nomeadamente objectos visuais, permitindo o seu acesso, manipulação e reutilização. Uma cena pode ser composta por objectos sintéticos (gerados por computador) e/ou naturais (captados do mundo real – por exemplo por câmara ou microfone). Cada objecto pode ter uma qualidade e protecção contra erros dependente da sua relevância. O MPEG-4 tem duas partes de vídeo: a parte dois Visual, que especifica a codificação de objectos naturais, sintéticos e visuais; a parte dez, *Advanced Video Coding* (AVC), que especifica a codificação avançada de vídeo rectangular, dando origem ao também conhecido por H.264/AVC, visto que foi desenvolvida em conjunto com a *International Telecom Union* (ITU-T). O H.264/AVC representa um aumento de eficiência na codificação de cerca 50% relativamente ao MPEG-2 Vídeo, em troca necessita de mais memória e capacidade computacional.

Após esta breve introdução de algumas normas de codificação de conteúdos, será em seguida abordado o problema da escalabilidade na codificação.

---

<sup>6</sup> Consórcio que define as normas para a transmissão digital de televisão em vários meios como, terrestre (DVB-T), cabo (DVB-C), entre outros.

<sup>7</sup> Definido em ISO/IEC 13818.

<sup>8</sup> Definido em ISO/IEC 14496.

### 2.1.1 Escalabilidade na codificação

A existência dum número cada vez maior de dispositivos com a capacidade de receber conteúdos digitais, fez surgir a necessidade de ter escalabilidade nos conteúdos multimédia, de forma a permitir que dispositivos com diferentes capacidades computacionais os possam visionar.

A escalabilidade na codificação de vídeo tem sido um tópico de discussão, em especial a partir de 2003 com o surgimento do H.264/AVC [7] [8]. Na verdade, a necessidade de conseguir transmitir o vídeo de forma eficiente, com baixa largura de banda, tinha sido reconhecida no standard MPEG-1, o qual, na parte de Vídeo definiu três tipos de tramas: tramas I, que não usam predição temporal e que constituem pontos de acesso aleatório<sup>9</sup> dentro do vídeo; tramas P que usam a predição para a frente para tramas I ou P; e tramas B, que usam predição para a frente e/ou para trás para tramas I ou P. Na realidade, apesar do MPEG-1 não especificar o uso do termo escalabilidade, a especificação das tramas bidireccionais (tramas B) criou um novo conceito, pois possibilitou o descarte destas tramas sem impacto na integridade da codificação, exceptuando a perda do número de imagens por segundo.

A escalabilidade foi introduzida pela primeira vez pelo MPEG-2 Vídeo, com a definição de um perfil com escalabilidade em três dimensões: temporal, qualidade (SNR) e espacial. O perfil foi um fracasso essencialmente devido à falta de aplicações, suporte de rede e a existência duma rudimentar integração com o MPEG-2 Sistema, entre outras razões.

Uma nova tentativa de introduzir ferramentas de escalabilidade foi no MPEG-4 Visual, mas mais uma vez, a falta de interesse da indústria nesta problemática, fez com que este tipo de ferramentas não fosse adoptado. O segundo passo para a inovação foi a inclusão do *Fine Granularity Scalability* (FGS) (vídeo com várias camadas, as quais poderiam ser descartadas) dentro do MPEG-4 Visual por volta do ano 2000. A vantagem do FGS era o controlo de congestão, que permitia a realização de uma adaptação “fina” da largura de banda. Este sistema de suportar camadas nunca foi muito apoiado, daí o FGS nunca ter sido muito desenvolvido, por falta de especificação deste sistema de camadas e falta de apoio da indústria.

Após o desenvolvimento do H.264/AVC, a *Joint Video Team* (JVT) considerou uma extensão dessa norma, que designou por H.264/SVC<sup>10</sup>, a qual será pormenorizada na próxima secção. Também será detalhada uma extensão do MPEG-4, o MPEG-J<sup>11</sup>, que permite de certa forma escalabilidade.

---

<sup>9</sup> São acessos a uma determinada parte do vídeo num intervalo limitado de tempo.

<sup>10</sup> Para mais informação ver [10].

<sup>11</sup> Junção do MPEG-4 com a tecnologia JAVA, para informação mais detalhada ver ISO/IEC 14496-21.

## 2.1.2 H.264/SVC

No H.264/SVC, um sinal de vídeo é representado por uma camada base e uma ou mais camadas superiores, em oposição às normas antecedentes que tinham uma estrutura linear (apenas com uma camada) [9]. A camada base do H.264/SVC é compatível (*backward compatible*) com o H.264/AVC. As camadas superiores oferecem um aumento da qualidade do vídeo, por combinação de três factores: resolução temporal – *Frame Per Second* (FPS), resolução espacial e qualidade do conteúdo do vídeo (SNR).

Na Figura 4, encontra-se um exemplo de como são definidas as camadas no H.264/SVC, a qual tem três resoluções espaciais: *Common Intermediate Format* (CIF)<sup>12</sup>, *Quarter CIF* (QCIF)<sup>13</sup>, e *4 times CIF* (4QCIF)<sup>14</sup>; três resoluções temporais: 7,5 FPS, 15 FPS e 30 FPS; duas camadas SNR: B0 e B1. A camada com menos requisitos (camada base) tem T0, S0 e B0. Com estes três campos é possível definir-se diferentes camadas com diversos requisitos de largura de banda.

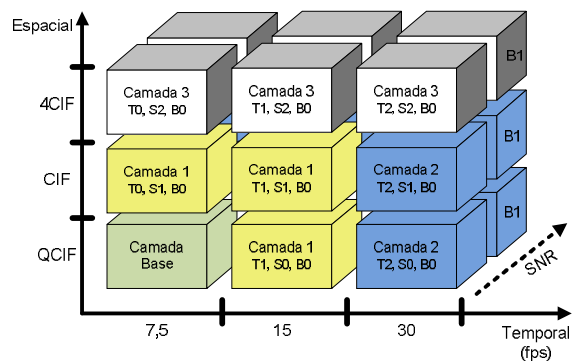


Figura 4 – Distribuição das camadas pelos três factores.<sup>15</sup>

Como ilustrado na Figura 4, ao se remover aumentos seja temporais, de qualidade ou espaciais, o fluxo de vídeo obtido é válido para um decodificador H.264/SVC. Por exemplo, removendo todas as camadas temporais maiores que 15 FPS (camada dois e três) e removendo a camada espacial mais elevada (4CIF), o fluxo de vídeo obtido constituído apenas pela camada base e camada um, seria válido.

A escalabilidade nos conteúdos é fornecida por uma camada de abstracção na codificação do vídeo, designada por unidade *Network Abstraction Layer* (NAL). Este conceito foi primeiramente introduzido pelo H.264/AVC, com o intuito de organizar (especificar o conteúdo codificado) a codificação de bits de vídeo, tendo apenas um *byte* de cabeçalho. O H.264/SVC utiliza também este conceito de

<sup>12</sup> A imagem tem a resolução de 352 x 288.

<sup>13</sup> A imagem tem a resolução de 176 x 144.

<sup>14</sup> A imagem tem a resolução de 704 x 576.

<sup>15</sup> Baseado na figura presente em [11].

unidade NAL, mas com um cabeçalho de três bytes<sup>16</sup>. As unidades NAL são depois encapsuladas em pacotes ou outras estruturas, com a finalidade de serem transmitidas sobre uma rede de pacotes ou serem guardadas em ficheiros estruturados. Com estas unidades é possível descartar pacotes ao nível da camada de transporte do vídeo. As vantagens do uso das unidades NAL são:

1. Fornecem uma abstracção conveniente dos dados do vídeo para diferentes hierarquias de protocolos de transporte (como por exemplo, o *Real-time Transport Protocol* (RTP)<sup>17</sup>) ou para sistemas de armazenamento.
2. Para sistemas baseados em pacotes, como o RTP, a camada de transporte pode ser identificada pelos limites da unidade NAL sem usar *startcodes*, ou seja, reduz o *overhead* desses bits.
3. A estrutura da unidade NAL fornece uma capacidade de extensão flexível usando novos tipos de unidade NAL.

No H.264/SVC, a escalabilidade espacial é generalizada (ao contrário dos seus antecessores), permitindo que as camadas superiores cortem ou aumentem a camada base. É possível que uma camada superior contenha partes (no domínio espacial) que não tenham sido preditas da camada base, assim esta não tem de cobrir a mesma área espacial que a cena original. Isto só é possível porque o H.264/SVC utiliza um conceito conhecido como *single-loop decoding*. Com este novo conceito no máximo dois ciclos de descodificação são feitos, um como o ciclo de descodificação principal e o segundo para manter a integridade da cadeia de predição das imagens chave. Nas tecnologias anteriores de escalabilidade necessitavam de um ciclo de descodificação por cada camada espacial ou de qualidade, o que com várias camadas produzia um atraso muito elevado na descodificação (receptor).

Em H.264/SVC a comunicação entre o *media server* e o cliente é feita através de um *media gateway*, este elemento “inteligente” é muitas vezes referido como MANE [8]. Desta forma, o MANE aquando da adaptação dos fluxos RTP, pode realizar as seguintes operações sobre os pacotes: simplesmente descartar o pacote completo; descartar uma ou mais unidades NAL; fazer *forwarding*; agregar ou fragmentar pacotes<sup>18</sup>. Os MANEs terminam as sessões RTP.

---

<sup>16</sup> Detalhado no Anexo D – Estrutura da unidade NAL do H.264/SVC  
Anexo D – Estrutura da unidade NAL do H.264/SVC.

<sup>17</sup> Definido no RFC 3550.

<sup>18</sup> Para saber mais informações, sobre agregação e fragmentação de unidades NAL ver [8].

### 2.1.3 MPEG-J

O MPEG-J foi definido como parte do MPEG-4 Sistema, permitindo aos criadores de conteúdos embutir algoritmos para controlo de áudio e vídeo. Estes algoritmos são implementados na linguagem de programação JAVA™, sendo os bytes codificados transportados como os fluxos de áudio e vídeo [12].

O MPEG-4 define um mecanismo de compressão para objectos de áudio e vídeo. A parte dos sistemas do MPEG-4 que especifica como os objectos são descodificados de um fluxo binário, e apresentados sincronamente com outros objectos, é designado por *Presentation Engine*. A norma também define uma *Java Application Engine* que especifica como as aplicações são recebidas, descodificadas e executadas no terminal do cliente. Esta parte do MPEG-4 é designada por MPEG-J.

O MPEG-J define um conjunto de *Java API* (JAPI) para aceder e controlar os conteúdos nos terminais. Uma aplicação MPEG-J, que usa estas JAPIs pode ser embutida num conteúdo como uma MPEG-J *elementary stream* similar aos fluxos de vídeo e áudio. É definido um mecanismo de entrega para transportar o código e objectos JAVA que fazem uma aplicação MPEG-J. Algoritmos complexos de controlo podem ser introduzidos nessa aplicação para incluir características baseadas na variação com o tempo das condições de recepção dos terminais.

Usando a JAPI, as aplicações têm acesso programático à cena, rede e recursos descodificados, podendo os clientes adaptar, de uma forma elegante, os conteúdos a uma variação estática ou dinâmica das condições da rede.

O valor acrescentado pelo MPEG-J é a resposta do algoritmo à interactividade do utilizador, permitindo que os criadores de conteúdos possam incluir aplicações MPEG-J para responder a *triggers* dos utilizadores, resultando em respostas inteligentes para interacções do utilizador. Deste modo, o mesmo conteúdo pode ser adaptado para clientes com diferentes capacidades computacionais, permitindo inclusive, a monitorização da largura de banda (através da perda de pacotes).

As razões para a não utilização deste standard, podem estar relacionadas com a falta de aceitação por parte do mercado. As STBs necessitavam de ter implementado esta norma para ser possível enviarem informação da rede, algo que não aconteceu.

A não utilização de MPEG-J pelos fabricantes das STBs, pode estar relacionado com a necessidade de alteração das mesmas, de modo a que estas tenham suporte para JAVA™, o que seria um acrescento de carga computacional.

## 2.2 Taxionomia de gestão da *Quality of Service* (QoS)

Na Figura 5, apresenta-se uma taxionomia de classificação de QoS, onde se ilustram diferentes soluções, a aplicar em diversos pontos da rede, nomeadamente: *headend*, núcleo da rede, rede doméstica ou num ponto genérico da rede. Na rede de acesso não há técnicas de adaptação pois nesta parte da rede o sinal é analógico.

No *headend* foram consideradas duas soluções:

1. SLA e SIP – Utiliza o protocolo SIP para negociar os SLAs dos clientes.
2. SVC com MPEG-21/DIA – Conjuga a codificação de vídeo H.264/SVC com o *Digital Item Adaptation* (DIA) do MPEG-21. A ferramenta DIA permite uma descrição das diferentes camadas do H.264/SVC.

No núcleo da rede, o projecto *Dynamic and distributed Adaptation of scalable multimedia content in a context-Aware Environment* (DANAÉ) permite a adaptação num nó intermédio da rede, recorrendo a informação de contexto recebida dos dispositivos dos clientes.

Na rede doméstica o *Audio-visual Streaming platform for domestic Leisure and Security* (ASTRALS) baseia-se num *gateway* residencial que permite uma adaptação ao tipo de dispositivo do cliente.

Num ponto genérico da rede o projecto *Network-Integrated Multimedia Middleware* (NMM) é um *middleware* que pode ser implementado em qualquer ponto da rede. Todos os dispositivos que queiram ser parte integrante da solução têm de implementar este *middleware*.

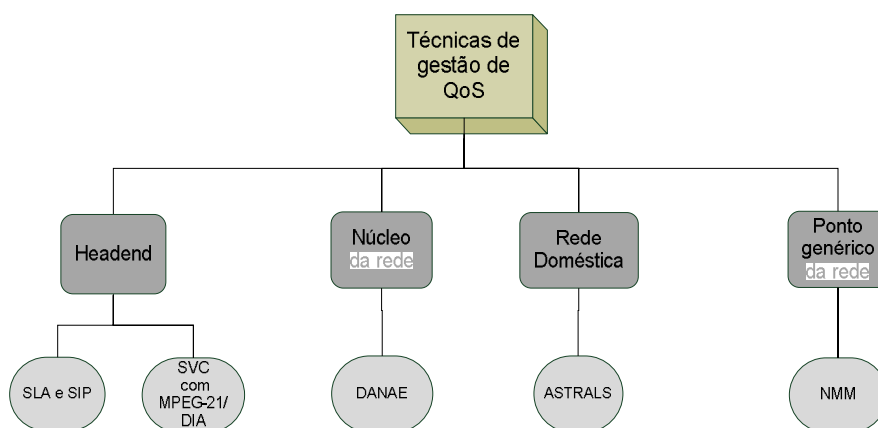


Figura 5 – Taxionomia das soluções para a gestão de QoS.

## 2.3 Técnicas de adaptação

### 2.3.1 *Headend*

#### 2.3.1.1 *SLA e SIP*

Em IPTV é fundamental permitir aos utilizadores ajustarem dinamicamente os seus serviços, de acordo com o tipo de dispositivo que utilizam e com o tipo de acesso que dispõem (sem fios ou com fios) [13]. Esta negociação baseia-se num *Service Level Agreement* (SLA), que basicamente é um contrato entre o fornecedor do serviço e o cliente.

O protocolo *Session Initiation Protocol* (SIP) é um protocolo de sinalização para controlar várias sessões multimédia, que fornece um modo para estabelecer sessões de voz, vídeo e troca de mensagens entre dispositivos. O protocolo SIP não contém nenhum mecanismo para controlar nem a largura de banda nem a latência disponível e as redes IP também não fornecem essa informação. Como tal, é usado o protocolo *Session Description Protocol* (SDP)<sup>19</sup> como corpo da mensagem SIP. O SDP aparece como um protocolo de suporte estável para colocar parâmetros da ligação, relacionados com o QoS. A camada de sessão SDP é um portador adequado para informação de QoS, na forma de atributos SDP (usado também no RFC 3312, onde se negocia com sessões, recorrendo ao uso de SIP, como protocolo de sinalização e SDP para descrever os parâmetros da sessão).

Em [13] foi desenvolvido um projecto, em que o objectivo era juntar estes dois “mundos” descritos acima, isto é, negociar os SLAs recorrendo ao uso de SIP, oferecendo desta forma, uma solução de IPTV adaptativa. Para tal, são utilizados quatro parâmetros fundamentais que dão a indicação do estado da rede: largura de banda, *delay*, *jitter* e perda de pacotes. Contudo um cliente não tem de saber o que estes parâmetros significam e o modo como estes lhe afectam o serviço, logo o SLA não pode ser negociado, apenas com base nestes parâmetros. O modo encontrado foi atribuir níveis de qualidade, associando os três parâmetros de QoS a níveis de *Quality of Experience* (QoE)<sup>20</sup>, usando as *IP QoS Classes* do ITU-T Y.1541.

Como descrito em [14], uma das formas de medir o QoE, é através de uma análise subjectiva dos observadores, que classificam o vídeo segundo uma escala, tal como o *Mean Opinion Score* (MOS)<sup>21</sup>.

O procedimento de negociação dos SLAs começa com o cliente a enviar uma mensagem INVITE para pedir um determinado serviço ao fornecedor de serviços. A mensagem é recebida pelo servidor através dum *proxy*. O servidor avalia o pedido e a sua disponibilidade; caso seja capaz de satisfazer o pedido do cliente, envia uma mensagem SIP OK, para notificar que o serviço pedido é aceite e procede às configurações necessárias à satisfação do QoS requisitado pelo utilizador. Se o utilizador receber uma resposta negativa, então pode voltar a efectuar o pedido com um nível de qualidade inferior.

---

<sup>19</sup> Protocolo definido no RFC 2327.

<sup>20</sup> Definido em [14].

<sup>21</sup> Definido em [14].



O dinamismo desta solução na adaptação dos conteúdos aos clientes constitui um ponto importante, na medida em que o cliente pode ajustar o seu nível de serviço. Outra mais valia desta solução está relacionada com a interligação entre QoS e QoE, isto é, o cliente limita-se a pedir um nível de qualidade e o servidor associa o nível pedido aos correspondentes valores de QoS, verificando se pode ou não servir o pedido, ou seja, o cliente não tem de pedir largura de banda, tempo máximo de atraso, etc.

O ponto desvantajoso desta solução prende-se com a escalabilidade. O sistema não se adapta por si só, o cliente é que pede uma determinada qualidade, no caso de ser rejeitada ele tem que tentar novamente pedir um serviço com menor qualidade. Este processo de negociação pode ser lento, caso o cliente não tenha a percepção do nível de qualidade que a rede é capaz de suportar. Para além disso, o crescimento cada vez maior de dispositivos que suportam IPTV e aumento de clientes que utilizam este serviço podem fazer com que as negociações de qualidade “inundem” a rede.

### **2.3.1.2 SVC com MPEG-21/DIA**

A solução proposta, usa a ferramenta descritiva *AdaptationQoS* do MPEG-21<sup>22</sup> DIA para adaptar o fluxo H.264/SVC [15]. O *AdaptationQoS* descreve a relações entre os requisitos dos recursos (por exemplo, o ritmo binário), operações de adaptação e a correspondente qualidade. Os operadores de adaptação para vários tipos/formatos de conteúdos são fornecidos no esquema de classificações do MPEG-21 DIA. Os operadores considerados são: *SpatialLayers* indica o número de camadas espaciais superiores a serem descartadas do fluxo SVC; *TemporalLevels* indica o número de camadas temporais superiores a ser descartadas; *QualityReduction* indica a fracção de camadas SNR superiores a serem truncadas.

Na Figura 6 está ilustrado o típico sistema de adaptação, contendo basicamente o *Decision engine* e o *Scaling engine*. O *Decision engine* realiza a descrição do *AdaptationQoS* e outros metadados, tais como, o ritmo binário e a informação do *bitstream*; só depois decide as instruções para adaptação (valores de operação). De notar que a descrição das ferramentas para suporte do processamento do *Adaptation engine* é do âmbito do MPEG-21 DIA e a descodificação é do âmbito do H.264/SVC.

---

<sup>22</sup> Definido em ISO/IEC 21000.

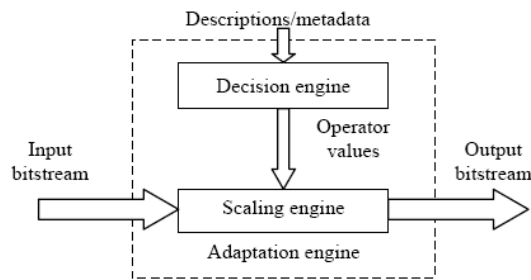


Figura 6 – Arquitectura da máquina de adaptação.<sup>23</sup>

O esquema que foi proposto para permitir uma adaptação dinâmica dos conteúdos sobre redes heterogêneas está ilustrado na Figura 7. No terminal, o ambiente do utilizador contempla: a capacidade do terminal (capacidades do codificador, propriedades do dispositivo, características I/O), as características do utilizador (sua informação, preferência de uso, histórico, acessibilidade, localização) e as características da rede (capacidades, condições). Estes elementos são representados usando a ferramenta *Usage Environment Description* (UED) (definida no MPEG-21 DIA), sendo posteriormente transmitidos para um servidor que contém o *Adaptation Decision Taking Engine* (ADTE). O ADTE encontra um ponto de adaptação óptimo através das dimensões: espacial, temporal e SNR. O *Dynamic Extractor* é utilizado para adaptar o conteúdo através das unidades NAL (reduzindo ou melhorando a qualidade do vídeo). O UED é transmitido do cliente para o servidor através do protocolo *Real Time Streaming Protocol* (RTSP)<sup>24</sup>. O ADTE para além da informação que recebe do UED, também recebe informação do cliente por *Real-Time Transport Control Protocol* (RTCP)<sup>25</sup>.

O “*Network Emulator*” apresentado na Figura 7, permitiu variar a largura de banda ao longo do tempo. O NIST Net<sup>26</sup> foi o emulador utilizado.

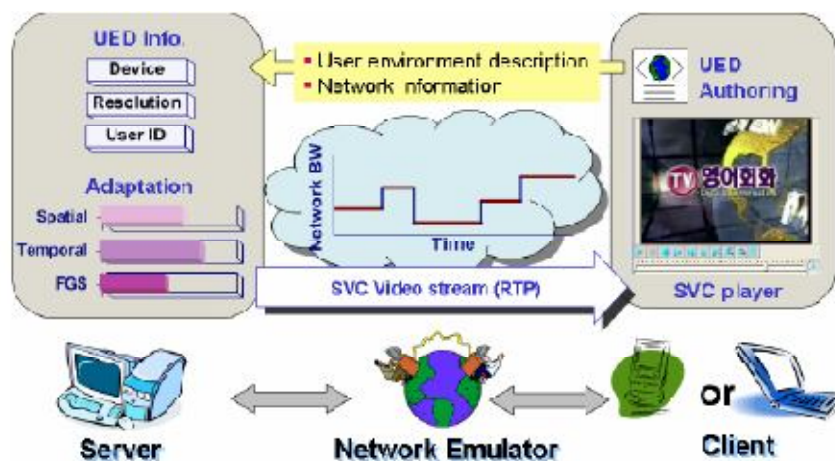


Figura 7 – Cenário de *streaming* adaptativo SVC.<sup>27</sup>

<sup>23</sup> Figura extraída de [15].

<sup>24</sup> Protocolo de controlo, na transferência de dados em aplicações de tempo real. Definido no RFC 2326.

<sup>25</sup> Protocolo de controlo, associado às ligações RTP. Definido no RFC 3550.

<sup>26</sup> Consultado em [21].

O H.264/SVC com informação do MPEG-21, pode tornar-se uma ferramenta bastante poderosa, sendo que nesta solução, o cliente apenas envia as suas características, o servidor é que se adapta consoante o tipo de características do cliente (UED) e estado da rede (RTCP). É uma vantagem em relação à solução anterior (secção 2.3.1.1), na qual necessitava de constante interacção entre o servidor e o cliente, para se encontrar a melhor adaptação.

