Dissertação

Mestrado em Computação Móvel

Arquitecturas IPTV

Paulo Jorge Mendes Cardoso Neto

Leiria, 30 de Dezembro 2010

Dissertação

Mestrado em Computação Móvel

Arquitecturas IPTV

Paulo Jorge Mendes Cardoso Neto

Dissertação de Mestrado realizada sob a orientação do Prof. Dr. Carlos Manuel da Silva Rabadão da Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria.

Leiria, 30 de Dezembro 2010

Dedicatória

À minha esposa Edite e à minha filha Sara

Agradecimentos

Quero aqui expressar os meus sinceros agradecimentos às pessoas que directa e indirectamente, contribuíram para a realização deste trabalho.

Começo por agradecer ao Professor Doutor Carlos Rabadão pela sua enorme disponibilidade, pela sabedoria dos seus conselhos de orientação de extrema importância ao longo das diversas etapas deste trabalho, sem os quais não teria sido possível concluí-lo com sucesso.

Agradeço ao Professor Doutor António Pereira pelo incentivo na escolha deste tema.

Ao Renato Martins obrigado pela disponibilidade.

Agradeço também ao Instituto Politécnico de Leiria e à Escola Superior de Tecnologia e Gestão de Leiria, pelas condições que colocaram ao meu dispor para a realização deste trabalho.

O meu sincero obrigado aos meus amigos Paulo, Milene, Balsas e Cátia bem como aos meus pais e familiares pelo apoio que me prestaram.

Por fim, quero agradecer em especial, à minha esposa Edite, por todo o apoio que me ofereceu.

Resumo

Os operadores de telecomunicações de todo o Mundo estão a investir em novas formas de fornecer aos seus clientes serviços de TV + Voz + *Internet*, em redes IP unificadas, que proporcionam novas oportunidades de negócio, e serviços inovadores, onde o IPTV se encaixa na perfeição.

Esta dissertação propõe um estudo na área de IPTV, com especial atenção nas arquitecturas de rede, *codecs*, protocolos, QoS/QoE, e redes *wireless*. O serviço de IPTV é extremamente sensível a perdas e atrasos de pacotes nas redes de cabo, mas nas redes *wireless* existem outras condicionantes que degradam o serviço ainda mais.

Baseado neste estudo é testado o comportamento de *live stream multicast* (*Live* IPTV), nas redes *wireless*, que apresentam várias limitações quando operam com IP *multicast.* Os testes são realizados numa rede laboratorial, que é uma réplica dum segmento existente na rede wireless da freguesia da Memória, concelho de Leiria. São realizados testes com o mesmo vídeo codificado nas qualididades SD, HD e FHD.

Palavras-Chave:

IPTV, *multicast*, *wireless*, *streaming*, *codecs*, QoE, QoS

Abstract

The World telecommunication companies are investing in ways to offer TV + Voice + Internet, in unified IP networks, in order to provide new business opportunities and innovative services, where the IPTV perfectly fits.

This dissertation propose a study over the World of IPTV, focused in network architectures, *codecs*, protocols, QoS/QoE, and wireless networks. IPTV service is extremely sensitive to packets loss and jitter in wired networks, but in wireless networks other constraints can spoil the IPTV service even more.

Based on this study, the live *stream* multicast behavior (Live IPTV), is tested in a wireless network. These networks have lots of limitations when working with IP multicast. These tests are preformed in a laboratory network, which is a replica of the existent shortest segment, in the wireless network located at Memoria parish council, Leiria. All tests are preformed with the same movie which is coded in SD, HD and FHD qualities.

Keywords:

IPTV, *multicast*, wireless, streaming, *codecs*, QoE, QoS

Índice de figuras

Índice de tabelas

Siglas

AC	Access Category
ACK	Acknowledgement
AIFS	Arbitration Inter Frame Space
AP	Access Point
ATIS	Alliance for Telecommunication Industry Solutions
AVC	Advanced Video Coding
CAM	Conditional Access Module
CAS	Conditional Acess System
CBR	Constant Bit Rate
CDN	Content Distribution Network
CFP	Contention Free Period
CNAME	Canonical Name
CRC	Cyclic Redundancy Code
CSMA/CA	Carrier Sense Multiple Access with Colision Avoidance
CTS	Clear to Send
CW	Contention Window
DCF	Distributed Coordination Function
DCCP	Datagram Congestion Control Protocol
DCT	Discrete Cosine Transform
DHCP	Dinamic Host Configuration Protocol
DiffServ	Differentiated Services Framework
DIFS	Distributed Inter Frame Space
DLNA	Digital Living Network Alliance
DNS	Domain Name System
DPCM	Differencial Pulse Code Modulation
DR	Designated Router
DRM	Digital Rights Management
DS	Distribution System
DSCP	Defferentiated Service Code Point
DTIM	Delivery Traffic Indication Message
DVB	Digital Video Broadcast
DVRMP	Distance Vector Multicast Routing Protocol
EDCA	Distributed Coordination Function