Contudo, o elemento responsável pela adaptação (*Dynamic Extractor*) encontra-se dentro do servidor. No caso de existirem muitos utilizadores o servidor poderá não ter capacidade de resposta. Convém também salientar que o RTCP tem graves problemas de escalabilidade, uma vez que utiliza dois tipos de pacotes: os *sender reports*, que são usados pelos transmissores de conteúdos (servidor) para enviar informação para os utilizadores e os *receiver reports* que são usados pelos receptores de conteúdos. O problema está relacionado com os *receiver reports*, pois estes são enviados de um receptor para todos os membros do grupo, surgindo deste modo alguns problemas [16]:

- Em grandes grupos *multicast*<sup>28</sup> que utilizem o RTCP, o feedback é enviado muito raramente, isto porque, o intervalo de dois *reports* cresce linearmente com o tamanho do grupo. Assim com o aumento do tamanho do grupo, o intervalo do RTCP aumenta resultando em frequentes *reports* com a diminuição do significado e valor do *feedback*. Então como resultado da demora no *feedback*, não há informação da experiência dos clientes durante curtos intervalos de tempo.
- É necessário calcular o tamanho do grupo. Cada elemento de um grupo tem de guardar um contador de membros diferentes que ouve durante a sessão. Pode existir problemas no armazenamento.
- *Receiver reports* são enviados em *multicast* para todos os membros na sessão, sendo principalmente usados pelos transmissores para adaptar os conteúdos. Há muito pouco benefício ter cada membro a enviar os seus *receiver reports* para todos os outros membros no grupo que não são transmissores. Além disso, um membro que tenha uma ligação de baixa largura de banda, preferirá receber mais dados ao invés de receber informação de controlo dos *receiver reports* que têm pouco benefício para ele.
- Se demasiados elementos do grupo fizerem um *join* ao mesmo tempo, não é possível obter uma estimativa correcta do tamanho do grupo. Cada membro, primeiro estima que o tamanho do grupo é um, e então todos os *reports* RTCP são enviados dentro do intervalo fixo inicial. Consequentemente, a congestão pode ocorrer na rede e especialmente nas ligações de baixo débito de alguns membros.

---

<sup>27</sup> Figura extraída de [15].

<sup>28</sup> Descrito detalhadamente no Anexo A – Tecnologias de suporte.

- Se muitos utilizadores deixarem o grupo ao mesmo tempo, o resultado é uma “inundação” de pacotes *Bye* que podem causar congestão.

## 2.3.2 Núcleo da rede

### 2.3.2.1 DANAE

Em [17] é apresentado como parte integrante do projecto DANAE, uma arquitectura de um mecanismo para adaptação de multimédia, que suporta decisão dinâmica e adaptação ao dispositivo. Utiliza o H.264/SVC para a adaptação, o qual permite escalabilidade nos conteúdos e a sua consequente adaptação. O mecanismo de adaptação é baseado no MPEG-21 DIA, que foi estendido para suportar adaptações dinâmicas. Adaptação dinâmica é entendido como sendo a adaptação de recursos multimédia, durante a sessão de *streaming*. Isto não exclui a adaptação no início da sessão, que é considerado como um sub tópico da adaptação dinâmica. Este conceito de adaptação pode ser aplicado, por exemplo, a um telemóvel de uma rede *Universal Mobile Telecommunications* (UMTS), podendo mudar para uma largura de banda maior, sempre que for detectado um novo ponto de acesso (por exemplo, entrar num café e usar o ponto de acesso local).

Na Figura 8, está ilustrado o cenário de adaptação distribuída. Este sistema baseia-se num cliente que envia um pedido ao servidor para ver um determinado canal. O servidor selecciona o nó de adaptação mais apropriado (por exemplo, baseado na sua localização) e envia para este o pedido do cliente. O nó de adaptação responde ao pedido e recebe informação de contexto do utilizador (por exemplo, as capacidades do terminal e condições da rede). Compara então a informação de contexto deste utilizador, com a de todos os outros utilizadores que ele serve naquele canal, e decide se a qualidade do conteúdo recebido no servidor é suficiente ou não. Se não for suficiente, é feito um pedido ao servidor de um fluxo com melhor qualidade. O conteúdo é adaptado para todos os utilizadores de uma forma genérica no nó de adaptação, baseando-se este nos metadados que lhe são enviados.

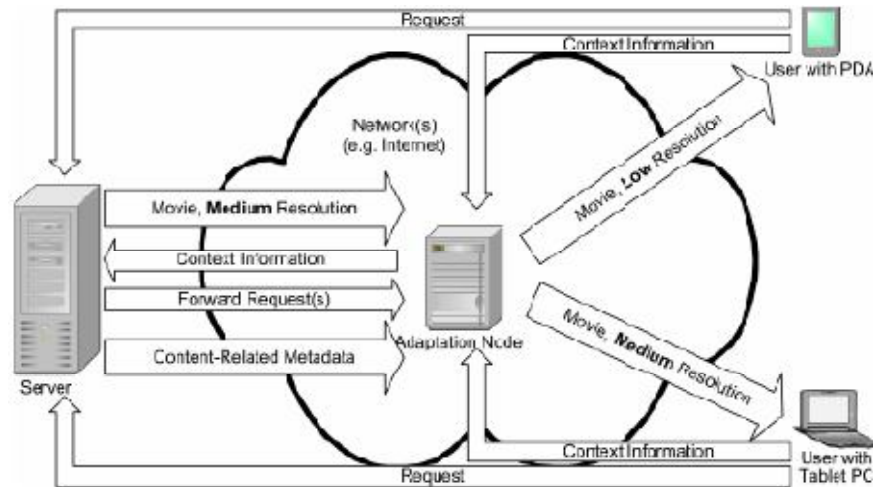


Figura 8 – Cenário de adaptação distribuída.<sup>29</sup>

A arquitectura do *Adaptation Node* pode ser dividida em três módulos: o *adaptation engine*, o *context merging* e o *context aggregation*. Este último é responsável por colectar informação de contexto do nó cliente e fornecer esta informação a outros módulos. Baseando-se nesta informação o *adaptation engine* pode decidir que tipo de adaptação deverá aplicar ao conteúdo multimédia que será transmitido para o cliente.

O módulo *context merging*, compila um conjunto de descritores de contexto, cada um dos quais indica a qualidade máxima, que qualquer um dos terminais (a que o nó de adaptação se encontra ligado) pode consumir. A informação com as adaptações suportadas é depois enviada para o servidor. Baseado na combinação de informação de contexto, o servidor pode adaptar o conteúdo para o nível de qualidade pedido pelo nó de adaptação.

O conteúdo adaptado corresponde à mais elevada qualidade dos conteúdos multimédia, relativamente aos pedidos de todos os terminais ligados ao nó adaptação. O servidor pode assim entregar ao nó de adaptação, este conteúdo, juntamente com metadados e descrições de duas ferramentas do MPEG-21 DIA, como o *AdaptationQoS* e o *Universal Constraints Descriptor (UCD)*, usadas para adquirir informação (por exemplo, o tamanho do ecrã ou o estado da ligação de rede). O *AdaptationQoS* já foi descrito na secção 2.3.1.2. O UCD pode complementar a informação do *AdaptationQoS* pela especificação de restrições (impostas por outras fontes ou pela própria rede). Uma função de optimização pode ser declarada no UCD para controlar a selecção da melhor opção de adaptação.

Subsequentemente, o nó de adaptação replica e fornece o conteúdo para cada terminal ligado. Assim, cada cliente recebe a máxima qualidade possível, sob certas restrições de um contexto específico de rede, utilizador e terminal.

<sup>29</sup> Figura extraída de [17].

O *adaptation engine* pode estar dentro do servidor, ou num nó de adaptação na rede. Caso esteja num elemento da rede, este tem interfaces para comunicar com o servidor e extrair deste os conteúdos e respectivos metadados. Esta informação adquirida é enviada para um módulo dentro do *adaptation engine*, que determina as configurações dos parâmetros de adaptação óptimos de acordo com as restrições de uso do ambiente, como a largura de banda disponível, a resolução do monitor do dispositivo, entre outros. Essas restrições do uso do ambiente são guardadas numa base de dados central (que se encontra fora do *adaptation engine*), excepto a mudança dinâmica da largura de banda. Para isso existe uma outra componente do mecanismo de controlo de adaptação que regularmente sonda o canal de transmissão para obter as estimativas da largura de banda.

Existe ainda um outro módulo dentro do *adaptation engine* que executa a adaptação, de acordo com os parâmetros óptimos e os conteúdos recebidos. Aquando das alterações das condições da rede durante uma sessão, é necessário que o *adaptation engine*, descrito anteriormente, possa dinamicamente reagir à mudança de informação de contexto, para gerar uma nova decisão de adaptação e aplicar esta decisão ao resto de recursos a serem transmitidos.

A escolha dos parâmetros óptimos requer pelo menos um conhecimento das possibilidades de adaptação, complementadas preferivelmente pelas características do recurso e da qualidade escolhida. É necessário para tal, um conjunto de valores descritivos do contexto. A adaptação de vídeo com o H.264/SVC, fornece total suporte às combinações espacial, temporal e SNR, que resultam num certo ritmo binário, enquanto que as características dos fluxos adaptados podem ser descritos pelo resultado médio do *Peak SNR* (PSNR).

Esta solução tem a vantagem de ter flexibilidade na adaptação, podendo ser realizada no servidor ou num nó intermédio. Também permite fazer uma adaptação dinâmica e distribuída. No caso específico do IPTV a adaptação dinâmica não é muito importante, pois as condições da rede não mudam tanto como num ambiente em que existe grande mobilidade.

Por outro lado, o pedido do cliente vai sempre até ao servidor, ou seja, no caso do nó intermédio ter capacidade para fazer a adaptação, o tempo do cliente pedir ao servidor o canal e o servidor reencaminhar para o nó de adaptação, é tempo perdido. De salientar também, que o processo de escolha da qualidade é bastante demorado, visto que o nó de adaptação tem de receber a informação de contexto do cliente e comparar com todos os outros clientes que ele serve naquele canal, seleccionar a melhor qualidade e dizer ao servidor para adaptar o vídeo com essa qualidade.

### 2.3.3 Rede doméstica

#### 2.3.3.1 ASTRALS

O ASTRALS é um projecto que se foca na escalabilidade da codificação de áudio e vídeo, guardando e distribuindo os conteúdos no domicílio do cliente via rede sem fios [18]. O objectivo é implementar soluções escaláveis que permitam uma inovação no campo audiovisual e serviços personalizados, realizando uma adaptação do vídeo tanto na rede, como pelos dispositivos heterogéneos do cliente desde dispositivos com requisitos computacionais baixos como um *Personal Digital Assistants* (PDA) até aos com elevados requisitos como o *home cinema*.

O poder de adaptação nesta arquitectura de rede, reside no *Residential Gateway*. Hoje em dia, a maior parte destes elementos são desenhados como *routers* com múltiplas interfaces com capacidades de baixo processamento. O *Residential Gateway* da ASTRALS é mais do que um simples *router*, fornecendo: melhoramentos na codificação audiovisual; *transcoding*; armazenamento e distribuição de conteúdos baseada em perfis; extensões optimizadas de banda larga sem fios (o ASTRALS especificou e criou um protótipo para uma optimização do *streaming*). Utiliza o H.264/AVC ou o H.264/SVC, com a característica da degradação dos conteúdos armazenados, em função das preferências dos clientes. Realiza uma eficiente adaptação dos conteúdos, baseado num feedback da camada de ligação de dados, *Media Access Control* (MAC) para as redes sem fios, e da camada de rede. Utiliza protocolos SDP (descrito em 2.3.1.1) e *Session Announcement Protocol* (SAP)<sup>30</sup> para *session announcement* e RTSP para *session handling*. Utiliza *Secure RTP* (SRTP)<sup>31</sup> para garantir a confidencialidade e protecção de ataques na comunicação sem fios, cifrando os pacotes RTP e RTCP. Através do protocolo RTCP é possível perceber quantos utilizadores estão presentes e estabelece-se um ritmo para a entrega de *reports* para evitar que seja alocada demasiada largura de banda pelas mensagens de controlo.

Esta solução tem a vantagem de não ter toda a carga computacional associada ao servidor, pois a adaptação é feita no *Residential Gateway*. É possível fazer uma adaptação dos conteúdos no núcleo da rede ou num *media gateway* que se designa por MANE, isto se for utilizado a norma de codificação de vídeo H.264/SVC.

Novamente surge o problema da escalabilidade relacionada com os *receiver reports* do RTCP, situação reportada na secção 2.3.1.2. Também convém notar que é necessário um *Residential Gateway* por cliente (casa), o qual poderá não querer pagar um custo acrescido por este equipamento com uma maior capacidade computacional. Convém ainda referir que o equipamento do cliente (no caso de televisão a STB) tem de suportar comunicação sem fios. Apesar de se partir do pressuposto que o vídeo chega em boas condições ao *Residential Gateway*, convém notar que a rede de acesso pode constituir

---

<sup>30</sup> Protocolo que emite anúncios periódicos para um grupo *multicast*, com a descrição SDP da sessão. Definido no RFC 2974.

<sup>31</sup> Perfil de segurança para o protocolo RTP, usado essencialmente para redes móveis. Definido no RFC 3711.

um ponto de estrangulamento, podendo perder-se informação importante do vídeo. Nesta situação, o *Residential Gateway* nada poderá fazer.

## 2.3.4 Ponto genérico na rede

### 2.3.4.1 NMM

O NMM é uma solução *open-source* da empresa Motama resultante de um projecto de investigação da universidade *Saarland* na Alemanha [19]. Parte da premissa que se verifica um aumento progressivo de dispositivos com suporte multimédia, tais como, câmaras *web*, dispositivos de áudio, PDAs, telefones celulares, STB e *Personal Computer* (PC). Baseia-se numa arquitectura de *middleware*<sup>32</sup> distribuída, a qual pode ser executada sobre vários sistemas operativos (por exemplo: *Linux*, *Windows* e *MacOS*). Esta arquitectura pode ser usada para as aplicações de multimédia locais ou para sistemas distribuídos.

Está representado na Figura 9, um sistema NMM e como este permite o acesso a todos os recursos dentro da rede: dispositivos de distribuição multimédia e componentes de *software* podem ser transparentemente controlados/integrados dentro da aplicação. Destaca-se das outras soluções multimédia por ser uma camada de *software* distribuída, onde até um PDA com pouco poder computacional que tem, pode correr tal aplicação. Os recursos para adaptar o conteúdo multimédia ao PDA podem assim estar distribuídos na rede. A distribuição é transparente para os programadores.

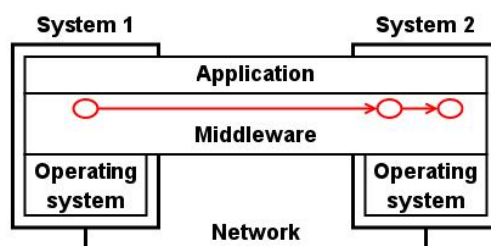


Figura 9 – Sistema NMM.<sup>33</sup>

Nesta arquitectura, todos os dispositivos de hardware e as componentes de software (por exemplo, os codificadores) são designados de nós. Um nó tem propriedades que incluem as portas de entrada e de saída, denominadas de *jacks*. Para ligar dois nós, é necessário que o *jack* de saída de um nó, suporte os mesmos formatos de multimédia do *jack* de entrada do outro nó. Existem vários tipos de nós (é até possível criar novos tipos de nós), interligando-os através de *jacks*, criam um grafo de fluxos.

Na Figura 10, está ilustrado um grafo de um fluxo multimédia, onde o *MPEG Read Node* é um tipo de nó que permite ler o conteúdo de ficheiros MPEG [20]. Este nó utiliza o *jack av/mpegps* para se ligar a

<sup>32</sup> *Middleware* é uma camada de software entre o sistema operativo e a camada aplicacional.

<sup>33</sup> Figura extraída de [19].



outro nó (*MPEG DemuxNode*), que desmultiplexa o conteúdo de áudio e vídeo, tendo dois *jacks* *video/mpeg* e *audio/mpeg* (especificam o tipo de conteúdo) e outros dois adicionais "video0" e "audio0" (são etiquetas para definir os dois ramos do grafo). Este mesmo nó ramifica o grafo em duas partes, vídeo e áudio. No vídeo, os *jacks* *video/mpeg* estabelecem a comunicação com um outro nó, que permite a descodificação do vídeo (*MPEG VideoDecodeNode*), que por sua vez se encontra ligado ao nó que permite visualizar o conteúdo descodificado (*XDisplayNode*), pelo *jack* *video/raw*. No áudio, os *jacks* *audio/mpeg* definem a comunicação com o nó de descodificação do áudio (*MPEG AudioDecodeNode*), o qual se encontra ligado ao nó de audição do conteúdo descodificado (*Playback Node*), pelo *jack* *audio/raw*. No final deste grafo ainda poderia existir um nó (*Sink Synchronizer*) ligado ao *XDisplayNode* e ao *Playback Node*, para sincronizar o áudio e o vídeo.

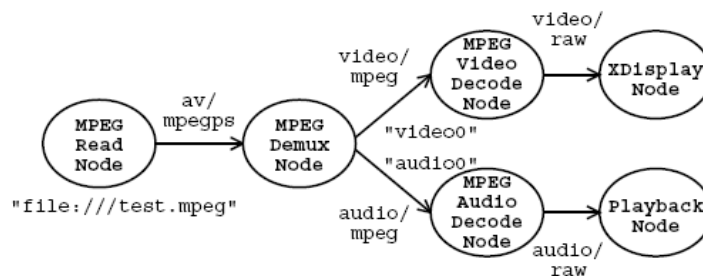


Figura 10 – Grafo de fluxo de conteúdo multimédia.<sup>34</sup>

O registo dos serviços no NMM permitem descobrir, reservar e instanciar nós disponíveis localmente ou remotamente. Em cada máquina, um *registry server* administra todos os nós disponíveis neste sistema, guardando uma descrição destes (tipo de nó, nome, interfaces que fornecem e os formatos de entrada e saídas suportadas). A aplicação usa o *registry client* para fazer pedidos a *registry servers* locais ou remotos.

Na adaptação do fluxo de vídeo o NMM permite vários tipos de escalabilidade dos quais é importante referir: temporal, reduzindo o número de amostras no domínio do tempo, por descarte de tramas de vídeo; espacial, alterando a resolução do vídeo. Também tem a possibilidade de em cada nó, recodificar os conteúdos alterando características da própria imagem (por exemplo, gama de cores).

Esta solução tem a vantagem de permitir uma grande interoperabilidade com os sistemas operativos existentes. Além disso, adapta a qualidade do vídeo ao tipo de dispositivo que o cliente tem. Outro ponto a favor é o facto de esta solução não estar restringida a um único ponto de adaptação, permitindo maior flexibilidade, podendo até o vídeo ser adaptado em mais do que um ponto da rede.

Por outro lado, este tipo de arquitectura faz com que todos os dispositivos que queiram ser parte integrante da solução tenham de implementar este *middleware*. Para além disso, não suporta as mais recentes normas de codificação de vídeo, como H.264/AVC ou o H.264/SVC. De referir que a

<sup>34</sup> Figura extraída de [20].

escalabilidade tem um “preço” bastante alto, pois o sistema necessita de descodificar o vídeo e codificar novamente, provocando atraso.

### 3 Escolha da técnica de adaptação

Foram estudadas duas alternativas na adaptação do *streaming* de vídeo, com o objectivo de o adaptar à capacidade de recepção do cliente, minimizando tanto quanto possível o impacto visual. Estas duas soluções, apesar de se basearem num conjunto comum de requisitos, constituem abordagens diferentes à adaptação do vídeo. Uma solução, designada de “degradação de conteúdos no MANE”, realiza a adaptação ao nível da camada de rede. Enquanto que a outra solução, intitulada de “degradação de conteúdos no servidor”, efectua a adaptação na camada de aplicação (nas tramas de vídeo).

#### 3.1 Requisitos

No desenho da solução de adaptação de vídeo foram analisados os diversos tipos de requisitos que normalmente são colocados aos sistemas informáticos: desempenho; escalabilidade; segurança e tolerância a faltas. Os requisitos de desempenho foram considerados fundamentais, na medida em que a solução tem de fornecer a cada cliente a qualidade que melhor se adequa às suas características, efectuando a adaptação com a maior rapidez possível, de forma a que o utilizador não tenha a percepção duma má qualidade durante o processo de adaptação. Relativamente à escalabilidade, esta foi considerada como um requisito, na medida em que a solução teria de suportar todos os clientes de IPTV. Uma vez que esta solução se enquadra num cenário de operador (ambiente de IPTV é inteiramente controlado pelo fornecedor de serviços), considerou-se que os requisitos de segurança seriam abordados num âmbito mais geral do serviço de IPTV, não sendo assim incluídos nesta solução, de forma autónoma. A tolerância a faltas, também não foi considerada, uma vez que o objectivo do trabalho é realizar um modelo de uma solução para a adaptação de vídeo e não o desenvolvimento de uma solução para o ambiente de produção.

Os requisitos específicos para o desenvolvimento da técnica de adaptação foram os seguintes:

1. Realizar a degradação dos conteúdos com o mínimo impacto visual. O objectivo não era degradar os conteúdos a qualquer custo, mas sim reduzir o máximo ritmo binário, sem que essa redução da qualidade tenha elevado impacto visual no cliente.

2. A técnica teria de suportar as normas de codificação mais recentes como o H.264/AVC e ser tanto quanto possível, flexível de modo a que pudesse ser aplicada a novas normas que surjam, como o caso do H.264/SVC.

## 3.2 Técnicas de adaptação

Com base nos requisitos descritos anteriormente, foram analisados duas técnicas de adaptação: a degradação dos conteúdos no MANE (depois da codificação) e no servidor (antes da codificação). Ambos os caminhos têm como pressuposto, o envio do vídeo com a melhor qualidade inicialmente e, posteriormente, com a menor capacidade de recepção do cliente, degradam-no de modo a adaptá-lo às suas condições.

### 3.2.1 Degradação de conteúdos no *MANE*

Esta solução de adaptar os conteúdos no MANE, contempla apenas a adaptação a nível da camada de rede [35]. Caso esta fosse efectuada a nível do vídeo, ter-se-iam implicações bastante graves no desempenho e na escalabilidade, uma vez que, para além da recodificação dos conteúdos, seria também necessário calcular novamente alguns campos dos cabeçalhos dos protocolos IP, como por exemplo: os campos *length* e *checksum*, no IP e no *User Datagram Protocol System* (UDP); o campo *timestamp* no RTP. Estes cálculos iriam provocar um atraso significativo (podendo ter como consequência as imagens não chegarem a tempo de serem visualizadas) e/ou um aumento substancial da carga computacional.

Com a adaptação na camada de rede, optou-se pela utilização de um MANE transparente, tanto para o servidor como para o cliente. Nestas circunstâncias o servidor limita-se a enviar o fluxo para o cliente e este a visionar os conteúdos, sem que qualquer elemento intermédio (como o MANE) termine a sessão RTP. Finalizando a sessão RTP, ter-se-ia a vantagem de recodificar (adaptar) o conteúdo de vídeo e criar um novo fluxo personalizado para cada cliente. Contudo, tal solução provocaria um atraso que poderia ser excessivo e conseqüente impacto visual no cliente. Para além disso, a escalabilidade era diminuta, uma vez que o conteúdo teria de ser adaptado para cada cliente, necessitando de elevados recursos computacionais neste elemento de rede.

Não terminando a sessão RTP, perde-se a possibilidade de realizar uma “adaptação fina” dos conteúdos de vídeo para cada cliente, num elemento intermédio da rede, sendo apenas possível fazer

cortes ao nível da camada de rede. Mas por outro lado, não sobrecarrega tanto o MANE, nem produz atrasos excessivos do vídeo.

É importante referir que uma trama de vídeo engloba quase sempre vários pacotes IP. De modo a perceber que impacto tem a predição temporal foram considerados descartes no início e no fim do *Group of Pictures* (GOP)<sup>35</sup> de tramas completas. Por outro lado, também foi analisado o descarte do último pacote IP no final das tramas maiores e nas mais pequenas, de modo a perceber qual o tipo de descarte que afectava mais a qualidade da imagem.

Estes descartes selectivos tinham o objectivo de identificar a métrica que provocava menor impacto visual, analisando no final se essa métrica é de qualidade aceitável ou não. De salientar que o objectivo é adaptar o vídeo, daí que não tenham sido contemplados outros tipos de conteúdos, como legendas ou áudio.

### 3.2.1.1 Métricas de descarte

As métricas de descarte selectivo consideradas foram as seguintes:

- Métrica do último pacote IP: Descarta o último pacote IP de uma trama P, com um determinado tamanho. Foram consideradas variantes, isto é, o descarte do último pacote IP em tramas P maiores e em tramas P menores. Na Figura 11, está representado o descarte do último pacote IP, nas tramas P mais pequenas (segunda e quarta trama do GOP).



Figura 11 – Exemplo da métrica do último pacote IP, com descarte nas tramas mais pequenas.

- Métrica de trama: Descarta todos os pacotes IP possíveis de uma trama P. As variantes desta métrica consideradas, foram o descarte no início e no fim do GOP. Na Figura 12, está exemplificado o descarte de duas tramas que se encontram mais à direita no GOP.



Figura 12 – Exemplo da métrica de trama com descartes no final do GOP.

Nas métricas de descarte apenas foi considerado, vídeos com tramas *Instantaneous Decoding Refresh* (IDR) e P. Contudo as métricas descritas podem ser generalizadas e serem aplicadas às tramas I e B. A razão do não uso deste tipo de tramas foi a simplificação das métricas. Assim as duas métricas

<sup>35</sup> Conjuntos de imagens codificadas que criam um movimento, começam sempre com uma trama I, terminando na trama anterior à trama I seguinte.

implementadas partem do pressuposto que o descarte apenas se realiza em tramas P. A razão pela qual o descarte não é efectuado nas tramas IDR deve-se ao facto de todas as tramas P de um GOP terem como predição a trama IDR. Deste modo, o descarte deste tipo de trama, significaria que todas as tramas P desse GOP seriam afectadas, o que provocaria um impacto muito grande na qualidade do vídeo.

### 3.2.1.2 Testes

Foram realizados diferentes testes, de modo a identificar a melhor métrica de descarte em pacotes IP. A descrição dos testes está ilustrada na Tabela 1.

**Tabela 1 – Descrição dos testes realizados.**

Identificador do Teste	Descrição
1	Análise do impacto visual de cada uma das métricas de descarte selectivo de pacotes, para perdas de pacotes idênticas. Na métrica do último pacote IP, descartando o último pacote IP de cada trama P, comparou-se o descarte em tramas mais pequenas, com o descarte em tramas maiores. Na métrica de trama, comparou-se respectivamente o descarte de tramas P mais à esquerda e mais à direita do GOP. Por fim, comparou-se os melhores tipos de descarte das duas métricas.
2	Combinou-se a melhor opção (obtida no teste um), com as opções de descarte da outra métrica.
3	Comparou-se a melhor métrica obtida no teste 2, com o descarte aleatório para simular uma rede com perdas. Para o descarte aleatório foi usado o emulador <i>NIST Net</i> , no <i>softRouter</i> . Verificou-se para a melhor métrica o limite a partir do qual as perdas se começaram a notar.

Na classificação dos testes, foi avaliada a performance do sistema (“extremo-a-extremo”) na perspectiva do utilizador, também designado por avaliação QoE (definido em 2.3.1). O método subjectivo foi o modo escolhido para qualificar a técnica de adaptação, uma vez que é considerado o mais preciso e intuitivo na medição da qualidade do vídeo (como descrito em [14]).

Embora neste método fosse aconselhável ter um grupo significativo de pessoas a analisar o vídeo ao mesmo tempo, para a visualização e classificação do mesmo conteúdo, tal não foi possível por se tratar de um projecto desenvolvido numa empresa. Assim a análise do vídeo, foi feita por cinco pessoas, que o visualizaram e classificaram de modo independentemente, segundo os diferentes níveis da Tabela 2.

**Tabela 2 – Classificação dos valores de QoE.**

Valor do QoE	Descrição
0	Não mostra imagem.
1	Vídeo com muitas paragens de movimento, conteúdo bastante degradado.
2	Vídeo degradado, podendo notar-se pequenas paragens de movimento.
3	Vídeo com frequente efeito de bloco, raramente ocorrem perdas.
4	Vídeo com raro efeito bloco.
5	Vídeo sem perdas.

Foi permitido dar o valor intermédio entre dois níveis (por exemplo: 1,5; 2,5). É importante referir que a Tabela 2, foi criada com base nos níveis definidos no MOS – estabelecido em [22]. Contudo a tabela foi adaptada, com a introdução do nível 0, para quando não é visualizada a imagem. Este nível foi considerado relevante uma vez que se estava a degradar o vídeo e havia a possibilidade da degradação ser tão elevada que poderia impossibilitar o visionamento da imagem. Também foi descrito com maior detalhe os níveis, de modo a existir um maior rigor na análise dos observadores, não definindo simplesmente como razoável, bom e excelente. Considerou-se que as paragens no vídeo deviam penalizar mais do que o efeito de bloco. Por outro lado, teve de existir um compromisso entre o número de perdas e o número de vezes que estas ocorreram, isto é, um vídeo que tem muito raramente perdas, mas que quando as tem, estas produzem um impacto muito grande no vídeo, deve ter um valor de QoE idêntico ao de um vídeo que tem perdas frequentemente com menor impacto visual.

### **3.2.1.3 Ambiente de Teste**

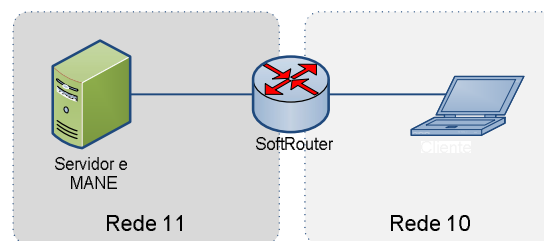
A *Local Area Network* (LAN) da empresa onde a solução foi desenvolvida, constituiu o ambiente de testes (totalmente controlado). O *streamer* usado foi o *VideoLan Client* (VLC) [23]. Este software é *opensource* e suporta as principais características necessárias para o desenvolvimento da solução, isto é, suporta as mais recentes normas de codificação como é o caso do H.264/AVC. O VLC pode utilizar como tipo de comunicação o *unicast* ou *multicast*, incluindo o *Internet Group Management Protocol* (IGMPv3)<sup>36</sup> e suporta igualmente os protocolos de transporte UDP e RTP. O VLC permite ainda uma recodificação do vídeo em tempo real, permitindo assim alterar o ritmo binário, a resolução espacial e temporal. Contudo, existem outras opções como são os casos do MPEG4IP [24] e do *Darwin Streaming Server* (DSS) [25]. O MPEG4IP apesar de suportar MPEG-4, deixou de ser desenvolvido, daí não

<sup>36</sup> Protocolo de comunicação *multicast*. Definido no RFC 3376.

constituir uma alternativa ao VLC. O DSS por ser um projecto menos popular, com menos informação disponível e com muito menos funcionalidades do que o VLC, não constitui uma alternativa a este.

Para todos os testes apenas foi usado um vídeo, que contém partes com muito e pouco movimento, de modo a analisar o seu desempenho, sendo que as partes de maior movimento são mais propícias à quebra desse mesmo movimento e as partes de menor movimento permite uma melhor visualização do efeito de bloco. O vídeo tem uma resolução de 720x576 píxeis, com o perfil *Main* (sem tramas B) do H.264/AVC, nível 5.1, com 25 FPS. A resolução foi a máxima possível para o servidor transmitir o vídeo sem perdas, isto é, sem exceder a sua carga computacional. Foi utilizado o VLC com recodificação em tempo real.

O ambiente de teste, ilustrado na Figura 13, caracteriza-se pela utilização de dois computadores com OpenSUSE 10.3: um como servidor e outro como cliente, cada um usando o VLC 0.8.6f, para enviar e receber os conteúdos de vídeo, respectivamente.



**Figura 13 – Ambiente de teste.**

O *softrouter* é um PC instalado com um sistema operativo CentOS 4.5, com 1 GB de *Random Access Memory* (RAM) e dotado de um *Central Processing Unit* (CPU) com 1.8 GHz. Garante-se que todo o tráfego de vídeo passa por este elemento, pois é o único ponto da rede que interliga as duas redes (recorrendo ao uso de duas placas de rede). O servidor tem 2 GB de RAM e um CPU de 2,8 GHz. O cliente tem 1 GB de RAM e um CPU de 1,73 GHz.

Os testes apenas foram realizados em *unicast* por simplificação. Contudo poderiam ter sido realizados em *multicast*, sendo que, a única diferença seria a recepção dos pacotes IGMP (mensagens *join* e *leave*), que em nada iria alterar as métricas aplicadas, uma vez que as métricas filtram os pacotes IP a descartar, pelo protocolo UDP (conteúdo de vídeo). O modo como foram filtrados os conteúdos de vídeo e realizado o descarte é explicado no Anexo B.

### 3.2.1.4 Resultados

É importante salientar que devido à própria natureza das métricas aplicadas e da codificação feita pelo VLC, em tempo real (as tramas de vídeo são codificadas de acordo com a carga do CPU), é

extremamente difícil obter o mesmo número de perdas. Deste modo, uma métrica com as mesmas características de corte, visionada duas vezes, o mais expectável é que a perda de pacotes seja diferente (apesar de ser um valor próximo). Daí que, idealmente, o vídeo deveria ser observado em simultâneo. Na prática, tal não foi possível pois necessitaria que a empresa parasse para realizar a experiência. Se os resultados com este número de pessoas, não for favorável ao uso destas métricas, então é possível concluir que não é uma boa solução. Contudo, se os resultados indicarem que é uma boa solução, será necessário estar nas condições ideais para a análise de vídeo descritas na secção 3.2.1.2.

Na Tabela 3 encontram-se descritos os resultados do primeiro teste. Como o vídeo foi visionado de modo independente por cinco pessoas, a variação dos valores de QoE assim como dos de perda de pacotes, levaram à necessidade de introduzir o desvio padrão no cálculo de perda de pacotes e do QoE.

Na referida tabela, nas duas colunas mais à esquerda, estão as duas métricas e suas variações respectivamente. A coluna “valores aplicados no descarte” indica as condições que foram aplicadas para se proceder ao descarte. No caso da métrica do último pacote IP, o descarte foi calculado por decremento do número de TS de uma trama (corresponde a um PES<sup>37</sup>) tendo como conhecimento prévio de que cada pacote IP leva 7 TS.<sup>38</sup> Quando uma trama P tinha o número de TS dentro do intervalo especificado, então o último pacote dessa trama era descartado. Por exemplo,  $TS > 6$  e  $TS < 10$ , vai ser descartado o último pacote IP de todas as tramas que têm o tamanho entre 6 e 10 TS. Para a métrica de trama, conta-se o número de tramas e enquanto esse contador estiver no intervalo estipulado, descartam-se todas as tramas. Por exemplo,  $PES > 53$ , se o GOP tiver mais de 53 tramas P, então todas as tramas posteriores à quinquagésima terceira irão ser cortadas. A penúltima coluna mais à direita da tabela (QoE) representa a média das classificações atribuídas pelas pessoas que observaram os vídeos.

---

<sup>37</sup> Este PES começa com o valor zero, ou seja, quando chega a segunda trama P tem o valor um.

<sup>38</sup> Um pacote IP com MTU de 1500 bytes e 1 TS tem um tamanho fixo de 188 bytes, sendo 4 bytes de cabeçalho



**Tabela 3 – Resultados do primeiro teste.**

Métrica	Tipo de descarte nas tramas P	Valores aplicados no descarte	Perda de Pacotes (%)	Desvio padrão da perda de pacote (%)	QoE	Desvio padrão do QoE
Métrica do último pacote IP	Mais pequenas	TS > 6 e TS < 10	0,53	0,02	3,10	0,42
		TS > 6 e TS < 12	1,69	0,03	2,70	0,27
		TS > 6 e TS < 14	4,09	0,08	2,10	0,22
		TS > 6 e TS < 16	6,91	0,27	1,90	0,22
		TS > 6 e TS < 19	9,93	0,17	1,30	0,27
		TS > 6 e TS < 25	20,37	0,18	0,80	0,27
	Maiores	TS > 53	0,51	0,03	3,60	0,42
		TS > 39	1,65	0,04	2,90	0,22
		TS > 30	4,06	0,06	2,50	0,50
		TS > 26	6,85	0,08	2,20	0,27
		TS > 24	9,88	0,05	1,80	0,27
		TS > 18	19,96	0,23	1,40	0,42
Métrica de trama	Mais à esquerda no GOP	PES < 1	1,73	0,17	2,30	0,45
		PES < 5	4,09	0,14	1,90	0,42
		PES < 8	6,97	0,08	1,50	0,50
		PES < 12	9,98	0,11	1,20	0,27
		PES < 27	20,75	0,12	0,70	0,27
	Mais à direita no GOP	PES > 127	0,53	0,02	4,30	0,45
		PES > 120	1,60	0,11	3,80	0,45
		PES > 108	4,19	0,05	3,50	0,35
		PES > 98	6,88	0,04	2,70	0,27
		PES > 90	9,84	0,05	2,50	0,35
		PES > 67	20,16	0,06	1,70	0,27

Por limitação da métrica de “descarte de tramas P mais à esquerda no GOP”, foi impossível obter um valor de perda de pacotes a rondar os 0,50%, daí esta métrica ter menos um valor de análise. Uma vez que obteve uma perda de pacotes de 1,73% para o descarte da primeira trama do GOP (PES < 1).

De uma forma geral, a classificação obtida de QoE apresenta valores inferiores a 3. Para a métrica de descarte do último pacote IP os resultados são melhores quando os pacotes são maiores (ou seja, quando se descarta apenas nos PES maiores). Isto justifica-se porque numa trama grande, cada pacote corresponde a uma área menor, logo o impacto da perda desse pacote é menor, para além da menor probabilidade das tramas seguintes fazerem uma predição de uma zona “degradada”, logo há menos

propagação de erros. Relativamente à métrica de descarte de trama, são obtidos melhores resultados para o descarte mais à direita (isto é, com o valor de PES mais elevado). A razão de tal comportamento, deve-se à propagação dos erros. Isto é, até chegar à próxima trama I, os erros vão-se propagando, uma vez que essas tramas descartadas são normalmente usadas como predição para tramas posteriores nesse GOP.

De acordo com a filosofia defendida anteriormente, como os resultados são maus, apesar de os testes não terem sido feitos nas condições ideais, é possível concluir que as duas métricas (último pacote IP e trama) não são uma boa solução. Podia-se ter optado por garantir perdas de pacotes mais parecidas, se por exemplo, na métrica de “descarte do último pacote IP nas tramas P maiores” (ver Tabela 3), em vez de ser apenas limitada inferiormente ( $TS > 53$  para uma percentagem de perda de pacotes 0,51%), fosse também limitada superiormente ( $TS < 80$ ). Assim seria possível “manusear” o intervalo nas duas extremidades e, como tal, obter taxas de perda de pacotes mais idênticas. Mas o objectivo foi testar as métricas sem alterar as suas características. Tentou-se assim, para os três testes obter os valores mais próximos de perda de pacotes para cada métrica, de modo a tornar possível a sua comparação. De salientar que a pequena diferença na perda de pacotes, foi praticamente irrelevante para quem atribuiu o QoE.

Na Figura 14, está representado o gráfico correspondente à Tabela 3. Neste gráfico, o eixo das abcissas está representada a perda de pacotes em percentagem obtida nos diferentes testes realizados e no eixo das ordenadas os valores indicados subjectivamente pelas pessoas que observaram o vídeo (QoE).

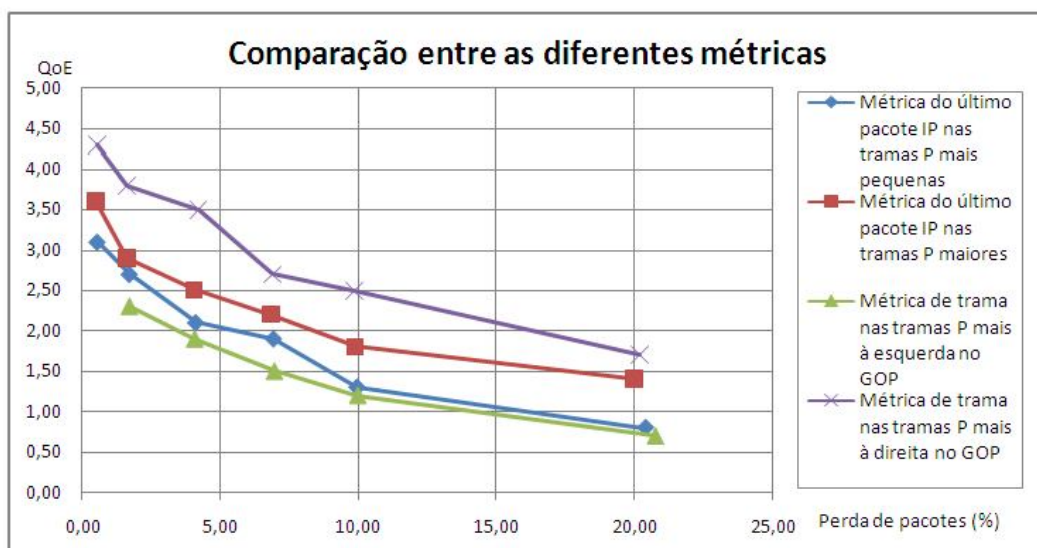


Figura 14 – Gráfico da comparação das métricas analisadas no teste 1.

Analisando a Figura 14, a melhor métrica é claramente a “métrica de trama com descarte das tramas P mais à direita no GOP”. A segunda melhor foi claramente a “métrica do último pacote IP com

descarte das tramas P maiores”. A terceira melhor métrica foi a “métrica do último pacote IP com descarte das tramas P mais pequenas”. A pior métrica foi a “métrica de trama com descarte das tramas P mais à esquerda no GOP”. De salientar que a partir do valor correspondente a 10% de perdas de pacotes, as duas piores métricas são praticamente indistinguíveis uma da outra.

Outro ponto importante é o facto do valor de todas as métricas diminuírem exponencialmente de qualidade com o aumento da perda de pacotes. Só quando as perdas já são elevadas e a qualidade do conteúdo muito má, é que decresce mais lentamente, uma vez que para o cliente a qualidade já é tão má que mais perdas pouco influenciam a sua avaliação. A Figura 14 apresenta um limite aceitável de perdas para a melhor métrica de 5%. Todas as métricas tendem a decair muito rapidamente de qualidade com o aumento da perda de pacotes.

Comparando a melhor métrica com a pior métrica, notou-se que os erros se propagam muito menos, isto é, a melhor métrica recupera mais depressa das perdas. Em termos de flexibilidade na introdução de perdas, a pior métrica é também a menos flexível.

A segunda melhor métrica foi o “descarte de pacotes IP nas tramas maiores”. Notou-se que esta métrica degrada preferencialmente a parte debaixo da imagem. Esta métrica é melhor que a de “descarte de pacotes IP em tramas menores”.

Verificou-se que o VLC no servidor ocupou cerca de 105% o CPU (máximo é 200%, uma vez que é *dual core*) e de 2.2% a memória. Sendo que a lógica a nível da aplicação introduzida no MANE, não tem expressão em termos de recursos consumidos (ao nível do sistema operativo).

Na Tabela 4, estão os resultados referentes ao segundo teste. Nesta tabela é possível observar as variantes das métricas que foram contempladas, a média da perda de pacotes e do QoE obtidos, assim como os seus desvios padrões. Tal como a Tabela 3, tem uma coluna “Valores aplicados no descarte”, que explica como os pacotes foram descartados selectivamente (notação idêntica à da tabela anterior) e apresentam também os desvios padrão, que ilustram a discrepância nas classificações dos observadores.

**Tabela 4 – Resultados obtidos do segundo teste.**

Métrica	Tipo de descarte nas tramas P	Valores aplicados no descarte	Perda de Pacotes (%)	Desvio padrão da perda de pacotes (%)	QoE	Desvio padrão do QoE
Combinação da melhor métrica com a opção “descarte do último pacote IP, nas tramas menores”	Prioridade ao tamanho da trama em vez da sua localização.	PES > 32 e (TS < 17 e TS > 6)	4,13	0,03	2,70	0,45
		PES > 37 e (TS < 21 e TS > 6)	9,85	0,17	1,70	0,27
	Prioridade à localização da trama em vez do seu tamanho.	PES > 94 e TS < 24	4,13	0,12	3,30	0,27
		PES > 83 e (TS < 30 e TS > 6)	9,82	0,24	2,40	0,22
Combinação da melhor métrica com a opção “descarte do último pacote IP, nas tramas maiores”	Prioridade ao tamanho da trama em vez da sua localização.	PES > 36 e TS > 33	4,06	0,05	2,90	0,42
		PES > 24 e TS > 28	9,93	0,13	1,80	0,45
	Prioridade à localização da trama em vez do seu tamanho.	PES > 94 e TS > 22	4,06	0,04	3,30	0,27
		PES > 83 e TS > 18	9,82	0,13	2,20	0,27

Analisando a Tabela 4, verificou-se que as combinações efectuadas das métricas anteriores (ver Tabela 3), foram sempre piores que a melhor métrica anterior (“métrica de trama com descarte mais à direita no GOP”). Entre as combinações, é de notar que aquelas que mais se aproximam da melhor métrica, as que dão prioridade à localização foram melhores do que aquelas que dão prioridade ao tamanho da trama.

Na Tabela 5, encontram-se os resultados do terceiro teste, referentes ao descarte aleatório. Este teste pretende simular um ambiente onde existem perdas (que ocorrem de forma aleatória), para que se possa perceber até que ponto as métricas, e suas variantes com o descarte selectivo, são melhores.

**Tabela 5 – Resultados do terceiro teste.**

Métrica	Perda de Pacotes (%)	Desvio padrão da perda de pacotes (%)	QoE	Desvio padrão do QoE
NIST Net	1,60	0,11	2,00	0,50
	6,88	0,05	1,30	0,45
	9,84	0,05	0,70	0,45

Recorrendo à tabela, é fácil observar que o descarte aleatório é pior do que a melhor métrica obtida no segundo teste. Também se verificou que na melhor solução, as perdas começam a fazer-se notar (efeito de bloco) a partir 1,60% de perda de pacotes.

### 3.2.1.5 Conclusões

Foram realizados testes de adaptação de vídeo ao nível da camada de rede, com diferentes métricas, que tinham como objectivo identificar a viabilidade de realizar a referida adaptação, recorrendo a técnicas de descarte simples através da avaliação da QoE percebida pelos utilizadores. Duma forma geral, não se obtiveram resultados satisfatórios com nenhuma das métricas propostas, nas suas diversas variantes.

As diferentes métricas propostas requerem esforços de realização prática diferente, que é necessário equacionar. Isto porque no ambiente de teste utilizado as características do vídeo são conhecidas, algo que não acontece num ambiente real onde existe um *streaming* contínuo de conteúdos. Assim a métrica do último pacote IP é simples de implementar no ambiente real, na medida em que, no início de cada trama, vem o tamanho desta, sendo apenas necessário ir decrementando um contador até o tamanho de um pacote IP. A métrica de trama, mais à direita no GOP, é mais complexa, uma vez que os GOPs não têm tamanho fixo, nem é possível saber previamente o tamanho de cada um. Contudo, uma alternativa seria, por exemplo, analisar três ou cinco GOPs, obter o tamanho médio desses GOPs, assumir essa média como tamanho médio dos GOPs e, com base nisso, definir um limite para o descarte de tramas. De referir que esta solução poderá não ser muito eficiente, se os GOPs não tiverem tamanhos idênticos. A alternativa seria especificar um tamanho de GOP, na recodificação do vídeo no VLC, o que poderia provocar degradação da qualidade do vídeo (se o intervalo definido fosse muito grande) ou aumento excessivo do ritmo binário (no caso do intervalo ser pequeno de mais).

Um outro ponto desfavorável à qualidade da adaptação apresentada por esta métrica, está relacionado com a própria codificação, que faz com que os pacotes IP não constituam fronteiras entre tramas, ou seja, é frequente num pacote IP ter o final de uma trama e o início de outra. Assim sendo, o mais natural é que não se consiga eliminar uma trama completa (informação que devia ser removida, tornando pouco eficiente a degradação dos conteúdos), sem pôr em causa as tramas posteriores. Para que tal fosse possível seria necessário que a trama começasse no primeiro TS e terminasse no último TS de um pacote IP, algo que raramente acontece. Do meu ponto de vista, tal situação não faz sentido uma vez que o que se ganha em alinhar os pacotes IP com os TS, perde-se em eficiência na codificação.

É importante salientar que o facto de apenas se ter feito testes com vídeo, serviu essencialmente para simplificar. Contudo é bastante natural que o IPTV possa ter outro tipo de conteúdos como áudio e legendas. O descarte nestas situações pode induzir em faltas de sincronismos entre o som, a imagem e o texto, que agravam a qualidade do vídeo que os clientes visualizam. Apesar de ser de simples identificação o tipo de conteúdo, visto que cada TS identifica o conteúdo através de um *Packet Identifier* (PID), no caso destes irem multiplexados, num mesmo pacote IP, podem existir vários tipos de conteúdos, os quais seriam todos degradados, utilizando estas métricas. Uma possível solução para este

potencial problema seria proceder ao envio dos outros conteúdos separadamente (por exemplo, portos diferentes).

Seria interessante usar a melhor métrica de descarte num vídeo com tramas B, em que apenas se descartaria estas tramas em detrimento das tramas P. Contudo os problemas de ineficiência tanto no descarte como na implementação da melhor métrica, são independentes do tipo de tramas utilizadas, mantendo-se com maior ou pior qualidade na percepção visual. Daí que os testes com essa métrica, não tenham sido contemplados.

A conclusão que se retira é que esta solução, tal como ela foi concebida, não serve para degradar o conteúdo de vídeo sem grande impacto na experiência visual de um cliente de IPTV. Talvez com o surgimento do H.264/SVC, seja possível usar a melhor métrica de descarte encontrada, mas para isso é necessário, ultrapassar todas as limitações que esta solução tem.