ESTG	Escola Superior de Tecnologia e Gestão de Leiria
ETSI	European Telecommunications Standards Institute
ETSI-TISPAN	Telecommunications and Internet converged Services and Protocols for Advanced Networking
FHD	Full High Definition
FLV	Flash Video
FPS	Frames por segundo
GOP	Group of Pictures
GSI	Global Standards Initiative
GUI	Graphical User *Interface*
HCCA	Hybrid Coordination Channel Access
HD	High Definition
HVS	Humana Vision System
IEEE	Institute of Electrical and Electronics Engineers
IGMP	Interior Gateway Multicast Protocol
IIF	Interoprability Forum
IMS	IP Multimedia Subsystem
IntServ	Integrated Services Achitecture
IP	Internet Protocol
IPTV	Internet Protocol Television
ISP	Internet Service Provider
ITU-T	International Telecomunication Union
MAC	Medium Access Control
MBGP	Multiprotocol Bonduary Gateway Protocol
MOSPF	Multicast Open Shortest Path First
MRIB	Multicast Routing Information Base
MSDU	MAC Service Data Unit
MSN	Microsoft Network
NAT	Network Address Translation
NAV	Network Allocation Vector
NGN	Next Generation Networks
OTT	Over the Top
PCF	Point Coordination Function
PCM	Pulse Code Modulation
PHY	Physical Layer
PIM	Protocol Independent Multicast
PVR	Personal Video Recorder

QoE	Quality of Experience
QoS	Quality of Service
RGB	Red, Green, Blue
RP	Rendezvous Point
RPF	Reverse Path Forwarding
RPT	Rendezvous Point Trees
RR	Receiver Report
RSVP	Resource Reservation Protocol
RTCP	Real Time Transport Control Protocol
RTP	Real Time Transport Protocol
RTS	Request to Send
RTSP	Real Time Streaming Protocol
SD	Standard Definition
SDES	Source Description Items
SPT	Shortest-path trees
SR	Sender Report
SSH	Secure Shell
STB	Set-top box
TDMA	Time Division Multiple Access
TIB	Tree Information Base
ToS	Type of Service
TXOP	Transmition Opportunity
VBR	Variable Bit Rate
VI	Text Editor
VLC	Video Lan Client
VOIP	Voice over IP
VOD	Video-on-Demand
WMM	Wi-Fi Multimedia
WMV	Windows Media Video

Índice

1. Introdução

Nos dias de hoje assiste-se a um crescimento do número de utilizadores de IPTV[1] e VoIP[2], sendo a tecnologia IP o ponto de convergência das actuais tecnologias. Surgem novas oportunidades de negócio, onde integração, interactividade e personalização são as palavras de ordem, (Hjelm, 2008). Estudos demonstram que os clientes preferem pagar uma só factura que contenha um leque abrangente de serviços como o *triple-play* (Siemens Communications and Juniper Networks)

O conceito de *triple-play* (Internet, televisão e telefone) existente em muitos operadores é oferecido pela mesma rede de acesso, mas não é integrado, isto é, os serviços são acedidos através de equipamentos distintos sem qualquer interacção entre os mesmos. Com a convergência para IP os serviços oferecidos ao cliente são mais atractivos e interactivos, o que cria novas oportunidades de negócio, mas aumenta também as preocupações de garantia de serviço por parte dos fornecedores, porque o sucesso do leque de serviços oferecidos depende da satisfação, e das expectativas do cliente, (Held, 2006).