### **3.2.2 Degradação de conteúdos no servidor**

Como ficou demonstrado na secção anterior, a adaptação da qualidade do vídeo baseada no descarte selectivo de pacotes IP não é uma solução viável. Isto porque o vídeo com pouco mais de 5% de perdas (ver secção 3.2.1.4), deixa de ter uma qualidade aceitável, ou seja, a compressão obtida é relativamente pouca e os danos demasiado grandes, para o cliente os aceitar. Deste modo, a alternativa pensada foi a degradação dos conteúdos ser realizada no servidor. Nesta solução, o *streamer* recodifica o conteúdo multimédia em vários fluxos, cada um dos quais correspondente a um nível de qualidade, cabendo ao MANE escolher a que melhor se adequa ao cliente.

#### **3.2.2.1 Pressupostos**

Foram assumidos quatro pressupostos:

- A infra-estrutura do fornecedor de serviços (excepto a rede de acesso) tem a capacidade de suportar o tráfego de IPTV com os fluxos adicionais.
- Assumiu-se que os endereços IP distinguem canais e que portos distinguem níveis de qualidade. Isto é, o *streamer* envia diferentes níveis de qualidade de um mesmo conteúdo, utilizando o mesmo endereço IP, mas com portos diferentes.

- Tem de existir um elemento na rede, que bloqueie todos os fluxos com níveis de qualidade diferentes daquele que o MANE considerada ideal para o cliente. Isto é, tem de ser garantido, que na rede de acesso, apenas existe um fluxo de vídeo.
- É necessário que o *streamer* tenha a capacidade de codificar os conteúdos e enviá-los em tempo real, isto é, a máquina tem de ter capacidade de processamento para as qualidades existentes.

### 3.2.2.2 *Diferenciação dos níveis de qualidade*

Para distinguir os diferentes níveis de qualidade utilizou-se, como ponto de partida, a diferenciação feita pelo H.264/SVC que se baseia em três pilares: resolução espacial, temporal e/ou qualidade (SNR). Estes três factores produzem um determinado ritmo binário, que identifica a qualidade do vídeo. Contudo, pelo facto de apenas se considerar um tipo de dispositivo (televisão), a resolução espacial não foi contemplada. Isto porque o aumento/diminuição da resolução teria um impacto visual muito grande no cliente e o objectivo é que o cliente tanto quanto possível não note a diferença entre os diferentes níveis de qualidade, algo que não poderia ser alcançado com a mudança de resolução. A alteração da resolução espacial só faria sentido, se a adaptação também abrangesse outros dispositivos.

Para a resolução temporal (FPS) usou-se o parâmetro do VLC que permite definir o número de tramas por segundo. Relativamente à qualidade (SNR), esta corresponde ao passo de quantificação. Como se tem uma aproximação a ritmo constante, o VLC adapta automaticamente os passos de quantificação, de modo a obter o ritmo binário especificado. Por exemplo, quando se diminui o ritmo, o VLC dinamicamente aumenta o passo de quantificação, o que faz com que existam menos níveis para codificar, reduzindo assim o número de bits para codificar cada nível. Caso fosse usado um ritmo variável, era possível definir um passo de quantificação fixo, que consoante o movimento do vídeo assim variava o ritmo binário. Assim tanto o FPS, como o ritmo binário, são previamente definidos no ficheiro de configuração do VLC.

A relação entre o SNR e o FPS deve ser definida de acordo com o tipo de conteúdo, isto é, para os canais com mais movimento, por exemplo, canais de desporto, deve-se diminuir mais o ritmo binário, em detrimento do FPS. Por outro lado, para canais com pouco movimento (tipicamente canais de informação) deve-se reduzir mais o FPS. Isto porque, para vídeos com muito movimento, o olho humano é mais sensível à perda desse mesmo movimento, do que à qualidade da imagem.

Nesta técnica de adaptação foram considerados três níveis de qualidade, com as seguintes características:

- Alta qualidade – Neste nível de qualidade o conteúdo é transmitido com a melhor qualidade possível, ou seja, com o ritmo binário e FPS estipulados para a transmissão dos canais.
- Qualidade intermédia – É reduzido o ritmo binário da alta qualidade em 25% e o FPS pode ser reduzido, se o vídeo com o FPS igual ao da alta qualidade não tiver uma qualidade aceitável.
- Baixa qualidade – É reduzido o ritmo binário da alta qualidade para metade e o número de tramas é também reduzido de acordo com o tipo de conteúdo. Isto porque, o conteúdo com pouco movimento é possível diminuir o FPS sem tanto impacto visual do que com um conteúdo com mais movimento.

Considerou-se que o cliente de IPTV é exigente em termos da qualidade do serviço, uma vez que tipicamente dispõe de televisões com um ecrã grande, onde qualquer problema se faz notar muito mais do que por exemplo, num ecrã de um telemóvel. Assim, definiu-se que o ritmo binário não deveria ser inferior a 50% do ritmo do canal, de modo a que o nível de pior qualidade, seja minimamente aceitável para o cliente. Por exemplo, no caso do cliente estar a receber um vídeo, onde a qualidade é tão má que o vídeo raramente mostra as imagens, a adaptação desse cliente para uma qualidade melhor, mas ainda assim com demasiado efeito de bloco, levará igualmente o cliente a reclamar.

O ritmo binário da qualidade intermédia foi assumido como o meio-termo entre a melhor qualidade e a pior qualidade. Com estes ritmos e uma vez que o ritmo binário de um canal de IPTV é tipicamente cerca de 3 Mb/s, permite ao fornecedor de serviços, dispor de um conjunto de flexibilidades:

- Na qualidade intermédia, este ritmo passaria para 2,25 Mb/s, permitindo reservar maior largura de banda para o serviço de acesso à Internet dos clientes.
- Na pior qualidade, com um ritmo binário por canal de 1,5 Mb/s iria permitir que os clientes, pudessem ter duas STBs em casa a receber ao mesmo tempo, ocupando a largura de banda de um canal normal.

A escolha das percentagens dos ritmos binários foi definida como única para todos os canais, contudo existem canais que exigem maior largura de banda (tipicamente canais com maior movimento) do que outros. Deste modo, os canais devem ser agrupados por características (pouco ou muito movimento), definindo-se as percentagens de redução dos ritmos binários mais de acordo com o tipo do canal. Assim, por exemplo, para um canal informativo onde raramente existe movimento, tem uma maior redução do ritmo binário, relativamente a um canal de desporto.

Para o FPS, convém notar que, dependendo do movimento de vídeo, poderá ser necessário mais ou menos tramas por segundo, daí que esse valor deva também variar com o tipo de conteúdo. Contudo, não convém utilizar valores inferiores a 15 FPS, pois o cliente irá com certeza notar falta de movimento (mesmo para conteúdos com pouco movimento).



### 3.2.2.3 Testes

O ambiente teste usado foi idêntico ao da Figura 13, mas sem *softrouter*, isto é, tanto o *streamer* como o cliente estavam na rede 11. Para as duas técnicas de descarte terem comparação possível, foi usado o mesmo vídeo, que já tinha sido usado nos testes da outra técnica de adaptação.

Os testes realizados consistiram em:

- Encontrar o ritmo binário para a melhor solução. Para tal, foi enviado vídeo com o VLC, sem recodificação, e analisado o ritmo médio obtido. Esse ritmo foi extrapolado para as outras qualidades, isto é, 50% e 75% do ritmo da melhor qualidade, para a pior qualidade e qualidade intermédia respectivamente.
- Analisou-se a qualidade intermédia, com o mesmo FPS que a alta qualidade e averiguou-se a eventual necessidade da sua redução.
- Para a pior qualidade fez-se variar a resolução temporal, com o objectivo de obter a melhor qualidade possível.
- Com vários observadores, analisar e classificar as diferentes qualidades através da métrica de QoE.

### 3.2.2.4 Resultados

Os testes permitiram obter uma métrica, que está ilustrada na Figura 15. Onde o quadrado com o azul mais escuro, significa a melhor qualidade do vídeo, com um ritmo binário de 1200 kb/s e com 25 FPS. A qualidade intermédia tem 900 kb/s e 25 FPS. A pior qualidade, representada por um quadrado com o azul mais claro, tem um ritmo binário de 600 kb/s e 23 FPS.

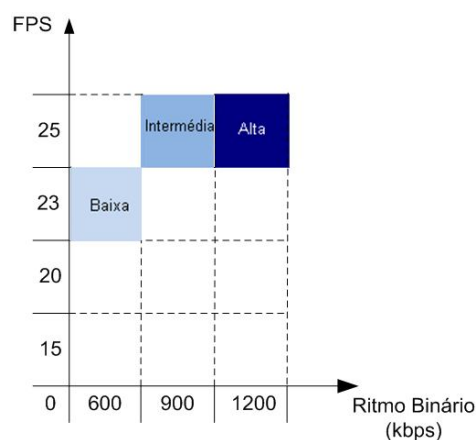


Figura 15 – Qualidades obtidas (devidamente identificadas) para o vídeo considerado.

Nesta adaptação, como o vídeo tinha zonas com bastante movimento, optou-se por uma maior diminuição do ritmo binário em detrimento do FPS. Deste modo, para a qualidade intermédia optou-se por manter a resolução temporal da melhor qualidade, uma vez que a qualidade da imagem foi considerada boa.

Verificou-se que devido às características do vídeo (muito movimento), a redução em demasia do FPS provocava pequenas paragens, que resultavam num grande impacto visual. Daí a pior qualidade ter um FPS de 23, isto é, apenas menos 2 FPS que as outras duas qualidades.

Na Tabela 6, estão descritos os resultados dos testes efectuados ao vídeo com as diferentes qualidades, definidas na Figura 15. A coluna mais à esquerda identifica os três níveis de qualidade, a coluna seguinte identifica o ritmo binário obtido nos testes. A quarta coluna a contar da esquerda, representa os passos de quantificação para as tramas existentes no vídeo (I e P) e a segunda coluna mais à direita identifica a classificação QoE atribuído pelos observadores. Associado a cada uma destas colunas referidas existe uma coluna com desvio padrão, isto porque o vídeo foi visionado de modo independente por cinco pessoas, levando à variação dos valores obtidos.

**Tabela 6 – Tabela com os resultados obtidos da adaptação realizada.**

Nível de qualidade	Ritmo binário teórico médio (kb/s)	Ritmo binário experimental médio (kb/s)	Desvio padrão do ritmo binário	Quantificação média		Desvio padrão da Quantificação		QoE médio	Desvio padrão do QoE
				Tramas I	Tramas P	Tramas I	Tramas P		
Baixa	600	588,62	0,53	25,74	24,89	0,57	0,04	3,40	0,22
Intermédia	900	881,07	0,29	21,30	22,51	0,84	0,04	4,40	0,22
Alta	1200	1172,83	0,54	20,44	20,56	0,92	0,03	5	0

Como é possível observar na Tabela 6, o ritmo binário médio experimental foi perto do valor teórico, daí os valores de QoE (níveis definidos na Tabela 2) poderem ser validados para os níveis da Figura 15. Como seria de esperar o passo de quantificação aumenta com a diminuição da qualidade. Isto porque, diminui o número de níveis disponíveis para codificar, logo maiores serão os erros (devido a arredondamentos) na codificação de cada *pixel*.

Relativamente ao QoE, de salientar que na alta qualidade o desvio padrão foi igual a zero, uma vez que, não houve discrepâncias na atribuição do nível de qualidade, nos resultados indicados pelos observadores.

Na Figura 16, está ilustrado a azul a melhor métrica encontrada na degradação de conteúdos no MANE e a degradação de conteúdos no servidor. Como é visível, a diferença entre as duas técnicas é

enorme, obtendo-se com a degradação no servidor, reduções superiores e sem tanto impacto visual, em comparação com as que se obtiveram na degradação efectuada no MANE.

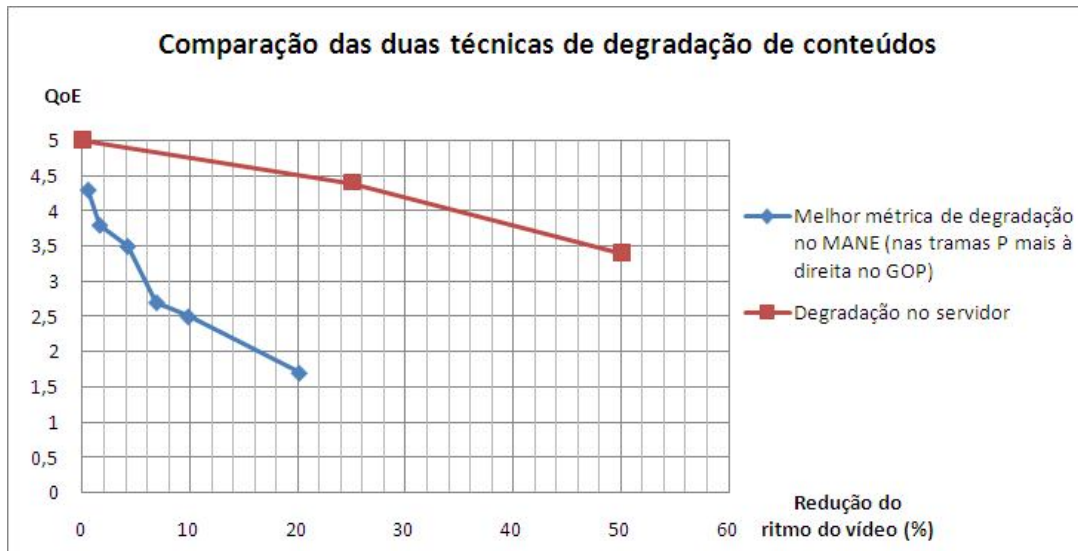


Figura 16 – Comparação entre a melhor métrica de degradação no MANE e a degradação no servidor.

De salientar que para a recodificação e envio deste vídeo foi usado em média 125% da capacidade do processador do *streamer* e 2,2% de memória.

### **3.2.2.5 Conclusões**

Foram realizados testes de adaptação de vídeo com degradação de conteúdos no servidor, que enviava vários fluxos cada um dos quais com um nível de qualidade diferente, sendo entregue ao cliente final apenas o fluxo que lhe oferecia melhor QoE. Nos testes realizados foram seleccionados três níveis de qualidade, cada um dos quais caracterizado por um ritmo de transmissão e um FPS. Duma forma geral, os resultados obtidos são bastante bons.

Nesta solução o número de fluxos, com diferentes níveis de qualidade, que são enviados representa um compromisso entre a maior capacidade de adaptação e a sobrecarga provocada na rede. Por outro lado, para que as diferentes qualidades dos conteúdos cheguem ao seu destino, é necessário um maior número de portos ou IPs abertos nos equipamentos, o que torna a rede mais vulnerável a ataques.

Comparativamente com a solução de descarte de informação ao nível da camada rede, esta solução é bastante melhor, porque os descartes são feitos ao nível do vídeo e não ao nível da camada IP. Deste modo, as perdas constituem pequenas desfocagens nas imagens, muitas vezes imperceptíveis ao olho humano, enquanto que no descarte de pacotes IP as perdas se notam sempre. Para além disso, esta técnica quando comparada com a anterior, permite reduções no ritmo binário muito mais elevadas.

### **3.2.3 Síntese Final**

Neste capítulo foram estudadas duas técnicas de adaptação: o descarte selectivo de informação ao nível da camada de rede e a degradação de conteúdos no servidor. Para a técnica de descarte selectivo ao nível da camada de rede, a degradação foi realizado num MANE de acordo com métricas preestabelecidas. A melhor métrica (métrica de trama, com descarte nas tramas P mais à direita no GOP), isto é, aquela que consegue obter melhor percepção visual para uma determinada perda de pacotes, demonstrou ser insuficiente, uma vez que apenas permite perdas na ordem dos 5%, pois a partir desse valor começa a ter um efeito visual muito negativo.

Para a técnica de degradação de conteúdos no servidor, ficou bem patente a elevada capacidade de redução de ritmo, sem que a qualidade seja demasiado prejudicada, conseguindo obter, por exemplo, reduções de 50% no ritmo binário sem grande impacto visual.

## 4 Adaptação dinâmica

### 4.1 Considerações iniciais

Após a escolha da técnica de adaptação, descrever-se-á o modo como é realizada a adaptação dinâmica dos conteúdos. O objectivo é obter informação do estado da recepção dos clientes, durante o *streaming*, de modo a tomar decisões que visem melhorar a experiência visual do utilizador, para que o cliente receba sempre a melhor qualidade possível, tendo em conta as suas condições actuais de recepção.

Uma das opções para a criação de um canal que permitisse adaptação dinâmica seria a utilização da informação veiculada pelo protocolo RTCP. Num ambiente de IPTV, tipicamente os conteúdos são transmitidos em RTP, o RTCP poderia ser usado como canal de retorno da qualidade dos clientes. Contudo os graves problemas que o RTCP tem com escalabilidade (descritos na secção 2.3.1.2), fazem com que este protocolo deixe de ser uma solução viável.

Assim a alternativa considerada foi a utilização de uma solução designada de *Axess*<sup>39</sup>, para obter informação da capacidade dos clientes. Esta baseia-se na especificação TR-069<sup>40</sup> do DSL *Fórum* intitulada de *CPE WAN Management Protocol (CWMP)*<sup>41</sup> [26].

### 4.2 Solução dinâmica

Utilizando a melhor técnica de adaptação analisada na secção 3.2, juntamente com o *Axess* é possível analisar a capacidade de recepção do dispositivo do cliente, adaptando-o à qualidade que melhor se lhe ajusta num determinado momento.

Inicialmente quando um cliente pede um determinado canal, não é analisada a sua capacidade de recepção, uma vez que este ainda não se encontra a receber os conteúdos multimédia. Portanto é assumido que, quando os clientes fazem o seu primeiro pedido, estes conseguem receber com a melhor qualidade. Contudo, no caso de já existir histórico (existir registo de pedidos no passado) sobre os clientes, então vai ser usada a qualidade do canal, com o ritmo binário mais próximo da média dos valores históricos. No caso de estar à mesma distância entre um nível de melhor e pior qualidade, prevalece o de pior qualidade, por se assumir que, em condições iguais de recepção, é o que tem menor impacto.

---

<sup>39</sup> Explicado detalhadamente na secção 0.

<sup>40</sup> Explicado detalhadamente no Anexo C – CWMP.

<sup>41</sup> Explicado detalhadamente no Anexo C – CWMP.

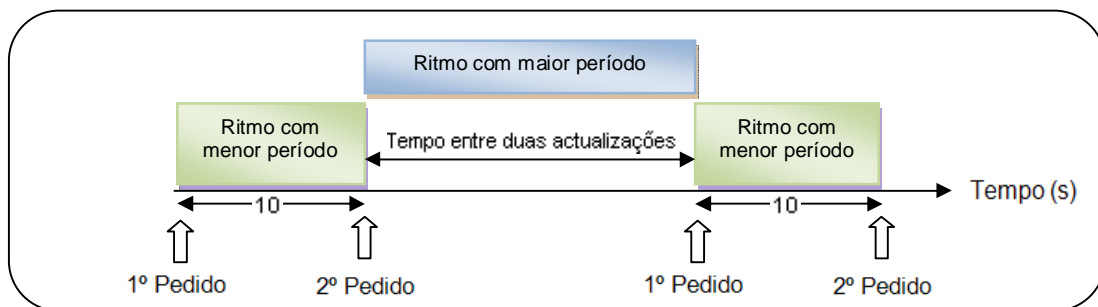
Deste modo, na solução que foi desenhada considera-se que, numa fase inicial, o vídeo é enviado com a melhor qualidade possível, sendo as adaptações às condições reais de uso, realizadas posteriormente em função da informação recebida do dispositivo do cliente. O vídeo é transmitido com um ritmo aproximadamente constante, isto é, o ritmo do vídeo devido a sua natureza, tem de variar para ser garantido a boa qualidade da imagem. Deste modo, analisando o vídeo num intervalo curto de tempo, notar-se-ão oscilações no ritmo, contudo, analisando-o num intervalo maior ver-se-á que em média o ritmo do vídeo é constante. Assim, esta aproximação não inviabiliza a existência de oscilações de ritmo, sentidas principalmente quando se analisa o ritmo em intervalos de tempo muito curtos, e que podem influenciar as adaptações a realizar. Para tal, foram criadas precauções de modo a evitar este potencial problema. Com este propósito são efectuadas duas medidas:

- ✓ A primeira medida designada de “ritmo com menor período” consiste em efectuar dois pedidos ao CPE, separados por um intervalo de tempo predefinido. Em cada pedido é obtido o número de bytes recebidos do exterior, porta *Wide Area Network* (WAN), assim como o *uptime*, de modo a calcular o ritmo binário. Estes dois pedidos poderiam ser substituídos por um único, no caso do CPE fornecer um parâmetro de ritmo binário instantâneo. Contudo, obter simplesmente esse ritmo dos CPEs tinha um risco muito maior de ser influenciado pelas oscilações naturais do vídeo, de que obter um valor mediano (através dos dois pedidos efectuados ao CPE). É importante salientar que o intervalo de tempo predefinido é um *tradeoff*, entre o tempo que demora a actualizar o ritmo binário e a precisão dessa mesma adaptação. Pois para intervalos curtos de tempo a probabilidade desse valor ser influenciado por características do vídeo (como por exemplo, por períodos de menos movimento do vídeo) é muito maior. Nos testes que foram realizados, foi utilizado o valor de dez segundos, porque se considerou um período razoável de tempo para obter uma boa média do ritmo binário, contudo outros valores poderiam ter sido escolhidos.
- ✓ A segunda medida designada de “ritmo com maior período”, consiste em efectuar um pedido ao CPE, para obter o número de bytes recebidos do exterior (na porta WAN) e o *uptime*. Assim calcula-se o ritmo binário médio, fazendo a diferença entre os valores obtidos e os valores do segundo pedido da actualização anterior (obtidos para o cálculo do “ritmo com menor período” anterior, previamente guardados). Esta medida serve para evitar que o “ritmo com menor período” dê uma ideia errada da capacidade de recepção do cliente, ou seja, serve para evitar os problemas dos picos descritos anteriormente. O “ritmo de maior período”, só não será calculado quando:
  - É a primeira a actualização, uma vez que não tem histórico.
  - Entre duas actualizações, o cliente mudou de canal ou desligou a televisão. Uma vez que no caso de mudar de canal, este pode ser um canal codificado com menos ritmo binário (um canal com outro perfil, por exemplo, passar de um canal de desporto para

um canal de notícias) e induzir em erro. Desligando a televisão o ritmo binário do intervalo vai diminuir drasticamente.

- É feito *reset* ao CPE entre duas actualizações. Uma vez que o valor de *uptime* é colocado a zero.

Na Figura 17, estão descritas estas duas medidas, a verde o “ritmo com menor período”, onde são feitos os dois pedidos, com o intervalo de dez segundos. A azul está o “ritmo com maior período”, que é a medida do ritmo binário entre o segundo pedido do primeiro “ritmo com menor período” e o primeiro pedido do segundo “ritmo com menor período”, correspondendo assim ao ritmo binário entre duas actualizações de capacidade do cliente. Estas actualizações foram predefinidas com um intervalo de cinco minutos, por se considerar um valor razoável, uma vez que quanto menor o intervalo, maior será o número de actualizações e conseqüentemente, maior sobrecarga no servidor. Para além disso, o intervalo não pode ser muito grande porque senão raramente é calculado, uma vez que este intervalo para ser calculado necessita que o cliente não mude de canal, nem desligue a televisão, situações que podem ocorrer com alguma frequência.



**Figura 17 – Ritmos de maior e menor período.**

Na adaptação, tanto o aumento como a redução da qualidade, são feitos de modo incremental. A decisão na adaptação consiste em comparar as medidas efectuadas (“ritmo com maior período” e “ritmo com menor período”), com um valor do ritmo binário de referência, o qual é designado de “ritmo de fronteira”, que é definido com uma margem de segurança (foi estipulada de 5%), abaixo do ritmo binário do canal. Esta margem de segurança serve essencialmente para que não sejam tomadas decisões precipitadas, motivadas por uma estimacão incorrecta da situação em curso. Uma vez que como já referido, devido às oscilações no ritmo do vídeo, as medidas obtidas raramente coincidem com o valor teórico estipulado.

A decisão de adaptar é tomada de acordo com o algoritmo descrito na Figura 18. É importante salientar que neste algoritmo, o “ritmo de maior período” só é utilizado quando o “ritmo de menor período” sugere a alteração da qualidade. Contudo, o “ritmo de maior período” para servir de factor de decisão na mudança da qualidade, tem de poder ser calculado (existir uma actualização anterior) e estar em condições de ser credível, isto é, entre a actualização anterior e a actual, o cliente não efectuou

nenhuma mudança de canal, nem outra acção que desvirtue o ritmo obtido, como por exemplo, desligar a televisão. Caso estas condições não se verifiquem, então o “ritmo de menor período” é soberano na decisão.

Quando o “ritmo de menor período” apontar para a descida de qualidade e o “ritmo de maior período” indicar boa qualidade, considera-se que é um período de menos movimento no vídeo, em detrimento de um problema real, uma vez que é muito improvável que o problema surja durante o cálculo do “ritmo de menor período”.

No caso do “ritmo com menor período” apontar para a melhoria de qualidade e o “ritmo de maior período” indicar que a qualidade não esteve boa, não se altera para uma qualidade melhor, uma vez que se considera que esta pode ser só uma fase temporariamente boa ou com mais movimento, pois seria muito improvável a qualidade melhorar no instante em que é obtido o “ritmo de menor período”.

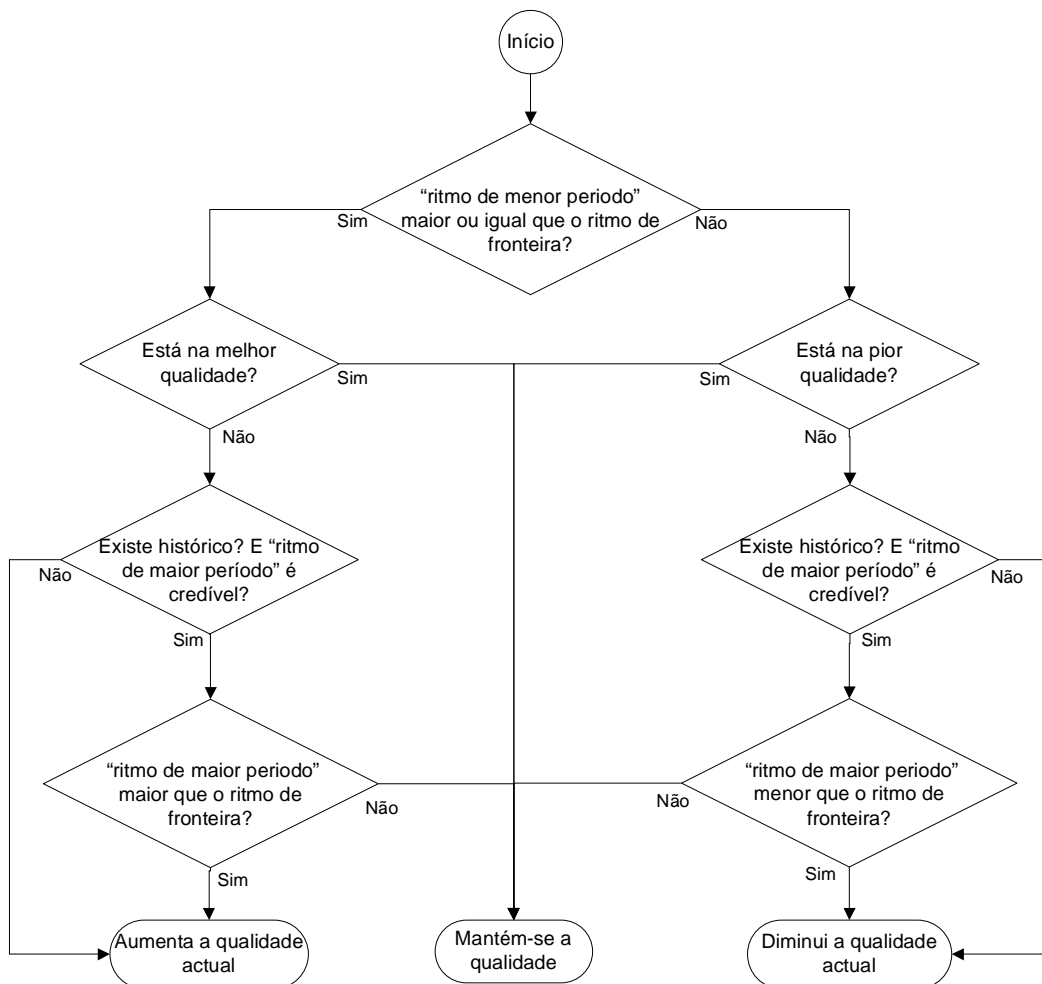


Figura 18 – Fluxograma com o algoritmo de adaptação.



Este algoritmo tem o objectivo de colocar sempre que possível o cliente com a melhor qualidade. Contudo, poder-se-á estar aqui na presença de um efeito de *ping pong* entre a qualidade que o cliente recebe bem (mas que não é a melhor qualidade do canal) e a qualidade do canal acima do ritmo binário recebido pelo cliente, que este não consegue visualizar na perfeição.

Na Figura 19, está exemplificado o efeito de *ping pong*. Nesta figura, existem duas qualidades de um vídeo (a tracejado) enviadas pelo servidor, onde a melhor qualidade com 300 kb/s e a segunda qualidade (75% da melhor qualidade, como definido na secção 3.2.2) com 225 kb/s. A vermelho está indicado o ritmo recebido pelo cliente. Assim, no caso de não existir histórico, o cliente vai utilizar o canal de maior ritmo, contudo o cliente apenas consegue receber 256 kb/s. Posteriormente quando forem feitas as actualizações do ritmo (identificadas na figura com barras a cinzento, cada uma das quais, tem a altura correspondente ao ritmo recebido pelo cliente), as duas medidas (“ritmo com menor período” e “ritmo com maior período”), irão indicar um ritmo binário de cerca de 256 kb/s, então será diminuída a qualidade. Posteriormente com nova actualização, as medidas (“ritmo com menor período” e “ritmo com maior período”) irão indicar um ritmo de cerca de 225 kb/s, a opção passa assim por aumentar a qualidade. Caso este procedimento se prolongue no tempo, isto é, a qualidade máxima que o cliente consegue receber (256 kb/s), está entre as duas qualidades fornecidas pelo servidor, não faz sentido andar em constantes adaptações.

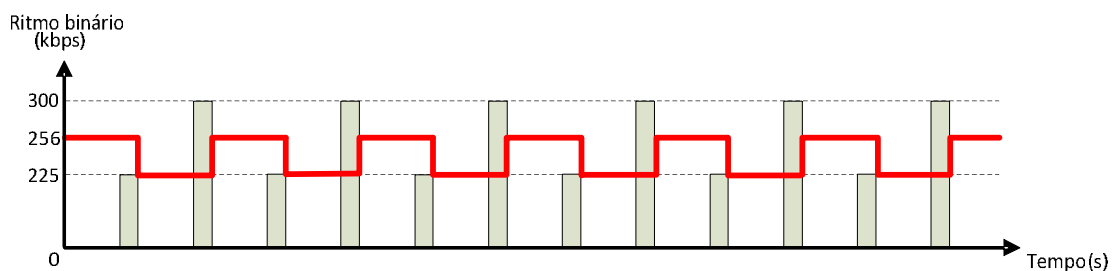
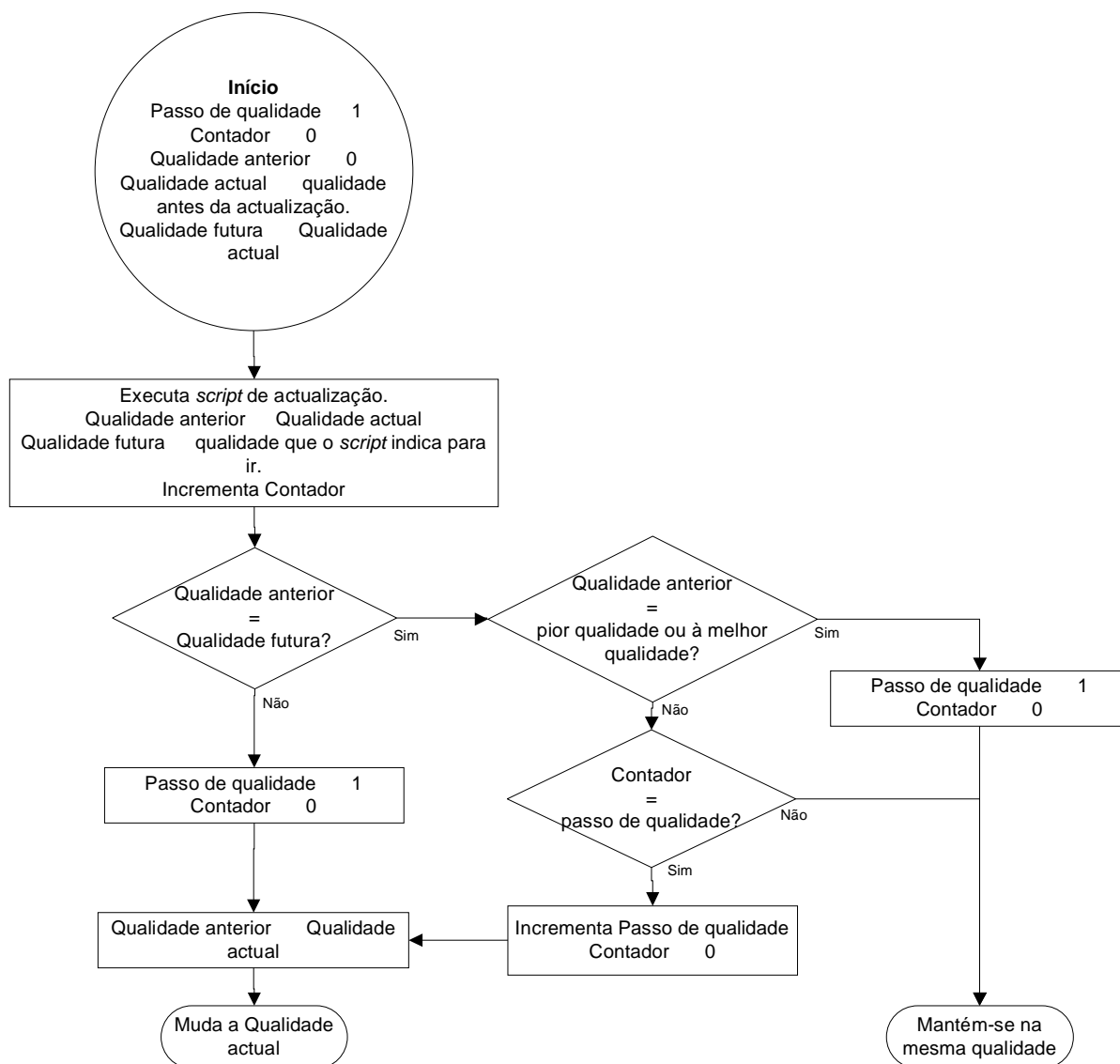


Figura 19 – Exemplo do efeito de *ping pong*.

Deste modo, surge aqui um *tradeoff* entre rapidez na tentativa de adaptação e a fiabilidade dessa adaptação (ser ou não a melhor adaptação). Para evitar este efeito de *ping pong*, foi utilizado um contador que é incrementado cada vez que ocorre uma actualização. Este contador foi auxiliado por um passo de qualidade, que representa o número de actualizações menos uma, necessárias para voltar a mudar de qualidade, enquanto se verificar que a qualidade anterior é igual à qualidade futura. Com o efeito de *ping pong*, só é mudada a qualidade quando o valor do contador é igual ao do passo de qualidade. Se ocorrer mudança de qualidade, mas a qualidade futura continuar a ser igual à anterior, o passo de qualidade é incrementado, senão volta ao valor um. Assim com estes dois elementos, no caso de o cliente estar num efeito de *ping pong*, a sua mudança de qualidade vai ser feita cada vez mais tarde, uma vez que a janela cresce de modo incremental.

Na Figura 20, está ilustrada a solução que permite aliviar o problema do efeito de *ping pong*. No início não existe qualidade anterior, o passo de qualidade começa com o valor um e o contador com zero. Sempre que há uma actualização da capacidade de recepção do cliente, é incrementado o contador e posteriormente verificado se a qualidade futura é igual à qualidade anterior. Se for igual, é verificado se a qualidade anterior é a pior qualidade ou melhor qualidade. Isto para que por exemplo, no caso de o cliente estar a receber a melhor qualidade e for sempre indicado nos ritmos calculados que a qualidade futura é a melhor qualidade, ir-se-á ter sempre qualidade anterior igual à qualidade futura. No entanto, não se trata de um efeito de *ping pong*, daí manter-se na mesma qualidade, reiniciando o contador e passo de qualidade. Se este não for o caso, então estamos na presença do efeito de *ping pong*. Como tal, verifica-se se o contador é igual ao passo de qualidade, caso seja verdade, vai para a qualidade futura, incrementando o passo de qualidade e colocando o contador com o valor zero. Também é actualizada a qualidade actual como qualidade anterior. Caso o contador não seja igual ao passo de qualidade, então não se muda de qualidade.

Se numa actualização a qualidade anterior for diferente da qualidade futura, então coloca o passo de qualidade com o valor um, e o contador com o valor zero. Para além disso, põe a qualidade actual como qualidade anterior e muda de qualidade.



**Figura 20 – Fluxograma com o método utilizado para reduzir o *ping pong*.**

Na Figura 21, está exemplificado o efeito da utilização do mecanismo que reduz o efeito de *ping pong* (descrito no fluxograma da Figura 20). Esta figura é idêntica à Figura 19, sendo adicionado o passo de qualidade (representado na figura por PQ), que começa com o valor um e o contador (representado na figura por C), que inicia com o valor zero. Quando ocorre a primeira actualização, como a qualidade anterior (por ser a primeira actualização) não existe, volta novamente a por o contador a zero, mudando para a qualidade de 225 kb/s. Na segunda actualização, a qualidade anterior é igual à qualidade futura, o contador é incrementado, ficando com o mesmo valor do passo de qualidade. Assim nesta actualização da qualidade, vai-se novamente mudar de qualidade, mas o passo de qualidade é incrementado e o

contador colocado a zero. Na terceira actualização, apesar de indicar para a mudança de qualidade, como o contador é incrementado (fica com o valor um), mas o passo de qualidade está com o valor dois, então não vai mudar de qualidade. E assim sucessivamente até o cliente, ter uma qualidade futura diferente da anterior.

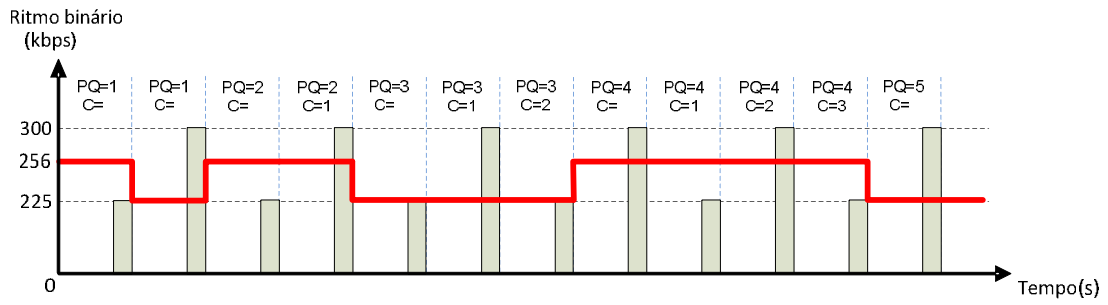


Figura 21 – Utilização do mecanismo de redução do efeito de ping pong.

## 5 Solução proposta para adaptação do vídeo na rede de acesso

Neste capítulo descrever-se-á uma solução de adaptação baseada na técnica de adaptação, “degradação de conteúdos no servidor” (descrita em 3.2.2), usando uma adaptação dinâmica baseada em CWMP (descrita no capítulo 4). Esta solução será antes de mais, um modelo para prova de conceito, aferindo a possibilidade que a melhor técnica de adaptação (obtida do capítulo 3) tem de ser aplicada num sistema de IPTV e não um protótipo para implementação num ambiente real.

### 5.1 Descrição geral

A solução que foi concebida e implementada consiste num servidor que envia cada canal com diferentes níveis de qualidade. Entre o servidor e o cliente, existe um elemento responsável por cortar todos os fluxos que correspondam a níveis de qualidade não adequados ao cliente.

Na Figura 22, está ilustrada a solução apenas com um canal, onde os diferentes tons de cinzento distinguem os diferentes fluxos, correspondente aos diferentes níveis de qualidade, enviados pelo *streamer*. A azul está representada a melhor adaptação ao cliente.

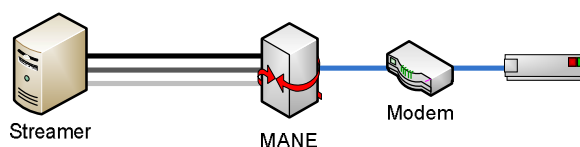


Figura 22 – Ambiente geral da solução de reencaminhamento de portas.

O *streamer* envia os conteúdos para um endereço IP *multicast*, com determinado porto; os níveis de qualidade do mesmo conteúdo são enviadas para o mesmo endereço IP, mas para portos diferentes. O cliente quando se associa a um grupo *multicast*, vai-se associar a todas os portos desse grupo. O MANE que se encontra entre o *streamer* e o *Modem*, inicialmente bloqueia todos os fluxos de vídeo, excepto o que é recebido pelo porto que transporta o fluxo com a melhor qualidade. Posteriormente, é analisado o débito a que o *Modem* recebe os conteúdos e consoante o resultado dessa análise pode ser reencaminhado outro conteúdo em detrimento do actual.

É importante salientar que esta solução é totalmente independente do modo como é feita a deterioração na recodificação feito pelo *streamer*. Contudo basear-se-á no esquema de codificação proposto (descrito na Figura 15).

### 5.1.1 Pressupostos

Os pressupostos fundamentais para a construção desta solução foram:

- ✓ O MANE tem sempre a informação (endereços IPs e portos) de todos os canais disponíveis, assim como a informação sobre os seus níveis de qualidade. Esta informação é vital para bloquear os conteúdos vindos do *streamer* e para fazer o reencaminhamento dos conteúdos para o cliente.
- ✓ O MANE é avisado cada vez que um cliente faz um pedido *multicast*.
- ✓ Os conteúdos de multimédia enviados seguem a um ritmo aproximadamente constante, sendo os níveis de qualidade essencialmente diferenciados pelo seu ritmo. Caso se tivesse ritmo variável a análise poderia induzir em erro, isto é, poder-se-ia baixar a qualidade sem justificação, apenas porque, devido à própria natureza do vídeo, se estava a receber uma parte do vídeo com menos movimento.

## 5.2 Arquitectura da solução

### 5.2.1 Descrição geral

Na Figura 23, está ilustrada a arquitectura da solução com os seus principais componentes. O *streamer* de acordo com a secção 3.2.2 envia três fluxos, dos quais apenas um é escolhido pelo MANE para ser enviado para o cliente. Esta filtragem é feita através do *iptables*<sup>42</sup>, que permite analisar o tráfego que passa pelo MANE e tomar uma decisão sobre o seu destino. Este mecanismo funciona com base em conjuntos de regras, que permitem essencialmente filtrar e reencaminhar o tráfego para o cliente. Assim cada regra possui um filtro com: porto de destino, IP de origem, IP de destino e protocolo UDP. Este filtro permite determinar se a regra é ou não aplicada ao pacote IP e a acção a tomar caso o filtro se aplique (descartar, aceitar ou reencaminhar).

A verde encontra-se representado o *Auto Configuration Server (ACS)*, que permite ao MANE obter informação dos CPEs<sup>43</sup>. As setas a preto indicam os fluxos enviados do *streamer* para o cliente, dos quais apenas um deles chega ao cliente. As setas a vermelho e a roxo constituem o canal de adaptação dinâmica, a diferença está no facto da seta a vermelho, comunicar apenas com o ACS para obter informação presente na sua base de dados. Enquanto que as setas a roxo comunicam com o ACS, para

<sup>42</sup> Descrito detalhadamente no Anexo A – Tecnologias de suporte.

<sup>43</sup> É importante referir que por limitação não se usou STB, daí ir-se buscar o IP ao CPE.

este efectuar um determinado pedido ao CPE. As setas a cinzento representam as interacções entre os diversos blocos funcionais do MANE.

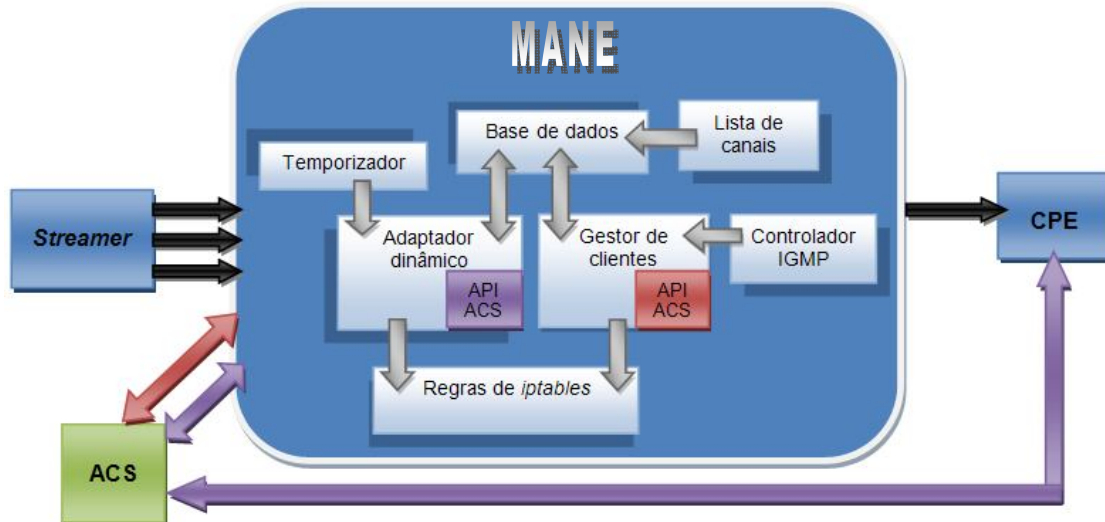


Figura 23 – Arquitectura da solução.

## 5.2.2 Modo de funcionamento

A arquitectura apresentada funciona do seguinte modo: a lista de canais é colocada inicialmente na base de dados. Quando o cliente faz um pedido para um novo canal, a informação referente a esse pedido é obtida no controlador IGMP, que vai enviar o pedido para o gestor de clientes. Este módulo com o IP do CPE (obtido do pedido *multicast*), vai pedir ao ACS o identificador do CPE, *cpeid*. Esta informação é importante porque ao contrário do *cpeid*, o IP do CPE pode mudar ao longo do tempo, assim em futuras actualizações o pedido efectuado ao ACS, é feito a um *cpeid* específico.

Com a informação do pedido do cliente (obtida no controlador IGMP), actualiza a base de dados, associando esse cliente ao canal pretendido e actualiza as regras de *iptables*, de modo a que o cliente receba inicialmente, apenas os conteúdos com a melhor qualidade. O adaptador dinâmico é despoletado periodicamente por um temporizador, onde a sua função é actualizar na base de dados local, o IP e o ritmo binário do CPE, podendo também actualizar as regras de *iptables* caso seja necessário (mudança de qualidade).

Assim quando um cliente faz um pedido, por exemplo, para ver o canal X, o controlador IGMP é notificado desse pedido e informa o gestor de clientes. Este vai actualizar a base de dados com a nova associação do cliente ao canal X e actualizar as regras de *iptables*, para a qualidade que o cliente deve

receber. Periodicamente o adaptador dinâmico irá analisar a qualidade de recepção do cliente com o canal X.

## 5.2.3 Descrição detalhada dos blocos funcionais

### 5.2.3.1 Controlador IGMP

O controlador IGMP é o módulo responsável por interpretar os pedidos dos clientes (mensagens de controlo do grupo *multicast*). Recorrendo ao uso de um *web service*, obtém-se o IP do cliente e do canal a que este se pretende associar, assim como, a acção que o cliente pretende efectuar (associação ou desassociação a um determinado canal). Este controlador recebe do elemento na rede responsável por tratar os pedidos *multicast* dos clientes numa rede de IPTV (por exemplo, o DSLAM), o tipo de mensagem (*join* ou *leave*), os endereços IP do cliente e do canal que o cliente se pretende associar. A informação é passada para o módulo “Gestor de clientes”.

De salientar que este módulo também poderia ser um simples “*sniffer*”, em que recebia os pedidos dos clientes e os analisava, por exemplo recorrendo ao *target QUEUE* do *iptables*<sup>44</sup>. Contudo, seria necessário, que o MANE estivesse entre o cliente e o elemento responsável por tratar os pedidos, de modo a receber o pedido e a identificar o cliente.

### 5.2.3.2 Gestor de clientes

O Gestor de Clientes é o módulo que recebe informação do controlador, actualiza a base de dados e as tabelas de *iptables*. O módulo usa uma *Application Programming Interface* (API), que se baseia em chamadas *Remote Procedure Call* (RPC), para comunicar com o ACS.