IPTV é uma sigla que no seu conceito base, descreve um sistema onde a TV digital é entregue ao cliente através do IP (*Internet Protocol*), através de uma rede de banda larga pública, ou numa rede local, (J.Walko, Dec. 2005), na qual os conteúdos multimédia são codificados, e encapsulados em pacotes IP para de seguida serem distribuídos pela rede. A codificação pode ser feita através de diversos *codecs*, tais como, MPEG-2, MPEG-4, H.264, WMV, Divx, e Xvid. Devem ser definidas políticas de QoS que garantam uma boa qualidade de serviço de video ao utilizador. Isto é, vídeo sem atrasos, arrastamentos ou erros, que de outra forma acontecem, devido à existência de outros fluxos de pacotes de

[1] IPTV - *Internet Protocol Television*
[2] VoIP – Voice over IP

dados, e VoIP que limitam a largura de banda disponível para este tipo de serviço. O grande problema do QoS é que as suas requisições podem não ser reconhecidas quando os pacotes circulam entre diversas redes, no entanto, se os pacotes circulam dentro da mesma rede é possível oferecer QoS.

O *multicast* é usado no IPTV para que as *streams* de TV sejam entregues aos vários receptores em simultâneo sem sobrecarregar a rede. Num sistema de IPTV tem de existir o protocolo IGMP *(Interior Gateway Multicast protocol)*, e o protocolo de encaminhamento *multicast*. O IGMP é usado pelo receptor para fazer o pedido de acesso (*Join*) a um grupo *multicast*, o protocolo *multicast* é usado para replicar as *streams* de pacotes *multicast*. Sempre que o receptor troca de canal são usadas mensagens IGMP para libertar os recursos, que a *stream* visualizada estava a ocupar na rede (*Leave*), e de seguida é feito um *Join* ao novo grupo *multicast* para a visualização do novo canal, (Chunglae Cho, 2007).

1.1 Motivação e Objectivos

Cada vez mais são usados sistemas de IPTV na hotelaria, nos hospitais, e também nos operadores de telecomunicações (ex: *MEO, CLIX*). Estes sistemas oferecem um vasto leque de serviços ao utilizador final, com um serviço quase sempre oferecido por rede de cablada, e não em infra-estruturas *wireless*.

Pretende-se com este trabalho investigar, e estudar o comportamento da IPTV nas redes *Wi-fi.* Nomeadamente, o comportamento das *streams* de vídeo através destas redes, que são alvo dum enorme número de perturbações por trabalharem num meio partilhado, sujeitos a todo o tipo de interferências que é o ar.

De forma a atingir este objectivo principal, acima referido, é realizada uma análise mais detalhada sobre o problema, levando a estabelecer os seguintes objectivos:

i. Levantamento de referências bibliográficas sobre IPTV, e estudo das mesmas.

ii. Estado da Arte, pesquisa sobre as diferentes arquitecturas usadas no IPTV, *codecs*, protocolos, e componentes.

iii. Estudo do comportamento do IPTV em redes *wireless*. Tópicos a abordar: requisitos duma topologia de rede de dados para suportar IPTV, QoS (mecanismos), comportamento dos *codecs*, e propor um cenário de teste para uma rede *wireless*.

iv. Implementar o cenário, fazer testes, recolher os dados, e analisá-los.

1.2 Metodologia de Investigação

A metodologia de investigação adoptada para atingir os principais objectivos deste trabalho compreende várias fases, evidenciadas pela lista de objectivos supracitados.

Numa primeira fase, tendo em conta as principais áreas de investigação subjacentes a este trabalho, realiza-se o levantamento de bibliografia referente a desenvolvimentos identificados como relevantes, em cada uma dessas áreas. Do estudo desta bibliografia, resulta uma avaliação do conhecimento científico em cada uma dessas áreas, fundamental para as tomadas de decisão durante a fase de concepção, e caracterização da proposta.

Numa segunda fase, sustentada no conhecimento científico anteriormente adquirido, é concebido e caracterizado um protótipo para o teste do comportamento das *streams* de IPTV nas redes *wireless*. Para validar a nova proposta é implementado um protótipo da arquitectura, procede-se à análise, e avaliação dos seus resultados de desempenho, usabilidade, e simplicidade.

1.3 Organização da Dissertação

Esta dissertação está organizada em oito capítulos, que reflectem o trabalho desenvolvido para atingir os objectivos anteriormente apresentados. A divisão dos temas por capítulos permite a compreenção das várias etapas até chegar à fase final de implementação.

No presente capítulo são apresentadas as motivações e objectivos desta tese, bem como, a metedologia de investigação usada.

No capítulo 2 são clarificados os conceitos necessários à compreensão do objecto de estudo desta dissertação, com uma breve introdução ao conceito de IPTV, protocolos de rede usados, normas de codificação de vídeo (*CoDec*), serviços fornecidos pelo IPTV e QoS. Alguns dos temas mencionados neste capítulo são detalhados nos capítulos 3, 4 e 5.

No capítulo 3 são descritos os protocolos de rede necessários para uma rede de IPTV, nomeadamente o IGMP, PIM, RTP, RTCP e RTSP.

No capítulo 4 são descritos os *codecs* mais usados nos sistemas de IPTV. É feita uma introdução sobre o sistema visual do ser humano para melhor entender o funcionamento dos *codecs.*

No capítulo 5 são definidos o QoS, os mecanismos de QoS, o QoE, e o *adaptative streaming*.

O capítulo 6 descreve a arquitectura da solução, desenhada com base em todo o conhecimento adquirido na investigação. São abordados os requisitos que uma rede local, com distribuição *wireless*, deve cumprir para suportar IPTV *Live*, e as limitações que os APs apresentam com o *multicast*.

O capítulo 7 destina-se à implementação do cenário proposto no capítulo anterior. A implementação passa por três fases, sendo a primeira a configuração do servidor de IPTV, a segunda a configuração do cenário, testes de

conectividade, e medições dos *links*. Por fim, a terceira consiste na configuração do *multicast* na rede, e os respectivos testes.

A conclusão da dissertação é efectuada no Capítulo 8, com a realização dum resumo sobre o trabalho de investigação, e perceber a dimensão de alcance dos objectivos inicialmente propostos. São também indicadas algumas considerações relativas a trabalho futuro, baseadas na análise do trabalho desenvolvido.

2. Estado da Arte

Este capítulo é apenas introdutório, mas de extrema importância. Pois clarifica os conceitos necessários à compreensão do objecto de estudo desta dissertação.

O primeiro tópico consiste numa breve introdução ao conceito de IPTV. Nos tópicos seguintes são descritas algumas das arquitecturas de IPTV mais importantes: protocolos de rede usados, normas de codificação de vídeo (*CoDec*), serviços fornecidos pelo IPTV e QoS.

2.1 Conceito de IPTV

Hoje em dia, os ISP[3] estão a expandir as suas ofertas através de pacotes *triple-play*[4] e *quadruple-play*[5], que proporcionam novas oportunidades de negócio.

Apesar dos serviços serem fornecidos pela mesma infra-estrutura física, ou seja, pela mesma rede de acesso, cada serviço é autónomo, e acedido através de equipamentos distintos sem qualquer interacção entre os mesmos.

A tendência de migrar todos os serviços para IP[6] é cada vez mais forte. Se tivermos todos os serviços em IP, pode ser fornecido um leque abrangente de ofertas, tais como: *e-mail*, telefone, tv, domótica, alarmística, vídeo-conferência, e todo um conjunto de serviços personalizados para o utilizador final.

A IPTV e o VoIP são a prova viva dessa tendência, são duas tecnologias recentes, que permitem que a tv e a voz circulem nas redes de dados IP. Para que tal seja possível, são despendidos enormes esforços em investigação.

A grande vantagem dos sistemas de IPTV reside na ampla capacidade de interacção com o utilizador, pois assentam no tipo de comunicação bidireccional.

[3] *Internet Service Provider*
[4] Pacote de serviços com TV + Telefone + Internet
[5] Pacote de serviços com TV + Telefone + Internet + Serviço Móvel
[6] *Internet Protocol*

Enquanto nos sistemas actuais existe *broadcast*, isto é, todos os utilizadores recebem o mesmo conteúdo, com IPTV, usa-se *multicast*, para que que não seja criada uma *stream* por utilizador no equipamento que está a disponibilizar a mesma. Outra das grandes vantagens da IPTV é a de cada utilizador receber apenas os canais que está a visualizar, em vez de ter a capacidade da sua rede de acesso esgotada com todos os conteúdos disponíveis. Desta forma, existe um maior aproveitamento dos recursos de rede, permitindo assim a existência de novos, e inovadores serviços.

Outra vantagem é permitir ao utilizador final a possibilidade de interagir com o que se está a passar no conteúdo visualizado. O conceito de TV interactiva não é novo, as primeiras experiências datam de 1990, só que não tem o sucesso esperado. A interacção é feita por votação telefónica, e permite que os acontecimentos dum filme sejam adaptados à audiência de acordo com os de votos efectuados pelos telespectadores. É óbvio, que é necessário o produtor do filme prever diversos desfechos alternativos, para responder à votação dos telespectadores, certo é, que mesmo os telespectadores que estam em minoria vêm o desfecho que o grupo com mais votações escolhe, pois na televisão convencional, um filme é linear.