A seta a vermelho é a comunicação entre o MANE e o ACS; esta comunicação é usada pelo gestor de utilizadores para obter informação do CPE, que já se encontra no ACS. Isto é, o CPE quando se liga ou quando muda o IP informa o ACS, deste modo o ACS tem sempre a informação do IP actualizada. Quando um cliente envia uma pedido de associação a um canal (*join*), o cliente é identificado nesse pedido pelo endereço de origem, logo é possível no ACS através desse endereço IP, pedir o identificador do CPE, *cpeid*, que é único e nunca muda. O *cpeid* servirá de identificação do cliente, para posteriores comunicações com o ACS.

---

<sup>44</sup> Para informação mais detalhada ver 0.



### 5.2.3.3 *Temporizador*

O Temporizador é responsável por despoletar periodicamente o “adaptador dinâmico”. Para tal, usou-se o ficheiro de configuração *crontab* do Linux, onde se agenda os comandos a serem executados pelo cron<sup>45</sup>. Este ficheiro de configuração associa o módulo “Adaptador dinâmico” ao tempo definido entre actualizações. Tempo esse, que é utilizado no cálculo do “ritmo de maior período”, neste caso cinco minutos.

### 5.2.3.4 *Adaptador Dinâmico*

O adaptador dinâmico é accionado pelo temporizador. Este módulo tem como função actualizar periodicamente a base de dados e as regras de *iptables*. Para tal recorre a uma API para comunicar com o CPE via ACS. As actualizações realizadas resumem-se ao endereço IP do CPE do cliente e posteriormente o tráfego consumido por este. De salientar que no caso do endereço IP do CPE mudar, é necessário actualizar a base de dados e no caso de se utilizar as regras de *iptables*, é necessário também actualizá-las com o novo endereço IP.

Com a análise do tráfego pode ser necessário actualizar as regras *iptables* de modo a alterar a qualidade dos conteúdos enviados para o cliente. Essa decisão é tomada de acordo com o descrito na secção 4.2.

A seta a roxo é a comunicação entre o MANE e o CPE, usada pelo módulo adaptador dinâmico para obter o ritmo binário do CPE. Na comunicação é usado como identificador do CPE o *cpeid*. O ritmo binário é obtido, pedindo duas vezes, com um intervalo de dez segundos (intervalo definido em 4.2), ao CPE dois parâmetros: o *uptime* e o *EthernetBytesReceived* (na porta WAN). Depois de converter o *EthernetBytesReceived* para *bits*, é medido o ritmo médio, fazendo a diferença entre os dois valores obtidos e dividindo pela diferença dos dois valores de *uptime*.

### 5.2.3.5 *Iptables*

O módulo de *iptables* é composto por um conjunto de regras, com o intuito de bloquear e reencaminhar tráfego. Estas regras são responsáveis pela qualidade dos conteúdos visionados pelo cliente. O MANE parte do princípio de bloquear qualquer reencaminhamento de tráfego, colocando assim *default policy drop* na *chain forward* da tabela *filter*. Posteriormente, aquando do pedido do cliente, são marcados todos os níveis de qualidade do canal pedido na tabela *mangle*, e é feito o *Network Address*

---

<sup>45</sup> O *cron* executa comandos automaticamente em *background*, de acordo com o agendado.

*Translator* (NAT) para cada um destes níveis de qualidades do canal, excepto o que corresponde à melhor qualidade (o cliente já está à espera de receber na porta que o *streamer* envia a melhor qualidade). O reencaminhamento do nível de qualidade considerado ideal para o cliente em cada actualização, é feito através da tabela *filter* na *chain forward*, onde se coloca a marca (definida na tabela *mangle*) correspondente à qualidade pretendida. Quando é recebida uma mensagem IGMP do tipo *leave*, são removidas todas as regras das tabelas *filter*, *mangle* e *nat* correspondentes a esse cliente.

### 5.2.3.6 **Lista de canais**

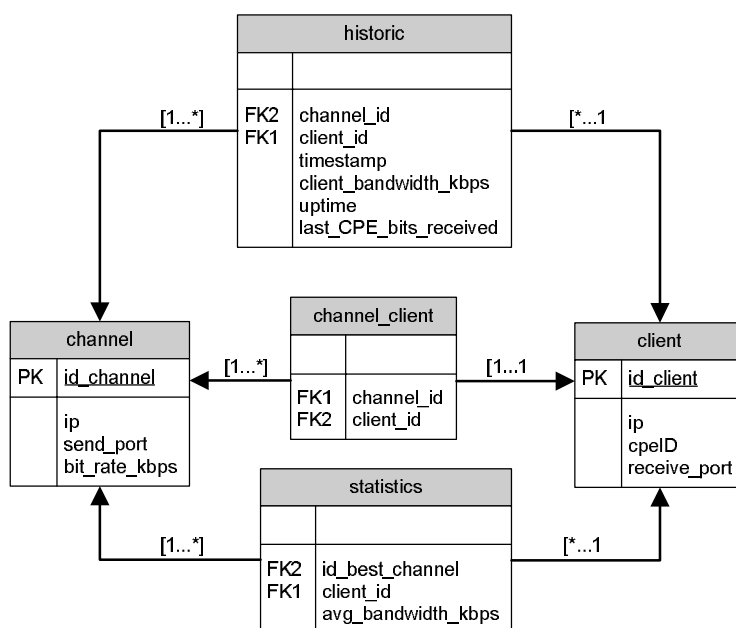
A Lista de canais é um *web service*, que recebe as listas de canais existentes (incluindo informação sobre os níveis de qualidade), permitindo deste modo, saber qual o porto que o cliente fica à escuta, assim como o ritmo binário de cada um destes níveis de qualidade. A informação recebida é actualizada na base de dados local.

### 5.2.3.7 **Base de dados**

A base de dados está implementada em *mysql*. Na Figura 24, estão representadas as quatro tabelas existentes, na base de dados. A tabela "*client*" representa as propriedades de um cliente. É importante salientar, que o campo "*receive port*" é a porta onde o cliente recebe os conteúdos. Corresponde à porta de melhor qualidade, ou seja, na tabela de NAT do *iptables* tem de estar como porta de destino a porta de melhor qualidade desse canal, essa porta portanto nunca dá indicação do tipo de qualidade que o cliente está a receber.

A tabela "*channel*" representa as propriedades de um canal, sendo um canal identificado univocamente pela combinação dos campos "*send\_port*" e "*ip*". A tabela "*channel\_client*" é uma tabela que serve para relacionar as outras duas tabelas, permitindo saber em cada instante, a que canal (IP e porta) está associado cada cliente. Nesta tabela apenas estão presentes os clientes activos. É actualização desta tabela é feita pelo "Gestor de utilizadores" no caso de um cliente ter feito um pedido, ou pelo "Adaptador dinâmico" caso seja necessário alterar a qualidade do canal.

A tabela "*historic*" é utilizada para o cálculo do "ritmo de maior período", através dos campos "*uptime*" e "*last\_CPE\_bits\_received*". O campo "*client\_bandwidth\_kb/s*", é utilizado pela tabela "*statistics*", para calcular a média dos ritmos obtidos do cliente. Esta tabela é utilizada, no caso do cliente se desassociar de um canal e posteriormente voltar-se a associar ao mesmo canal. Isto irá permitir numa fase inicial a adaptação do cliente, utilizando o nível com o ritmo binário imediatamente abaixo do valor médio dos ritmos binários obtidos anteriormente ("*avg\_bandwidth\_kb/s*"), tal como definido em 4.2.



**Figura 24 – Modelo relacional da base de dados.**

Na Tabela 7, estão descritos pormenorizadamente os campos da base de dados local, que contém informação importante dos clientes e dos canais existentes. É importante referir que na tabela *historic* foi colocado um limite de 10 resultados, utilizados para a média. Sendo os resultados mais antigos retirados da tabela e substituídos pelos mais recentes (análise feita recorrendo ao campo *timestamp*). A razão para esta opção prende-se com o facto de apenas interessar o passado recente, ou seja, no caso de não haver limite, existe uma probabilidade muito maior, da média obtida ter valores que em nada correspondem à situação actual e que, ao serem utilizados, poderiam prejudicar a qualidade inicial recebida pelo cliente. Ou seja, muitos resultados antigos poderiam distorcer a realidade actual do cliente.

Tabela 7 – Campos da base de dados local do MANE.

Tabela	Campo	Descrição
<b>Client</b>	<b>id_client</b>	Identificador, que começa em 1 e é incrementado, por cada novo cliente inserido.
	<b>ip</b>	Identifica o cliente, mas este só é usado como identificador quando o cliente realiza uma acção, isto é, só é usado como identificador pelo módulo “Gestor de utilizadores”.
	<b>cpeID</b>	O módulo “Adaptador dinâmico” usa sempre como identificador o <i>cpeid</i> , pois este nunca muda, enquanto que o IP do CPE pode ter mudado desde o pedido enviado pelo cliente.
	<b>receive_port</b>	É a porta onde o cliente recebe os conteúdos, ou seja, na tabela NAT do <i>iptables</i> tem de estar como porta de destino o “ <i>receive_port</i> ”. Como o cliente não pode andar a mudar de porta cada vez que se lhe altera a qualidade, assumiu-se que esta corresponde à porta enviada (“ <i>send_port</i> ”) com a melhor qualidade do canal.
<b>Channel</b>	<b>id_channel</b>	Identificador, que começa em 1 e é incrementado, por cada novo canal inserido
	<b>ip</b>	O IP do canal não é único na base de dados.
	<b>send_port</b>	Especifica a porta de um canal com determinada qualidade. A porta não é um identificador único.
	<b>bit_rate_kb/s</b>	Especifica em kb/s o ritmo constante a que são enviados os conteúdos. Esta informação é importante na decisão da qualidade por parte do “adaptador dinâmico”
<b>Historic</b>	<b>timestamp</b>	Indica o <i>timestamp</i> do último pedido ao CPE. Isto é importante para saber os ritmos binários mais recentes.
	<b>client_bandwidth_kb/s</b>	É actualizado pelo “Adaptador dinâmico” com o ritmo binário que o CPE recebe naquele momento.
	<b>uptime</b>	É utilizado para calcular a largura de banda entre duas actualizações da capacidade de recepção do cliente.
	<b>last_CPE_bits_received</b>	É utilizado para calcular a largura de banda entre duas actualizações da capacidade de recepção do cliente.
<b>Statistics</b>	<b>avg_bandwidth_kb/s</b>	Indica a média dos ritmos presentes na tabela <i>historic</i> . Este é importante para quando o cliente voltar a pedir um novo canal, receber o canal que tem o ritmo mais perto do ritmo médio. No caso de ser um valor intermédio entre dois canais, deve-se utilizar a pior qualidade.
	<b>id_best_channel</b>	Identificador do canal com a melhor qualidade. É importante, de modo a que a informação proveniente essencialmente do valor médio obtido dos resultados históricos, possa servir no futuro de factor de adaptação. Isto, aquando de uma nova associação ao mesmo canal.

### 5.3 Ambiente de teste

O ambiente de teste está representado na Figura 25, sendo constituído por:

- Um servidor como *streamer*: com 2 GB de RAM e um CPU de 2,8 GHz.
- Um servidor como ACS: com 512 MB de RAM e um CPU de 1,5 GHz. O ACS utiliza TR-069 para comunicar com os CPEs.
- Um *switch* de 16 portas (SURECOM EP-816DX-FW).
- Dois *modems* designados por *modem1* (um Thomson 585 v6) e *modem2* (um Thomson 546 v6).
- Um cliente com 1 GB de RAM e um CPU de 1,73 GHz.
- Um MANE com 1 GB de RAM e um CPU de 1,73 GHz.

Conforme se pode verificar na imagem, os três graus de cinzento representam um conjunto de três diferentes níveis de qualidade, que são enviados através de um cabo de rede de 100 Mb/s. Tanto o ACS, como o MANE e o servidor pertencem à LAN do *modem1* (constitui o ponto de comunicação com o exterior). Para se garantir que os conteúdos enviados pelo servidor passem sempre pelo MANE, foi isolado o servidor, criando uma segunda LAN entre o servidor e o MANE. O intuito é que o servidor não tenha ligação directa ao *modem1*, evitando a optimização de rotas, realizadas pelas tabelas *Address Resolution Protocol* (ARP), tendo sido colocado como *default gateway* do servidor o MANE (que tem na interface física a LAN do *modem1* e numa interface virtual a LAN com o servidor). Contudo, apenas um dos três fluxos é enviado para o cliente. Esse conteúdo passa pela Internet (ambiente não controlado) entre o *modem1* e o *modem2*. De salientar que estes dois equipamentos estão ligados a dois operadores distintos.

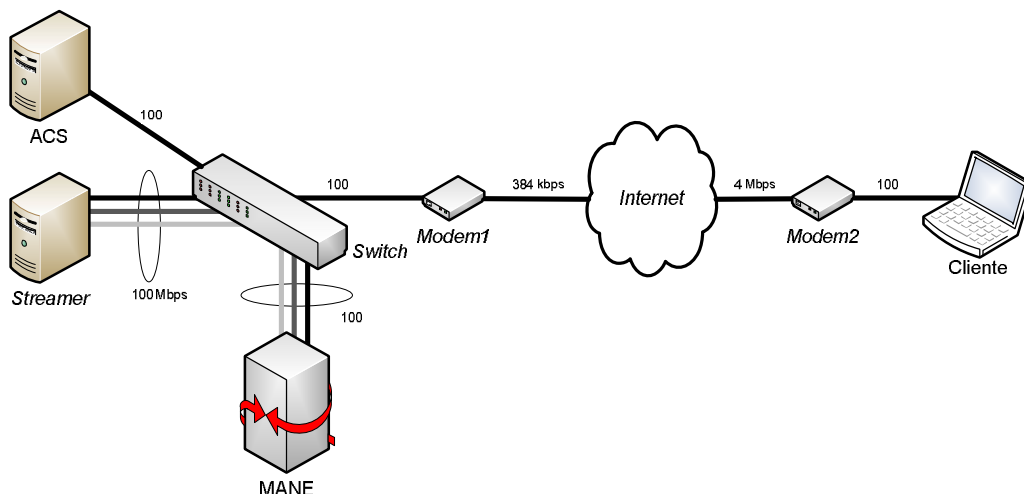


Figura 25 – Ambiente de teste da solução.

Por limitação do CPE (neste caso *modem2*), foi necessário enviar a informação pela Internet, uma vez que o CPE tinha de receber os conteúdos na porta WAN, de modo a posteriormente obter os valores de *uptime* e número de bytes recebidos. Isto porque nos portos de LAN do CPE, o parâmetro número de bytes recebidos, reiniciava ao fim de um certo tempo.

O vídeo utilizado, não foi o mesmo de outros testes, por limitações do CPU, uma vez que tinha de se codificar os três conteúdos simultaneamente. Assim foi utilizado um vídeo com menor resolução espacial (320x240), que também estava codificado em H.264/AVC.

No ambiente de teste foi apenas possível testar o conceito da solução, isto é, não foi possível aplicar um ambiente de *multicast*. Para que tal acontecesse seria necessário um DSLAM para simular internamente (sem passar pela Internet) uma rede de acesso, uma vez que é de necessário que o vídeo passe pela porta WAN do CPE (*modem* neste caso), afim de testar a adaptação dinâmica. Por outro lado, sem DSLAM os pedidos *multicast* dos clientes não chegam ao outro ponto da rede, mais especificamente ao MANE. Assim, foi necessário simular os pedidos dos clientes no MANE. Contudo, o papel do *multicast* não tem um cariz decisivo no funcionamento da solução, isto porque o objectivo é testar a adaptação, sendo o meio de comunicação utilizado para tal transparente para a solução. Deste modo, foi testado num ambiente de *unicast*, apenas com um cliente.

A solução encontrada para passar o vídeo pela porta WAN do *modem* (porta essa onde se obtém a informação para a adaptação dinâmica) sem recorrer ao uso de um DSLAM, foi a de ligar os dois *modems* a duas redes diferentes (neste caso de operadores diferentes). Deste modo, o *streamer* enviava os conteúdos com o endereço IP externo do *modem2*, com um determinado porto de destino. O *modem2* tinha um NAT no qual reencaminhava para o cliente tudo o que recebesse na porta especificada.

De referir que o vídeo usado nos testes tinha como ritmo binário para a melhor qualidade do vídeo de 400 kb/s, logo o *uplink* do *modem1* não constituía um ponto de limitação. Assim, o ponto da rede limitativo era a Internet, o qual não era passível de ser controlado. Os testes foram feitos à noite, de forma a ter maior disponibilidade da rede e com a excepção de qualquer problema na Internet, o vídeo foi recebido em boas condições. Contudo, a Internet é um elemento bastante interessante dentro do ambiente de teste, isto porque permite alguma aleatoriedade na qualidade de recepção do cliente, permitindo assim testar a adaptação dinâmica da qualidade. É importante realçar que o *modem2*, não está a receber qualquer informação na porta WAN para além do vídeo, factor que poderia influenciar os resultados.

Os testes efectuados tinham como objectivo validar a solução encontrada. Assim, realizou-se dois tipos de testes:

- Tempo de adaptação. Teve como objectivo avaliar a rapidez com que a solução demora a adaptar os conteúdos (após ter sido accionada), assim como, o tempo que o cliente espera até visualizar o vídeo com a nova qualidade.

- Qualidade na adaptação. Teve como objectivo verificar o impacto visual, que a mudança de qualidade tem para o cliente, assim como, se a qualidade posterior foi melhor que a qualidade anterior. Para este último caso, foi limitado o *switch* a 256 kb/s e foram enviadas três qualidades: a melhor qualidade com 300 kb/s, outra com 225 kb/s e a pior com 150 kb/s. Assim, começando o cliente a receber a melhor qualidade, quando houver actualização desce para a segunda qualidade, devido à limitação imposta no *switch*. Para se ilustrar a diferença na qualidade de imagem, tiraram-se *snapshots*, antes e depois da adaptação. Foi mostrada a diferença de qualidade da imagem recebida pelo cliente, quando o vídeo passa da melhor qualidade (maior ritmo) para a qualidade de 225 kb/s.

Para ambos os testes, foram utilizadas cinco pessoas, para analisar a qualidade na adaptação.

## 5.4 Resultados

Na Tabela 8, estão descritos os resultados do tempo de adaptação, aquando da actualização de um cliente. Nesta tabela, a coluna mais à esquerda, o intervalo de tempo representa o lapso entre os dois pedidos ao CPE (para obter o “ritmo de menor período”). Para o intervalo de tempo, não faz sentido o desvio padrão, uma vez que este valor é fixo. Na segunda coluna mais à esquerda, está o tempo de adaptação, isto é, o tempo que demora o *script* de actualização a ser executado, este tempo inclui também o intervalo de tempo referido anteriormente. No cálculo desta medida foi utilizada a aplicação *time* do Linux, que permite saber o tempo exacto de execução de um *script*. De salientar que este campo foi variando com os diferentes observadores, daí também estar presente na tabela o desvio padrão. Na coluna mais à direita está o tempo do impacto no cliente, isto é, o tempo que decorre entre o instante em que termina o processamento do *script* e o momento que o cliente se apercebe visualmente da mudança. Neste campo só é possível uma medição subjectiva.

**Tabela 8 – Resultados dos testes da solução, relativamente ao tempo de adaptação.**

Intervalo de tempo (s)	Tempo de actualização com mudança de qualidade (s)	Desvio padrão do Tempo de adaptação	Tempo do impacto no cliente (s)
10	30,55	0,14	< 1

Os resultados do tempo de adaptação, indicaram cerca de meio minuto para adaptar cada cliente, não parece ser um tempo excessivo, por se tratar de um meio de transmissão (cobre) sem mudanças bruscas de qualidade.

O tempo de impacto no cliente depende de três factores: tempo de transmissão da nova qualidade do conteúdo; tempo de espera nos buffers do receptor (VLC), tipicamente com um valor máximo de 300 ms; tempo de descodificação do conteúdo que depende da capacidade do dispositivo do cliente, sendo este factor tipicamente o que menos contribui para o tempo de impacto no cliente. O tempo de transmissão é o factor mais inconstante (porque depende das condições da rede), podendo aumentar significativamente o tempo de impacto no cliente. Como não foi possível aferir o tempo de impacto no cliente, apenas assegurar que foi inferior a 1 segundo (quase instantâneo), é possível deduzir que o factor tempo de transmissão teve pouco impacto neste valor, o que significa uma boa qualidade na transmissão dos conteúdos.

Na análise do impacto visual aquando da mudança de qualidade, foram considerados os seguintes parâmetros: efeito de bloco, distorção das cores, desfocagens e paragens. Nas mudanças de qualidade observadas (estes testes foram realizados em simultâneo com os testes de tempo de adaptação), não se notou qualquer paragem no movimento do vídeo e muito raramente notou-se distorção de cores. Contudo, notaram-se desfocagens e efeito de bloco. Estes dois últimos artefactos, apenas desapareciam totalmente aquando da mudança de cena (nova trama I). É de salientar que, após a mudança de qualidade e enquanto não ocorria nenhuma trama I, a melhoria de qualidade apenas se reflecta essencialmente, nos pixéis onde havia movimento. Isto está inteiramente relacionado com a lógica de funcionamento do vídeo, uma vez que as tramas P apenas vão codificar as diferenças, logo apenas esses pixéis (correspondentes ao movimento), vão ser codificados com a nova qualidade. Sendo todos os outros pendurados na trama I anterior que se encontra degradada.

Para analisar a qualidade antes e após a adaptação, foram feitos testes baseados em QoE. Os resultados são apresentados na Tabela 9. De salientar que se considerou que o depois da adaptação seria quando a imagem estivesse estável (depois de uma trama I).

**Tabela 9 – Análise de QoE antes e após a adaptação.**

QoE antes da adaptação	Desvio padrão do QoE antes da adaptação	QoE depois da adaptação	Desvio padrão do QoE depois da adaptação
1,8	0,27	3,9	0,41

Os resultados confirmam que a qualidade depois da adaptação é muito melhor do que a anterior. Pois antes da adaptação o QoE médio foi de 1,8, enquanto que depois da adaptação o QoE foi de 3,9.



Na Figura 26, está ilustrada a diferença entre o vídeo recebido com a melhor qualidade (imagem da esquerda) com um ritmo de 300 kb/s e a qualidade intermédia (imagem do lado direito) com um ritmo de 225 kb/s. Verificou-se que quando o cliente recebia os conteúdos com a melhor qualidade, como o ritmo excedia a sua capacidade de recepção (limitado a 256 kb/s), ocorriam perdas e consequentemente o vídeo era degradado. Por outro lado, quando recebia a segunda qualidade o cliente já conseguia receber sem ter perdas, uma vez que o ritmo do canal era inferior ao ritmo máximo da sua ligação.



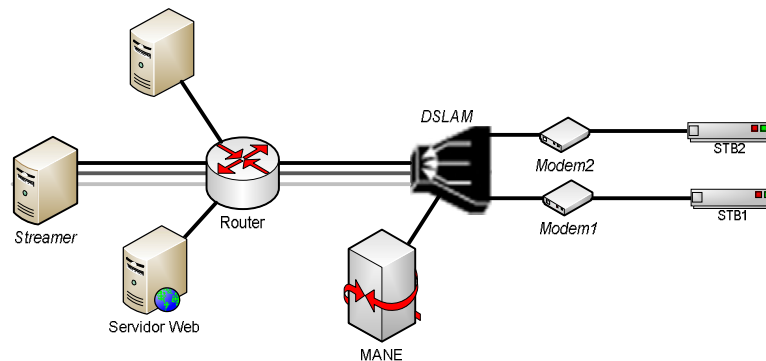
**Figura 26 – Imagens com a diferença de qualidade antes e após a adaptação.**

A Figura 26 demonstra que é melhor enviar um canal com menor qualidade do que perder pacotes num canal de alta qualidade.

A recodificação dos fluxos no servidor ocupou cerca de 110% do CPU e 3,4% de memória.

## 5.5 Condicionalismos da implementação num ambiente real

Na Figura 27, está ilustrado um exemplo de um ambiente real. Neste caso o MANE encontra-se ligado a uma carta do DSLAM (no lado digital da rede), com o intuito de servir apenas os clientes desse DSLAM. Contudo, isto é apenas um exemplo, uma vez que nada impede o MANE de estar num *router* no núcleo da rede, desde que este tenha capacidade computacional para servir todos os clientes.



**Figura 27 – Implementação num ambiente real.**

O servidor *Web* representado na Figura 27 é o elemento responsável por fornecer ao equipamento do cliente, o IP *multicast* a que este se deve associar. Este servidor, com esta solução, deve também actualizar o módulo da “lista de canais” presente no MANE (ver Figura 23).

Num ambiente real, seria também necessário analisar os *bytes* recebidos num *Permanent Virtual Circuit* (PVC) específico de vídeo. Tipicamente na rede de acesso DSL, é assinado um circuito virtual, especificamente para vídeo entre o DSLAM e o modem. No ambiente de teste produzido em 5.3, o cálculo da largura de banda era efectuado pelos bytes recebidos pelo CPE na porta WAN e garantia-se que o cliente não tinha tráfego adicional. Contudo, num ambiente real, o cliente poderia estar a receber tráfego em paralelo (por exemplo, *peer to peer*) o que iria desvirtuar a largura de banda obtida, daí ser necessário obter a largura de banda no PVC específico de vídeo.