Hoje em dia, nos sistemas de IPTV, o significado de TV interactiva vai muito mais além do que na experiência anterior, pois existe a possibilidade de ser escolhido o desfecho relativamente à selecção do utilizador, como se duma navegação *Web* se tratasse. É também possível obter estatísticas actualizadas sobre as provas dos jogos olímpicos que estão a ser emitidos em directo, ou mesmo a informação meteorológica, a evolução da bolsa de valores, atender uma chamada de vídeo-conferência, e memorizar todas as opções personalizadas no perfil do utilizador, que pode ser carregado automaticamente quando a câmara da TV detectar a face do utilizador – tudo isto através dum televisor.

A maturidade das redes IP que assentam na tecnologia *Ethernet* totalmente difundida, vulgarizada devido ao seu baixo custo de implementação, débitos de informação elevados, permite que sejam desenvolvidas novas aplicações e serviços orientados para o utilizador.

As mais-valias dos sistemas de IPTV podem traduzir-se na seguinte Figura 1:

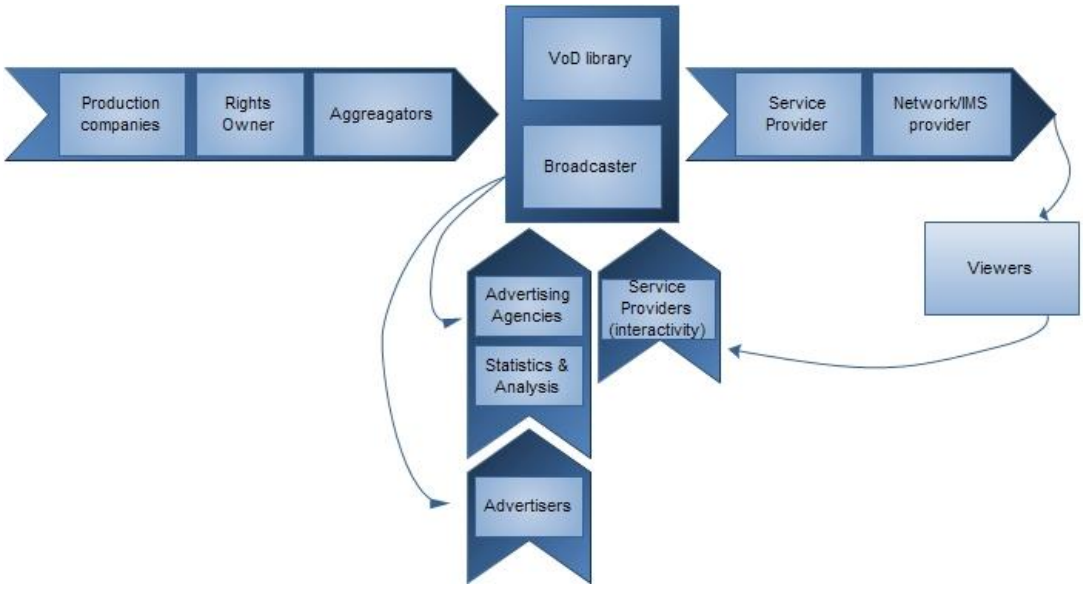

Figura 1 - *The Value Chain* (Hjelm, 2008)

A figura representa o encadeamento existente entre as várias entidades num sistema de IPTV, desde a produção do conteúdo até o telespectador. É possível constatar que toda a cadeia depende dos telespectadores. Os programas televisivos são produzidos com o objectivo de satisfazer o telespectador. Para tal, existem as agências de análise, e estatística que avaliam o impacto dos programas na audiência. Com base nestes dados, as agências de publicidade estudam a melhor forma de injectar a publicidade dos produtos, orientados para o telespectador nos intervalos dos programas com mais audiência. Mas também, para as produtoras criarem conteúdos adaptados à audiência.

As arquitecturas de IPTV devem conseguir satisfazer este tipo de *"ecossistema"* para serem implementados na prática. (Hjelm, 2008)

No entanto, tanto a IPTV como o VoIP não se comportam bem se existirem congestionamentos na rede, que infelizmente acontecem, e são uma das principais barreiras à expansão em massa destas tecnologias. Para resolver estes problemas, é necessário oferecer serviços convergentes com menor custo, e melhorar as redes de acesso. Grupos de telecomunicações de todo o mundo estão a investir grandes esforços para actualizar as suas infra-estruturas para as NGN[7] baseadas em IP.

O termo NGN não é uma tecnologia, mas sim um conceito de redes multi-serviços capaz de transportar voz, dados e vídeo. Para o efeito, este tipo de redes tem a camada de controlo/sinalização separada da camada de transporte/comutação. O transporte é feito em pacotes que podem conter todo o tipo de informação. O QoS é um dos principais pilares nestas redes, permitindo assim que os diferentes tipos de tráfego fluam de acordo com as suas necessidades.

2.2 Arquitecturas de IPTV

Neste momento existem várias maneiras de visualizar televisão com o protocolo IP, algumas delas, consistem na distribuição dos conteúdos através da rede pública (*Internet* TV), e outras através das redes privadas (IPTV).

Tanto nas redes públicas como nas privadas, podemos ter VoD[8] ou *streaming* (TV em directo).

Na internet, os primeiros casos de IPTV com sucesso, surgem em 1999. O conceito basea-se no VoD, devido à velocidade das ligações à internet que existem na altura, que impedem a transmissão duma *stream* de vídeo em tempo

[7] *Next-Generation-Networks*
[8] *Video-on-Demand*

real com qualidade aceitável. Ainda não existem tecnologias capazes de efectuar *adapative streaming*. Então a solução consiste no *download* prévio do conteúdo para o computador, para depois ser visualizado. O conceito de *progressive download* só surgiu anos mais tarde. Os filmes alugados são vistos através duma aplicação própria, que no fim de expirado o tempo do aluguer, é eliminado do computador.

Os dois maiores fornecedores de conteúdos de VoD, com a filosofia *download-and-play* através da *internet*, foram a *CinemaNow* e a *MovieLink*. (Held, 2006)

A *CinemaNow* representa um dos maiores fornecedores no que toca a distribuição de conteúdos de IPTV pela internet, fundado em 1999, actualmente tem uma biblioteca com mais de 7000 títulos desde programas de televisão, concertos de música e filmes. Entre os principais accionistas encontramos nomes como *Microsoft, Lions Gate Entertainment, Cisco Systems* e *Blockbuster*. Em 2005, a *CinemaNow* dá entrada na distribuição de conteúdos em HD[9]. Estes conteúdos são visualizados em duas modalidades, *download-to-own*, no qual se paga para poder ver o conteúdo para sempre, ou *Pay-Per-View*, que consiste no aluguer do conteúdo por um período de 24 horas.

A seguir ao *CinemaNow*, temos a *MovieLink* com a mesma filosofia do anterior, à excepção do *download-to-own*. Fundada em 2005 pelos cinco maiores estúdios*: MGM, Paramount, Sony Pictures, Universal* e *Warner Brothers*.

Neste momento, temos outros fornecedores de VoD, nos quais, se paga uma mensalidade, para ter um conjunto de filmes que se podem visualizar durante esse mês. Um deles é a *Netflix*, que oferece uma STB, conectando-se à internet e à TV, permitindo a navegação no portal da Netflix para alugar os filmes. O filme é transferido por "*progressive download*", permitindo seleccionar o filme, e começar imediatamente a visualizá-lo.

[9] *High Definition*

Ao nível do *streaming* temos alguns exemplos de fornecedores no nosso país, tais como o *Meo* e a *Clix*, que fazem streaming através da ligação à internet do utilizador, não sendo possível visualizar no PC, um canal disponível no nosso pacote de TV. Isto porque, o tráfego de tv, voz, e dados, apesar de já ser IP, vêm separados em VLANs diferentes de modo a garantir QoS através de classes de serviço, que são atribuídas com base nas VLANs, e também para garantir alguma segurança nos acessos. No entanto, este cenário ainda não se pode definir como uma rede unificada, porque não é possível a um utilizador usufruir do pacote de canais de tv no seu PC.

Existem também alguns sistemas de IPTV instalados nas LANs de hotéis, hospitais e câmaras municipais. São sistemas independentes, que fornecem serviços de *streaming*, VoD e Internet, nos quais alguns conteúdos são gratuitos e outros pagos.