Com a implementação desta solução num ambiente real o módulo de *iptables* descrito na arquitectura de 5.2 (na Figura 23) deixaria de existir. A razão para a sua exclusão deve-se ao ambiente *multicast*, intrínseco ao serviço IPTV, onde vários clientes podem estar associados ao mesmo canal, isto é, ao mesmo endereço *multicast*. Neste contexto, se o MANE decidir alterar a qualidade para um cliente específico, as regras de *iptables* irão fazer com que todos os clientes servidos por esse MANE recebam a nova qualidade. Assim, foram propostas duas soluções possíveis para este problema: no próprio DSLAM ou via TR-135.

### 5.5.1 Solução no DSLAM

Num ambiente real, com a capacidade do DSLAM de tratar os pedidos IGMP, a lógica de filtrar e redireccionar os conteúdos deixa de ser uma função do MANE para passar a ser uma função do DSLAM. Neste caso, os diferentes níveis de qualidade de um conteúdo teriam de chegar aos DSLAMs (como está ilustrado na Figura 27), sendo estes os responsáveis pela filtragem, através da sua tabela de encaminhamento, onde é estabelecida a associação entre o endereço IP e o porto físico.

Contudo, na tabela de encaminhamento do DSLAM teria de ser adicionado um terceiro campo com o porto lógico, para que os diferentes níveis de qualidade do *multicast* pudessem ser distinguidos. Isto porque no caso do MANE se encontrar entre o servidor e o DSLAM, e se duas ou mais STBs (associadas ao mesmo DSLAM, como na Figura 27) pedissem o mesmo canal, caso o MANE entendesse que uma das STBs devia receber uma qualidade inferior, então passariam a chegar dois fluxos ao DSLAM. Caso a sua tabela de encaminhamento tivesse apenas a associação entre o porto e o endereço IP, como a diferença entre os dois fluxos está apenas na porta lógica, então ambos seriam recebidos nas duas STBs. Por outro lado, no caso do DSLAM não tratar os pedidos *multicast* e o MANE recebê-los directamente, quando o cliente fizesse um pedido, passaria automaticamente a receber todos os níveis de qualidade do mesmo conteúdo. É assim necessário acrescentar mais um campo à tabela de encaminhamento do DSLAM, que faria a função equivalente ao uso do *iptables* na solução apresentada em 5.2.

A porta lógica na tabela de encaminhamento do DSLAM teria de ser controlada pelo MANE. No caso do porto introduzido pelo MANE não corresponder à de melhor qualidade do conteúdo, o DSLAM teria de reescrever o porto de destino dos pacotes, uma vez que o cliente continua à espera dos pacotes no mesmo porto. Assim o DSLAM deve interpretar os pedidos dos clientes e redireccionar ou informar o MANE desses mesmos pedidos. Deste modo via *web service*, por exemplo, seria possível actualizar o MANE, mais especificamente o controlador IGMP (ver Figura 23), relativamente ao tipo de mensagem enviado pelo cliente, assim como, com os IPs do canal e do próprio DSLAM. O MANE com essa informação, iria alterar a tabela de encaminhamento, por exemplo, via *web service*.

### 5.5.2 Solução no TR-135

Uma outra solução para o problema passa por utilizar a especificação da *DSL Forum*, TR-135 [27]. O TR-135 define um modelo de dados para gestão remota das STBs via CWMP (TR-069). Esta norma permite fazer actualizações ao *firmware* das STBs, configurar a STB e efectuar diagnósticos. O ACS pode monitorizar a performance da STB através: parâmetros de QoS (ritmo binário, *jitter*, perda de pacotes), parâmetros QoE e estatísticas úteis (como por exemplo, quantas STBs estão activas em cada instante). Esta especificação pode ser sempre usada na solução real, para obter informação da capacidade do cliente, isto é, em vez de obter informação ao CPE, poder-se-ia obter directamente na STB, tendo informação bastante útil para além do ritmo binário.

Para além de obter parâmetros específicos da STB, para a resolução do problema identificado em 5.5 esta norma pode ser bastante útil, uma vez que permite forçar o cliente a fazer nova associação a um novo serviço – nova qualidade. Neste caso, esta nova associação corresponde a um novo grupo *multicast*, logo as qualidades teriam de deixar de ser distinguidas pelos portos, passando a ser identificadas directamente pelo IP *multicast*. Assim quando o MANE se aperceber de que a qualidade

para o cliente não é a mais indicada, limita-se a enviar um pedido para o ACS, a pedir a alteração do *Uniform Resource Identifier* (URI) do serviço, que na realidade é apenas a alteração da qualidade. Contudo na especificação (TR-135) este URI é apenas utilizado para teste, isto é, quando há algum problema com o cliente, força-se um URI de teste para detectar a anomalia. No entanto esta funcionalidade pode ser evoluída no sentido de efectuar a mudança de qualidade. Deste modo, o MANE deixaria de ter a função de adaptar o conteúdo, ficando apenas responsável por decidir quando esta acontece.

O TR-135 iria também permitir que a sobrecarga introduzida pelas diferentes qualidades, deixasse de existir, uma vez que o *streamer* limitava-se a enviar os canais pedidos, ou seja, na rede nunca iriam estar a ser enviadas qualidades dos conteúdos multimédia que não estivessem a ser observados pelos clientes.

Esta norma também permite a realização de adaptações personalizadas na STB de cada cliente. Por exemplo, ao nível dos *buffers* de *de-jittering*, onde se pode aumentar o tamanho dos *buffers* para ser mais tolerante ao *jitter*, contudo isto irá provocar um maior atraso na visualização dos conteúdos.

Esta norma tem bastantes vantagens na sua utilização, todavia não foi utilizada na tese, uma vez que ainda não é suportada pelas STBs. Um outro problema que poderá existir prende-se com o facto desta solução ser extremamente intrusiva, uma vez que a mudança de canal tem sempre um impacto visual bastante grande para o cliente. Para que tal não aconteça, a alteração de qualidade tem de ser extremamente rápida de modo a que o cliente note o mínimo possível, essa mudança. Existe actualmente uma grande preocupação em diminuir a latência na mudança de canal, exemplo disso é a criação de um mecanismo, o *Instant Channel Change* (ICC) [28], que consiste em enviar o conteúdo directamente em *unicast* por um servidor perto do cliente, enquanto o conteúdo *multicast* não chegar ao cliente.

## 6 Conclusão

O serviço IPTV utiliza frequentemente como suporte as redes DSL e têm como principal ponto de falha a rede de acesso. Este serviço é muito sensível a perdas e a atrasos. Por conseguinte, o principal objectivo deste trabalho foi o estudo e desenvolvimento de um modelo de adaptação de vídeo à capacidade de recepção dos clientes de IPTV. Com esse intuito, criou-se uma solução de degradação de conteúdos (descartando pacotes IP) num MANE. Contudo, demonstrou-se que tal solução não foi adequada, devido ao grande impacto visual aquando de pequenos cortes no vídeo. Deste modo, foi criada uma segunda técnica, onde o servidor envia diferentes qualidades do mesmo conteúdo. A qualidade que o cliente irá receber é decidida por um MANE, antes dos conteúdos chegarem à rede de acesso. Esta solução demonstrou ser melhor que a anterior, permitindo maiores reduções de débito sem um elevado impacto visual.

Para aplicar a técnica de adaptação a um princípio de adaptação dinâmico, foi estudado um mecanismo para obter informação do cliente, recorrendo para isso à solução *Axess*, que se baseia na especificação CWMP do *Broadband Forum*.

Com o objectivo de agregar os dois mecanismos referidos anteriormente (adaptação dinâmica e técnica de degradação dos conteúdos), foi definido um modelo de uma solução baseada num MANE, que envia para o cliente a melhor qualidade que este pode receber, realizando actualizações periódicas de capacidade destes em receber os conteúdos. De salientar que por impossibilidade de recriar um ambiente real de IPTV (necessidade de DSLAMs), para a realização deste trabalho, não foi possível criar e testar uma solução nesse mesmo ambiente, contudo foi dedicada uma secção (5.5), onde se aborda o modo como esta solução poderia ser implementada nesse ambiente. Numa implementação real o problema identificado relativamente ao ambiente *multicast* fornece duas opções: a alteração dos DSLAMs ou a utilização da norma TR-135 nas STBs. A melhor escolha recai claramente na norma TR-135, que contudo, tem de ser experimentada e verificado o impacto visual que a mudança de canal tem no cliente. No entanto, a norma TR-135 pode ser sempre usada para obter informação da qualidade de recepção da STB do cliente, em detrimento da aquisição de informação no CPE.

Relativamente aos resultados, confirmou-se que a degradação num MANE, não era uma solução viável, devido ao grande impacto visual que o descarte ao nível do pacote IP provocava. Com esta técnica, o limite máximo de redução de ritmo binário foi de 5%, o que é muito pouco e sem interesse de aplicação real. A degradação no servidor com limite imposto de 50% de redução, foi aceitável do ponto de vista de experiência do cliente. No modelo desenvolvido, observou-se que o impacto da adaptação é reduzido, sem paragens na mudança de qualidade. Quando o cliente não tinha capacidade para receber uma determinada qualidade, os resultados demonstraram que é preferível o cliente receber uma qualidade pior, do que receber uma qualidade melhor com perdas. As observações também mostraram,

que o tempo do cliente visualizar essa nova qualidade sem desfocagens e efeito de bloco, depende da própria codificação do vídeo (ocorrência de tramas I). Em relação ao tempo que é necessário para a realização da adaptação, cerca de meio minuto para cada cliente, não parece ser um tempo excessivo, por se tratar de um meio de transmissão (cobre) sem mudanças bruscas de qualidade. Contudo este tempo pode ser drasticamente reduzido se for diminuído o “ritmo com menor período” (descrito na secção 4.2).

Neste trabalho, ficou bem patente a flexibilidade desta solução, no modo como se adapta a qualidade do vídeo à capacidade de recepção dos clientes, uma vez que suporta qualquer tipo de adaptação feito no servidor, isto é, as diferentes qualidades podem ser estabelecidas de acordo com as exigências do fornecedor do serviço IPTV, sem que para isso seja necessário alterar a solução. Por outro lado, a solução como foi concebida é em muito idêntica, ao conceito da nova norma de vídeo H.264/SVC. Pois tal como descrito em [8], existem vários modos de transmissão das camadas em H.264/SVC. Um dos quais, consiste no envio das camadas como fluxos independentes até ao cliente, o que é em tudo idêntico à solução apresentada. A única diferença para a solução actual é que esta é menos eficiente uma vez que envia várias qualidades, onde cada uma delas constitui um fluxo independente. No H.264/SVC tal não acontece, ou seja, existe uma camada base que é enviada na pior qualidade e depois os outros fluxos são apenas melhorias dessa mesma camada.

Os impactos negativos da solução são: a sobrecarga no servidor devido à codificação dos vários níveis de qualidade; na rede (do núcleo da rede até à rede de acesso) com o tráfego adicional das diferentes qualidades do mesmo conteúdo. Contudo, este problema é suavizado pelos elevados débitos dos núcleos de rede actuais, logo pode não constituir um problema real.

Deste modo, foram dadas contribuições importantes na área de adaptação do vídeo, demonstrando uma técnica de degradação de conteúdos viável e outra sem aplicação prática. Criou-se uma solução totalmente inovadora, para a adaptação do vídeo, que pode ser aplicada num ambiente real de IPTV. Contudo, para obtenção desta solução, foram dadas outras contribuições no que respeita a análise e realização de testes de duas técnicas de adaptação da qualidade do vídeo. As referidas técnicas tinham como principal referência a recente norma de codificação de vídeo escalável, o H.264/SVC, que foi desenhada com o objectivo de fornecer diferentes níveis de qualidade aos clientes. Esta solução é também inovadora no modo como se obtém a informação do cliente, utilizando para isso a especificação do *Broadband Forum* de gestão remota de CPEs. Esta tese contribuiu ainda com propostas de implementação da solução num ambiente real de IPTV.

## 6.1 Trabalho futuro

O ponto principal no trabalho futuro, é testar a solução num ambiente real de IPTV, tentando resolver o problema identificado em 5.5. Para solucionar o problema terá de ser testada a solução com a norma TR-135, de modo a verificar o impacto visual que a mudança de qualidade tem no cliente, incluindo a utilização do mecanismo ICC, para a mudança rápida de qualidade. Caso esta solução não funcione, é necessário optar pela solução no DSLAM, a qual parece ser mais complexa de implementar. Independentemente deste problema, seria também importante recorrer ao uso do TR-135 para obter informações directamente da STB, em detrimento do CPE.

O segundo ponto interessante seria adaptar o modelo proposto para um ambiente *Video on Demand* (VoD), onde faria sentido que o MANE estivesse no servidor, uma vez que os conteúdos são enviados para um cliente específico. Na arquitectura do MANE, pelo menos o módulo “Controlador IGMP” específico para *multicast* teria de ser adaptado para um contexto de *unicast* (ambiente utilizado no VoD).

Mediante o impacto que a mudança de qualidade tem, deve-se também definir o intervalo da actualização da capacidade de recepção dos clientes. Esses intervalos não devem ser muito grandes, pois acarretam grandes atrasos na adaptação, contudo intervalos pequenos podem provocar adaptações desnecessárias. Como o principal factor de medição da qualidade do serviço é a experiência do cliente, é importante referir que tal como indicado em [29], os clientes têm a tendência para não gostarem de mudanças frequentes na qualidade dos conteúdos e na realidade preferem uma qualidade inferior mas estável a uma qualidade superior com muitas variações. O diagrama da Figura 20, tem como objectivo reduzir estas mudanças de modo a que o cliente não esteja constantemente sujeito a mudanças de qualidade.

Seria também necessário a análise da informação dos PVCs na interface WAN do CPE, uma vez que num ambiente real, é definido um PVC entre o DSLAM e o CPE específico para vídeo. Caso tivesse grande impacto no vídeo, diminuía-se a qualidade durante o tempo que houvesse mais tráfego em paralelo.

Num âmbito mais ambicioso seria importante que esta solução adaptasse os conteúdos ao tipo de dispositivos, recorrendo ao *Axess* para obter informação das características do dispositivo, como por exemplo a resolução espacial. Desse modo na métrica a aplicar já faria sentido usar a resolução espacial na adaptação dos conteúdos.

Para finalizar, seria importante explorar o conceito das percentagens dos ritmos binários presentes na técnica de adaptação escolhida (descritos na secção 3.2.2). Como referido nessa secção, agrupar os canais em tipos e utilizar ritmos idênticos apenas para tipos idênticos de conteúdos. No caso de existir adaptação ao tipo de dispositivo até se poderia considerar a última qualidade com uma percentagem do

ritmo binário da melhor qualidade inferior a 50%, por exemplo, com 25%, isto claro dependendo do dispositivo em causa.



## Referências

- [1] L. Harte, “*IPTV Basics*”. Fuquay-Varina USA: Althos Publishing, 2006.
- [2] R. Smith, “*IPTV Global Forecast - 2007 to 2011 Semiannual IPTV Global Forecast Report*”, Outubro de 2007. Disponível em [http://www.mrgco.com/TOC\\_IPTV\\_GF1007.html](http://www.mrgco.com/TOC_IPTV_GF1007.html).
- [3] “*PT arranca com teste comercial de IPTV em Junho*”, 15 de Fevereiro de 2007. Disponível em <http://exameinformatica.clix.pt/noticias/internet/214727.html>
- [4] R. Murer, “*O que é IPTV*”. Softv 2007. Disponível em <http://www.softv.com.br/artigos/o-que-e-iptv.pdf>.
- [5] <http://www.broadband-forum.org/>
- [6] F. Pereira, Comunicação de Áudio e Vídeo. Disponível em <http://www.img.lx.it.pt/~fp/cav/com.htm>.
- [7] Ye-Kui Wang et al., “*System and Transport Interface of SVC*”, IEEE, vol. 17, pp. 1149-1163, Setembro de 2007.
- [8] S. Wenger, Y. Wang, e T. Schierl, “*Transport and Signaling of SVC in IP Networks*”, IEEE, vol. 17, pp. 1164-1173, Setembro de 2007.
- [9] H. Schwarz, D. Marpe, T. Wiegand, “*Overview of the Scalable Video Coding Extension of the H.264/AVC Standard*”, IEEE, vol. 17, pp. 1103-1120, Setembro de 2007.
- [10] T. Wiegand, et al., “*Joint Draft ITU-T Rec. H.264 | ISO/IEC 14496-10 / Amd.3 Scalable video coding*”, Geneva, Switzerland, 5 de Julho de 2007. Disponível em [http://ftp3.itu.ch/av-arch/jvt-site/2007\\_06\\_Geneva/JVT-X201.zip](http://ftp3.itu.ch/av-arch/jvt-site/2007_06_Geneva/JVT-X201.zip).
- [11] J. Ohm, “*Standardization in JVT: Scalable Video Coding*”, ITU-T VICA Workshop, 22-23 Julho de 2005. Disponível em [www.itu.int/ITU-T/worksem/vica/docs/presentations/S3\\_P2\\_Ohm.pdf](http://www.itu.int/ITU-T/worksem/vica/docs/presentations/S3_P2_Ohm.pdf).
- [12] V. Swaminathan, M. Bourges-Sévenier, “*MPEG-J White Paper*”, Julho de 2005. Disponível em <http://www.chiariglione.org/mpeg/technologies/mp04-j/index.htm>.
- [13] H. J. Park et al., “*QoS negotiation for IPTV service using SIP*”, vol. 2, Fevereiro de 2007.
- [14] T. Rahrer, R. Fiandra, S. Wright, “*Triple-play Services Quality of Experience (QoE) Requirements*”, 13 de Dezembro de 2006. Disponível em <http://www.broadband-forum.org/technical/download/TR-126.pdf>.
- [15] H. Choi et al., “*Dynamic and Interoperable adaptation of SVC for QoS-Enabled Streaming*”, IEEE, vol. 53, pp. 384-389, Maio de 2007.
- [16] R. El-Marakby e D. Hutchison, “*Scalability improvement of the real time control protocol*”, Elsevier, vol. 28, pp. 136-149, Julho de 2004.

- [17] A. Hutter et al., "Automatic adaptation of streaming multimedia content in a dynamic and distributed environment", IEEE, vol. 3, Setembro de 2005.
- [18] L. Celetto, K. Grüneberg, T. Zahariadis, "SVC Payload Format and Signalling", ASTRALS Consortium, Dezembro de 2006.
- [19] M. Lohse, "The Network-Integrated Multimedia Middleware (NMM): Basic Introduction", Julho de 2005. Disponível em <http://graphics.cs.uni-sb.de/NMM/current/Docs/intro/>
- [20] M. Lohse, "Network-Integrated Multimedia Middleware, Services, and Applications", Saarbrücken 2005. Disponível em [http://graphics.cs.uni-sb.de/~mlohse/publications/phd/marco\\_lohse-nmm-phd-2005.pdf](http://graphics.cs.uni-sb.de/~mlohse/publications/phd/marco_lohse-nmm-phd-2005.pdf).
- [21] <http://www-x.antd.nist.gov/nistnet/>
- [22] "Methods for subjective determination of transmission quality", 30 de Agosto de 1996. Disponível em <http://www.itu.int/rec/T-REC-P.800-199608-l/en>.
- [23] <http://www.videolan.org/vlc/>
- [24] <http://mpeg4ip.sourceforge.net/>
- [25] <http://developer.apple.com/opensource/server/streaming/>
- [26] J. Bernstein, T. Spets, "CPE WAN Management Protocol", Maio de 2004. Disponível em <http://www.broadband-forum.org/technical/download/TR-069.pdf>.
- [27] M. Quacchia, W. Lupton, G. Straub, "TR\_135 – Data Model for a TR-069 Enabled STB", Dezembro de 2007. Disponível em <http://www.broadband-forum.org/technical/download/TR-135.pdf>.
- [28] D. Banodkar, K.K. Ramakrishnan, S. Kalyanaraman, A. Gerber, O. Spatscheck, "Multicast Instant Channel Change in IPTV Systems", Bangalore, IEEE, pp. 370-379, Janeiro de 2008.
- [29] L. Burghstahler, K. Dolzer, C. Hauser, J. Jähnert, S. Junghans, C. Macián and W. Payer, "Beyond Technology: the missing pieces of QoS success", In: Proc. of ACM SIGCOMM workshop on Revisiting IP QoS, Session 2, Karlsruhe, Germany, 2003, pp. 121-130.
- [30] <http://www.netfilter.org/documentation/index.html#documentation-howto>
- [31] <http://www.iu.hio.no/teaching/materials/MS004A/index.phtml?show=L30.en&week=4>
- [32] <http://pt.wikipedia.org/wiki/Multicast>
- [33] "Introduction to IGMP for IPTV Networks", White Paper, Juniper Networks, Outubro de 2007.
- [34] <http://www.axiros.com>
- [35] B. Guedes, R. Pereira, T. Vazão, A. Varela, "Simple Media-Aware Packet Discard Algorithms" in "International Conference on Information Networking 2009", Chiang Mai Thailand, January 2009.

## Anexos

### Anexo A – Tecnologias de suporte

#### Netfilter e Iptables

O Netfilter é uma *framework* do *kernel* Linux que implementa a *firewall iptables* (sucessor do *ipchains*). Com o *iptables* é possível filtrar e alterar pacotes IP [30]. É assim possível obter o tráfego que passa pela máquina e tomar uma decisão sobre o seu destino.

O *iptables* funciona com base em tabelas, dentro das quais existem conjuntos de regras. Cada regra possui um filtro (por exemplo, endereço ou porta de origem ou de destino, tipo de protocolo, entre outros) para determinar se esta é ou não aplicada ao pacote IP e uma acção a tomar caso o filtro se aplique. Estes três elementos (tabelas, cadeias de regras e acções) vitais para o funcionamento do *iptables* serão explicados mais em detalhe.

No *iptables* existem quatro tabelas:

- *Mangle*: tabela que permite fazer alterações nos campos do cabeçalho IP relativos à prioridade dos pacotes, assim como marcar pacotes para os *routers* descartarem mais facilmente.

- NAT: tabela que permite redireccionar o tráfego, assim como alterar portas e IPs. A alteração da porta de destino designa-se por *port forward*.

- *Raw*: é principalmente usado para marcar os pacotes. O *iptables* é uma firewall que mantém o estado das ligações. Os pacotes podem ser marcados de modo a que o estado da ligação não possa ser analisado, ou para permitir a análise do trajecto percorrido pelo pacote IP no *iptables*.

- *Filter*: é usado para filtrar pacotes, sendo possível descartar ou aceitar os pacotes.

Uma acção é aplicada a um determinado pacote se o resultado do filtro aplicado for positivo. Existem vários tipos de acções, das quais se distinguem o ACCEPT, o DROP e o QUEUE. O DROP permite descartar o pacote, enquanto que o ACCEPT faz com que o pacote continue o seu trajecto (pacote aceite). O QUEUE é uma acção especial, que põem o pacote numa fila afim de ser tratado num processo utilizador.

É contudo possível fazer um “tratamento personalizado” dos pacotes IP, de uma maneira mais rápida do que o processo utilizador, recorrendo a módulos do kernel. O módulo pode já existir feito, ou ser construído de raiz.

As cadeias de regras predefinidas são: INPUT, OUTPUT, FORWARD, POSTROUTING e PREROUTING. Contudo é possível o utilizador definir a sua própria cadeia de regras.

Uma vez que as tabelas, *nat* e *filter*, são as únicas a serem utilizadas no projecto, as outras duas não foram contempladas no diagrama. A tabela *filter* é constituída pelas cadeias de regras, INPUT, OUTPUT e FORWARD. A tabela NAT é constituída por: POSTROUTING, PREROUTING e OUTPUT.

Na Figura 28, está descrito o fluxo normal de um pacote IP. Quando um pacote chega ao módulo de *iptables* via interface de rede, a primeira cadeia de regras é o PREROUTING que permite alterar o IP ou a porta de destino. Depois é feita a decisão de *routing* do pacote e analisado o IP de destino. Caso este seja a própria máquina então ele vai para a *chain* INPUT, se ele passar esta *chain*, qualquer processo local que esteja a espera do pacote irá recebê-lo.

Caso o pacote tenha outro IP de destino, se o *kernel* não tiver *forward* activo ou se não souber como fazê-lo, o pacote é descartado. Se o *forwarding* tiver activo e o pacote for destinado a outro IP, o pacote vai para a cadeia de regras FORWARD, se for aceite o pacote será enviado para o POSTROUTING. No POSTROUTING é possível alterar o IP e a porta de origem.