Estes sistemas privados assentam numa arquitectura comum, exemplificada na Figura 2:

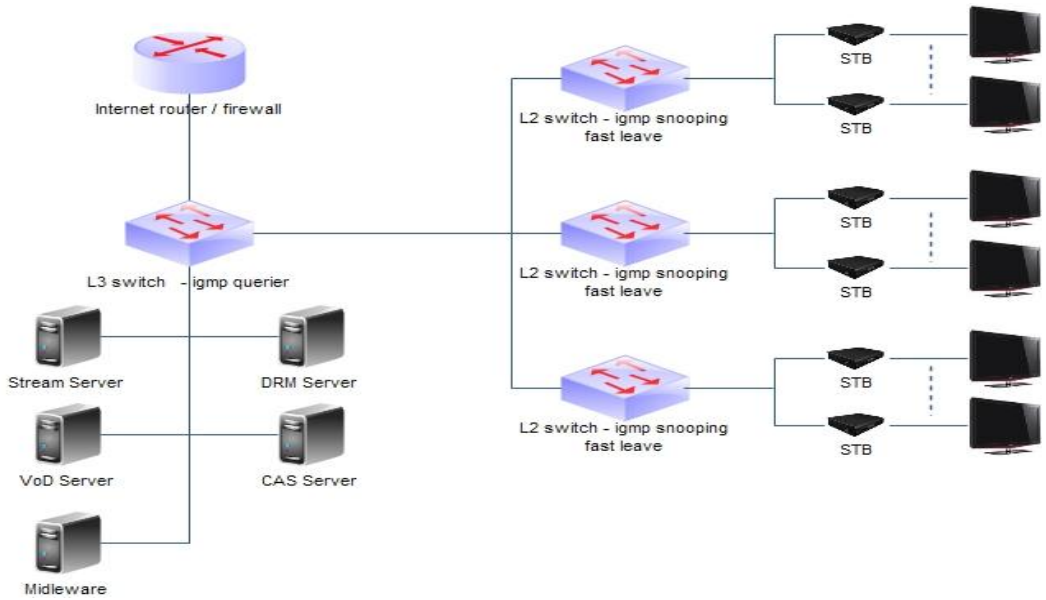

Figura 2 - *Head End* - IPTV

Um *head-end* IPTV é a zona onde se encontram todos os componentes que executam as funções do sistema, é aqui que os canais de DVB-S, DVB-C ou DVB-T

são recebidos, descodificados caso seja necessário, tratados posteriormente, formatados, e codificados para serem distribuídos na rede IP.

Stream Server

Neste equipamento, os canais DVB-S, DVB-C ou DVB-T são recebidos e descodificados através da CAM + *SmartCard*. Para cada canal, são definidas as *streams* de áudio e vídeo elementares, que são usadas para gerar o MPEG-TS (MPEG-*transport stream*), e de seguida é encapsulado num endereço IP *multicast* com o respectivo porto.

Sempre que é necessário descodificar canais, é importante ter em conta o número de PIDs, que a CAM[10] + *SmartCard* conseguem descodificar em simultâneo. Existem dois tipos de CAMs, as domésticas que tipicamente descodificam 4 PIDs, e as profissionais que descodificam 12 PIDs. Cada canal, consome no mínimo 2 PIDs (1 PID Áudio + 1 PID Vídeo)

VoD *server*

Este equipamento é responsável por armazenar os conteúdos de vídeo e áudio para serem vistos sempre que existir um pedido. Para suportar *trick modes*, (*Play, FastForward, FastRewind, Pause*), o VoD *Server* cria um índice de cada filme baseado nas *frames* I do ficheiro de vídeo. Este pode ter a função de *ingestion*, que consiste na gravação de um ou mais canais *live* durante um período de tempo, para que o *midleware* comute entre a emissão *live* e a respectiva gravação. Esta comutação é necessária para fornecer o serviço de *"pause TV"* ou mesmo, a função de começar a ver um programa do início, mesmo que já tenha começado anteriormente.

[10] CAM – *Conditional Access Module*

CAS

Este equipamento é responsável por codificar as *streams IP*, controlar os acessos às mesmas em conjunto com o *midleware*, isto é, a *set-top box* só consegue abrir a *stream* se o *midleware* autorizar.

DRM

Este sistema pode ser integrado com o VoD *Server*, para permitir acesso aos conteúdos VoD que estão codificados neste sistema nativamente. Normalmente, estes sistemas também fazem controlo do número de visualizações de cada filme, para posteriormente serem cobrados os respectivos direitos com base no número de visualizações. Mais uma vez, este sistema está interligado com o *midleware*.