Finalmente um programa correndo na própria máquina pode enviar pacotes, estes passam pela cadeia de regras *nat* OUTPUT (com funções idênticas ao de PREROUTING), seguindo o pacote para a cadeia *filter* OUTPUT, caso o pacote seja aceite é enviado para o POSTROUTING descrito acima. Posteriormente é enviado para o seu destino.

De salientar que apenas a tabela *filter* pode descartar os pacotes. Em caso de descarte (*drop*), o pacote já não vai ser analisado pelas outras cadeias de regras. Caso contrário continua a percorrer o diagrama.

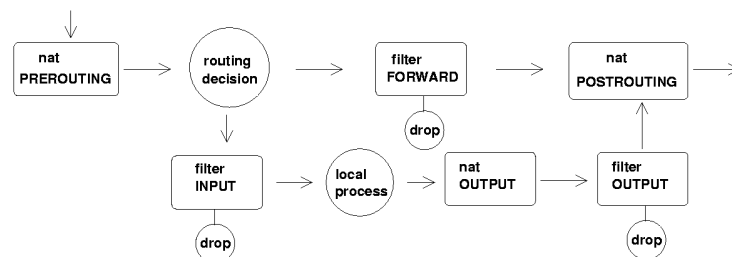


Figura 28 – Diagrama do *iptables* para as tabelas de NAT e *filter*.<sup>46</sup>

<sup>46</sup> Figura extraída de [31].

## Tipos de comunicação

O *broadcast* consiste na difusão da informação para todos os receptores, enquanto que o *unicast* consiste na transmissão ponto a ponto, ou seja o emissor envia a informação para um receptor específico [32]. O *broadcast* desperdiça recursos se nem todos os receptores tiverem “interesse” em receber a informação do emissor, funciona apenas em ambientes pequenos. Por outro lado, o *unicast* pode levar a um consumo exagerado de recursos, se muitos utilizadores quiserem obter a informação do emissor, visto que neste caso o emissor terá de enviar  $n$  vezes a mesma informação, onde  $n$  é o número de receptores. Deste modo o *multicast* (também designado como *broadcast* selectivo) diferencia-se destes dois tipos de comunicação, na medida em que o emissor apenas transmite para os utilizadores pertencentes a um determinado grupo. Deste modo, o *multicast* torna-se o tipo de comunicação utilizado para aplicações que necessitam de elevada largura de banda, como é o caso do IPTV. O *multicast* pode ser um tipo de comunicação, um para muitos ou muitos para muitos, isto significa que pode haver mais do que uma fonte. Na Figura 29, está ilustrado um exemplo destes três conceitos referidos, onde o emissor se encontra representado por uma bola vermelha, os emissores que recebem a informação por uma bola verde e os que não recebem por uma bola amarela.

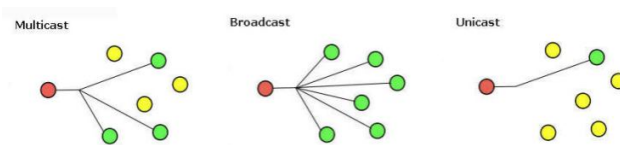


Figura 29 – Exemplo de *broadcast*, *unicast* e *multicast*.<sup>47</sup>

O *multicast* acarreta deste modo, uma redução significativa da largura de banda, num cenário onde exista um cliente e múltiplos receptores, como é o caso do IPTV.

Ao contrário do *broadcast* e *unicast*, o *multicast* não tem limites físicos nem geográficos. Isto é, o receptor pode associar-se a um grupo estando este algures numa rede interna ou na Internet, desde que no caminho entre o emissor e o receptor o *multicast* seja permitido.

### **Multicast**

O *multicast* é o conceito de transmitir simultaneamente informação, de um emissor para vários receptores [33]. Os emissores enviam os conteúdos para um determinado grupo *multicast* e os receptores necessitam apenas de se associar a esse grupo. Um exemplo análogo rudimentar é uma

---

<sup>47</sup> Figura extraída de [32].

emissão de rádio ou televisão, na medida em que é necessário sintonizar o equipamento receptor para obter a transmissão.

Um grupo *multicast* junta o emissor e um conjunto de receptores interessados nos conteúdos que o emissor difunde. Um canal *multicast* é a combinação entre o endereço do emissor (endereço de origem nos pacotes IP) e o endereço do grupo *multicast* (endereço de destino nos pacotes IP) para o qual a fonte enviará os conteúdos. Um grupo *multicast* é sempre identificado pelo endereço de destino.

Para receber um determinado *streaming* de *multicast* os receptores devem mandar um pedido de associação (“*join*”), recorrendo ao envio de uma mensagem IGMP, para o *router multicast* local. Existem três versões do IGMP: IGMPv1, IGMPv2 e IGMPv3. Sendo que as mais usadas são a versão dois e três, pois permitem aos receptores associarem-se ou não a um determinado grupo. A versão 1 não tem a mensagem “*leave*”, não permitindo os receptores desassociarem-se. A versão 3 é compatível com a 2.

O IP *multicast* tem uma gama reservada que vai desde o 224.0.0.0 até ao 239.255.255.255, sendo também conhecida endereços da classe D. O elemento de rede (por exemplo o *switch*) que envia os conteúdos para o cliente, guarda uma tabela de *forwarding* por grupo e por porta do *switch* baseado nas mensagens IGMP de *leave/join*. Um exemplo do funcionamento é o cliente envia um IGMP *join* para o 255.1.1.1, o *switch* recebe o *join* e adiciona essa porta à tabela de *forwarding* do IP de *multicast* 255.1.1.1, o destino dos pedidos dos clientes é sempre o IP *multicast* e o MAC respectivo 01:00:5E:xx:xx:xx. O *switch* faz o encaminhamento do pacote baseado na sua tabela *forwarding* do MAC. Os clientes devem fazer o pedido no IP e porta certa.

IGMP v2 (RFC 2236)	IGMP v3 (RFC 3376)	Notes
<b>Joining and Leaving Channels</b>		
<b>JOIN</b> <ul style="list-style-type: none"> <li>• STB issues a Membership Report.</li> <li>• The destination IP address is the multicast group to be joined.</li> </ul>	<b>JOIN</b> <ul style="list-style-type: none"> <li>• STB issues a Membership Report.</li> <li>• The destination address is the “all IGMPv3 routers” address.</li> <li>• The “State Change” Group Record field indicates the group(s) to be joined.</li> </ul>	<ul style="list-style-type: none"> <li>• Existing intermediate devices (DSLAMs/RGs) may not support IGMPv3 since packet format is different</li> <li>• Packet filters based on destination IP address are no longer valid</li> <li>• Deep packet inspection is required to read IGMPv3 Group Records. This implies higher processing and memory requirements. An IGMPv2 device may not be upgradeable to support IGMPv3.</li> <li>• Channel changes should occur faster when using IGMPv3 since a single message contains both join and leave information</li> </ul>
<b>LEAVE</b> <ul style="list-style-type: none"> <li>• STB issues a Leave Group</li> <li>• The destination IP address is the “all IGMPv2 multicast routers” address (224.0.0.2).</li> </ul>	<b>LEAVE</b> <ul style="list-style-type: none"> <li>• A Membership Report is issued by the STB.</li> <li>• The destination address is the “all IGMPv3 multicast routers” address (224.0.0.22).</li> <li>• The “State Change” Group Record field indicates the group(s) to be left.</li> </ul>	
<b>CHANNEL CHANGE</b> <ul style="list-style-type: none"> <li>• Requires two separate messages— Leave Group followed by Membership Report (to join the new channel).</li> </ul>	<b>CHANNEL CHANGE</b> <ul style="list-style-type: none"> <li>• One Membership report includes both Leave and Join information. The IGMP host and router only have to process a single message during each channel change.</li> </ul>	
<b>Special IP addresses</b>		
<ul style="list-style-type: none"> <li>• All IGMPv2 routers: 224.0.0.2</li> <li>• All IGMP hosts (clients): 224.0.0.1</li> </ul>	<ul style="list-style-type: none"> <li>• All IGMP v3 routers: 224.0.0.22</li> <li>• All IGMP hosts (clients): 224.0.0.1 (same as IGMPv2)</li> </ul>	<ul style="list-style-type: none"> <li>• All IGMPv3 implementation are backwards compatible with IGMPv2, so recognize the IGMPv2 “all IGMP routers” address also</li> </ul>

Figura 30 – Diferenças entre IGMPv2 e IGMPv3.<sup>48</sup>

<sup>48</sup> Figura extraída de [33].

## Anexo B

O MANE filtra a informação, através das regras de *firewall iptables*, baseadas em três parâmetros: endereço de origem (IP do servidor), endereço de destino (IP do cliente), protocolo UDP e por porta de destino do vídeo. Não se filtrou no *iptables* por porta de origem, visto que, por defeito, o VLC utiliza uma porta aleatória, embora seja possível utilizar uma porta específica. Nas regras utilizou-se o *target QUEUE*, que permite tratar os pacotes numa aplicação ao nível utilizador, recorrendo ao uso da biblioteca *libipq* (API que tem funções para obter uma cópia do pacote que se encontra no *kernel* e tomar decisões sobre este). No processo utilizador é analisado o conteúdo do pacote de vídeo de acordo com a métrica em análise e tomada uma decisão quanto ao seu destino (descartado ou enviado para o cliente).

No VLC do servidor é garantido que este apenas envia vídeo, através da recodificação sem áudio. O MANE encontra-se no servidor, para simplificar ao máximo o ambiente de teste. Existem duas diferenças quando está no meio da rede:

1. A regra de *firewall* teria que ser diferente, ou seja, se o MANE estivesse no servidor teria de se usar a *chain OUTPUT*, enquanto que se tivesse no meio da rede seria a *chain FORWARD*.
2. Seria necessário garantir que não existiam perdas até os conteúdos chegarem ao MANE e que estes passavam obrigatoriamente por este elemento.



## Anexo C – CWMP

### Especificação CWMP

A explosão que os CPEs tiveram nos últimos tempos, aliada à complexidade da sua configuração para um utilizador comum, constituiu o principal motivo, para o *Broadband Forum* ter desenvolvido a especificação TR-069<sup>49</sup>, intitulada de CWMP, com objectivo de gerir esses mesmos equipamentos.

A comunicação com os CPEs, é feita através de um ACS, permitindo deste modo, a auto-configuração e outras funções de gestão dos CPEs, das quais se destaca:

- ✓ Gestão do *firmware* dos CPEs.
- ✓ Monitorização do estado e desempenho dos CPEs.
- ✓ Diagnósticos.

Como é ilustrado na Figura 31, o ACS encontra-se na rede do operador, integrado com os elementos de gestão do fornecedor de serviços (por exemplo, OSS/BSS). O CWMP pode ser usado para gerir não só os DSL *Broadband Network Termination* (B-NT), mas também outros tipos de CPEs, incluindo *routers*, STBs e dispositivos na rede local do cliente. De salientar que em cada momento, um CPE apenas se encontra num ACS.

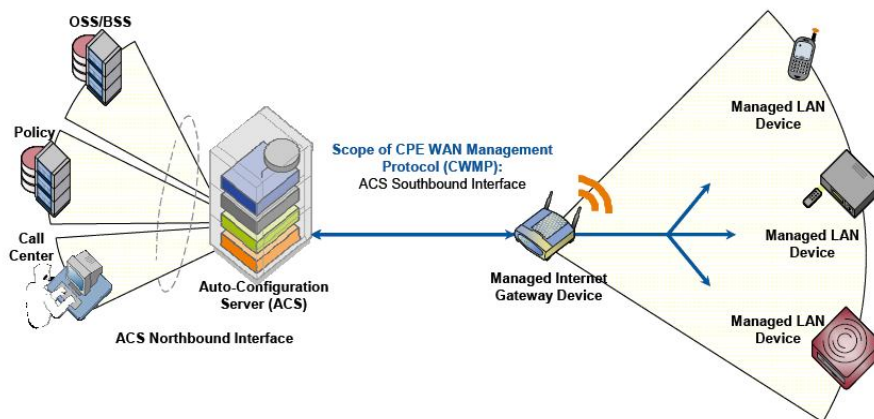


Figura 31 – CWMP e ACS.<sup>50</sup>

O CWMP define um protocolo na camada de aplicação, para gestão remota dos dispositivos dos clientes. A comunicação entre o CPE e o ACS é realizada usando HTTP. As componentes do protocolo CWMP estão ilustradas na Figura 32.

<sup>49</sup> Especificado em [26].

<sup>50</sup> Figura extraída de [26].

CPE/ACS Management Application
RPC Methods
SOAP
HTTP
SSL/TLS
TCP/IP

**Figura 32 – Componentes do CWMP.**<sup>51</sup>

O CWMP usa o protocolo TCP/IP ao nível da camada de rede. Usa também uma camada de segurança para transporte da informação, que fornece confidencialidade e integridade da informação, permitindo inclusive certificados entre CPE e ACS. Utiliza o protocolo HTTP como protocolo de comunicação ao nível da camada de aplicação, utilizando os métodos GET e POST para fazer pedidos. No nível acima, usa *Simple Object Access Protocol* (SOAP), para codificar o RPC, onde os pedidos são enviados em envelopes denominados SOAP *request* e as respostas em SOAP *response*.

O RPC *Methods* especifica um conjunto de métodos que estão definidos pelo CWMP, onde um ACS pode ler e escrever *Parameters*<sup>52</sup>, que permitem configurar e monitorizar o estado do CPE. O protocolo também suporta um mecanismo de auto descoberta que permite um ACS determinar que *Parameters* o CPE suporta. A aplicação usa o CWMP tanto nos CPEs, como no ACS. De salientar que a especificação TR-069 apenas define a comunicação entre o ACS e o CPE, isto é, o nome e a declaração de cada *Parameter* é definida em outros *Technical Reports*. Estes *Parameters* estão organizados em árvores, sendo a raiz o *InternetGatewayDevice*, que está especificado no TR-098.

Relativamente à comunicação, o *Inform* é o evento mais importante enviado pelo CPE. Contém informação específica, em formato TR-069, como por exemplo: fabricante, endereço IP e MAC, versão do *firmware*, entre outros. A informação do *Inform* é essencial para identificar o dispositivo no ACS. Existem três eventos que podem levar ao envio de um *Inform* do CPE para o ACS:

- ✓ *Value Change*, por exemplo, mudança de IP.
- ✓ *Connection Request*, ACS pede para ser comunicado.
- ✓ *Periodic Inform*, periodicamente são enviados *Inform*s.

É definido um mecanismo que permite um CPE informar o correspondente ACS de várias condições e assegurar que a comunicação entre o CPE e o ACS, se realiza com o mínimo de frequência. Quando a comunicação é estabelecida, o CPE é identificado pelo número de série e pelo fabricante. Quando a sessão é iniciada pelo CPE, este envia um *Inform request* (HTTP POST) para o ACS, isto indica o estado actual do CPE e que este se encontra pronto para aceitar pedidos do ACS. A sessão é iniciada com sucesso se for recebido um *Inform response* do ACS.

<sup>51</sup> Figura extraída de [26].

<sup>52</sup> Estão descritos em [26].

Também é possível, o ACS informar assincronamente de uma mudança na configuração do CPE, sendo um aspecto importante da auto-configuração dos CPEs. Isto permite que o mecanismo de auto-configuração possa ser usado para serviços, que requeiram reconfiguração quase em tempo real do CPE. Por exemplo, fornecer ao cliente imediato acesso ao serviço que o cliente subscreveu, sem ter de esperar pelo próximo contacto periódico. O CWMP incorpora para o efeito, um mecanismo que permite o ACS fazer *Connection Request*, em qualquer momento para o CPE, respondendo este com um *Inform Request*. Quando o ACS recebe um *Inform request* do CPE, se o ACS pretender iniciar a comunicação manda um *Inform response*. O *Connection Request*, não é mais do que o envio de um HTTP GET com o URL do CPE (recebido num *Inform* anteriormente).

Os motivos para que uma sessão em normal funcionamento (excluindo problemas) termine, são:

- ✓ O CPE/ACS receber um HTTP *response*/POST vazio do ACS/CPE, que significa que o ACS/CPE não tem mais pedidos para enviar para o CPE/ACS.
- ✓ O CPE/ACS recebeu todos os pedidos pendentes do ACS/CPE; o CPE/ACS enviou todas as respostas pendentes para o ACS/CPE.
- ✓ O CPE/ACS não tem mais pedidos para enviar para o ACS/CPE e o CPE/ACS emitiu um HTTP POST/Response para o ACS/CPE vazio.
- ✓ O CPE deve considerar terminada uma sessão se não receber um *Inform response* no intervalo de tempo estipulado.

Na Figura 33, está ilustrada a comunicação TR-069 entre o ACS e o CPE.

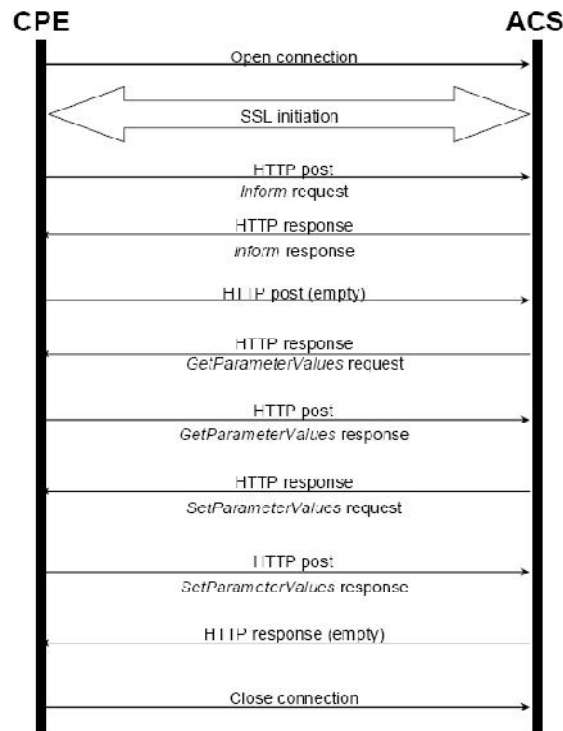


Figura 33 – Exemplo de uma sessão entre ACS e um CPE.

### **Axess – Plataforma de gestão de CPEs**

A solução de gestão Axess foi desenvolvida pela Axiros. Sendo a especificação TR-069 totalmente suportada.

A plataforma, que está ilustrada na Figura 34, efectua a gestão de *firmware* dos CPEs, incluindo actualizações automáticas. A solução permite informar o *software* OSS existente no fornecedor de serviços do estado da configuração dos CPEs; procede ainda à sua integração com repositórios que gerem a informação dos clientes, como por exemplo o *Remote Authentication Dial In User Service* (RADIUS). A plataforma é também capaz de gerir CPEs que não suportem TR-069, usando para tal, um *proxy* que “traduz” para a comunicação suportada pelo CPE, por exemplo, recorrendo ao *Secure Shell* (SSH), de modo a obter informação deste. No Axess pode ser desenvolvido um portal, de modo a criar mais valor acrescentado para os clientes, onde ao invés de o cliente configurar localmente o seu CPE, pode configurar no Axess, ficando a configuração armazenada no servidor.

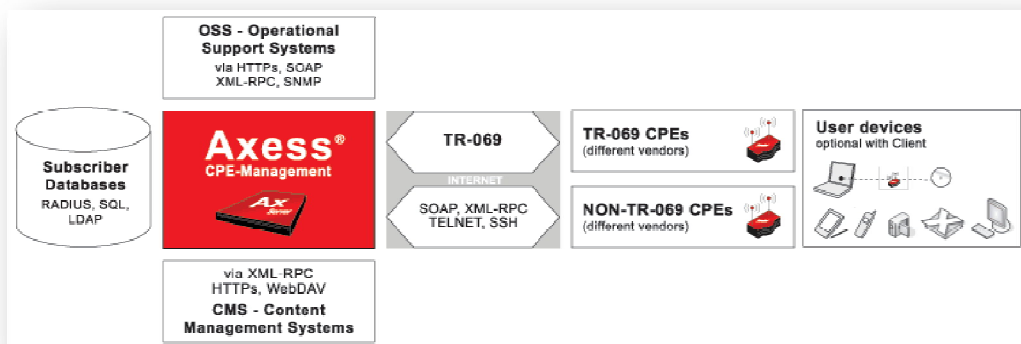


Figura 34 – Plataforma Axess.<sup>53</sup>

O Axess permite a integração da camada de adaptação do CPE na plataforma do servidor, resultando na possibilidade de gestão de qualquer CPE, de uma maneira integrada e flexível, sem pilhas de protocolos fixas no CPE, recorrendo a conjuntos de *Parameters* dinâmicos, extensíveis em qualquer instante sem necessitar das actualizações de *firmware*. Suporta todas as funcionalidades TR-069, sem comprometer a segurança.

No Axess existe um processo designado de *Axess Configurator*, responsável por fazer o *Connection Request*. Este informa o *Axess Server* de falhas de comunicação, caso não consiga comunicar com o CPE.

O Axess utiliza uma base de dados remota. Isto permite num ambiente real ter diversos Axess em *cluster* a aceder a uma base de dados. Um nó Axess está pronto para associar mais de cem TR-069 interações/seg. Na camada de segurança apenas foi implementada no Axess, a autenticação do CPE para o ACS (o CPE envia *username* e *password*), contudo a comunicação do ACS para os CPEs, isto é, os *Connection Requests* não são autenticados. Contudo isto não é problemático, por se considerar que o ambiente do fornecedor é controlado.

<sup>53</sup> Figura extraída de [34].

## Anexo D – Estrutura da unidade NAL do H.264/SVC

Na Figura 35, está presente a estrutura de uma unidade NAL, sendo que em H.264/AVC apenas existe o primeiro byte. Nesta estrutura convém referir o campo NRI que especifica o tipo de unidade NAL, assim como os campos: o Did, o TID e o QID, que se encontram presentes no H.264/SVC.

0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
F NRI		NUT						R I	PID						N	Did			QID				TID			U D O R2					

Figura 35 – Estrutura da unidade NAL em H.264/SVC<sup>54</sup>

O TID representa a escalabilidade temporal, sendo que, para um qualquer valor  $k$ , um fluxo de vídeo obtido pela remoção de todas as unidades de acesso<sup>55</sup> das camadas temporais superiores a  $k$ , continua a ser válido.

A escalabilidade espacial é obtida pelo Did, representado por S na Figura 4. Quando existe um aumento da resolução espacial de uma camada para outra, o Did é incrementado. As camadas mais baixas das imagens não necessitam de estar presentes em todas as unidades de acesso, sendo desta forma possível combinar escalabilidade temporal e espacial. Nas normas anteriores ao H.264/SVC, a predição era apenas possível dentro de uma dada camada e implicitamente da camada abaixo mais próxima na hierarquia. No H.264/SVC e com o uso do campo Did, é permitido uma predição entre camadas, podendo recorrer à utilização de todas as camadas com Did inferior ao da camada a ser predita, inclusive a camada base. A vantagem está na eficiência, pela possibilidade de obter o máximo de informação das camadas abaixo.

A escalabilidade na qualidade, pode ser considerada como um caso especial da escalabilidade espacial, sendo que o tamanho da imagem não é alterado, designando-se por *Coarse Grain Scalability* (CGS), ou seja, é quando entre duas camadas com Dids diferentes, a resolução espacial é a mesma. Assim é permitido o descarte de todas as unidades NAL com determinado Did, eliminando toda uma camada. Apenas se pode mudar entre camadas Did (inclui CGS) em pontos específicos (unidades de acesso IDR<sup>56</sup>) do fluxo binário. Fazendo a analogia com a Figura 4, usando a camada um com uma resolução espacial, por exemplo, de QCIF, ao se subir para a camada dois, esta pode não constituir um aumento espacial, mas o Did aumentou, então diz-se que é uma camada CGS.

Contudo para aumentar a flexibilidade na adaptação dos conteúdos e para melhorar a eficiência na codificação dos fluxos de vídeo com diversos ritmos binários, foi introduzido o *Medium Grain*

<sup>54</sup> Figura adaptada de [7].

<sup>55</sup> Uma unidade de acesso é um conjunto de unidades NAL, com determinadas propriedades e que decodificadas formam exactamente uma imagem.

<sup>56</sup> Significa que a unidade de acesso corrente e as seguintes são independentes das unidades de acesso já transmitidas, ou seja, é inteiramente constituída por tramas I.

*Scalability* (MGS). Este termo está associado ao QID, tendo a possibilidade de se alterar em cada unidade de acesso, por exemplo, descartando as unidades NAL com um QID superior a um determinado valor. Desta forma, a utilização do MGS permite refinamentos na qualidade e consequente “afinação” no débito. Fazendo a analogia com a Figura 4, é representado por B, ou seja, SNR.

As unidades de acesso MGS podem ser livremente descartadas sem afectar a conformidade dos resultados do fluxo binário, ao invés o CGS todas as unidades de acesso de uma camada inteira ou são processadas ou não.