Set-top box

Este equipamento é um pequeno computador com capacidade para interpretar *html*, *javascript*, *flash* (limitado), ou mesmo *silverlight*, que fica do lado do cliente, vai interagir com o *midleware* e com a televisão. A STB só conhece a ligação ao *midleware*, tudo o que sejam menus, listas de canais, jogos, *widgets*, VoD, entre outros, provem do *midleware* que valida a STB, e lhe dá permissões para carregar as listas de conteúdos com base no seu perfil.

Midleware

Este equipamento é o pilar central de todo o sistema de IPTV, pois é responsável por comandar todos os equipamentos de IPTV do *head-end*, e do cliente, tais como: VoD *Servers*, *Stream Servers*, CAS, DRM. É neste que se encontra o *layout* que as STB vão carregar. O *Layout* pode ser definido com base no perfil de cada grupo de utilizadores.

Na arquitectura cliente / servidor, o cliente é a STB, e o servidor é o *midlware* que implementa as camadas de serviços de identificação, autorização, directórios, certificados digitais, entre outros.

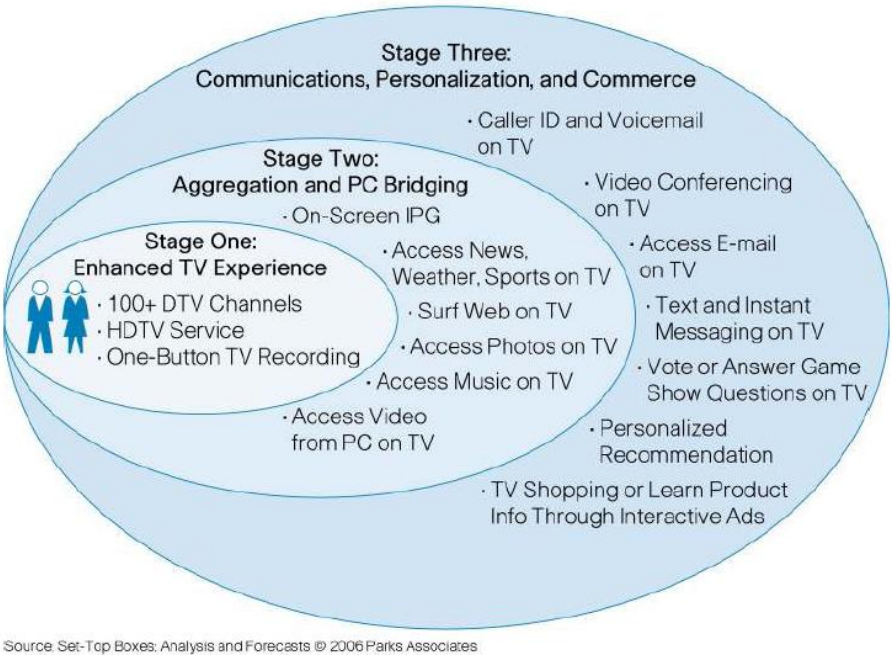

Figura 3 - Cenário Evolutivo do IPTV

Os sistemas de IPTV em conjunto com as NGN estão a passar por um cenário evolutivo como é representado na Figura 3.

Neste momento estamos a entrar no nível 2 do cenário evolutivo da Figura 3, já nos é possível em algumas *set-top box* e TVs, aceder ao IPG[11], navegar na internet, aceder ao estado do tempo, ver as notícias através de *widgets*, e aceder a conteúdos de vídeo no PC, através do DLNA[12].

O DLNA é um *standard* que permite a partilha de conteúdos multimédia entre os diversos dispositivos de entretenimento duma rede Local.

[11] *Interactive Program Guide*
[12] *Digital Living Network Alliance*

Existem várias entidades reguladoras (ITU-T[13], ETSI, DVB, IETF, ATIS), que estão a trabalhar na definição dos *standards* de IPTV.

Todas estas entidades estão a trabalhar em conjunto. Cada uma delas é responsável pelo desenvolvimento dos *standards* associados a cada uma das camadas que compõem um sistema de IPTV. O ITU-T IPTV Global *Standards* Initiative (*GSI*) tem vários grupos de investigação, que desenvolvem *standards* de IPTV em coordenação com outras entidades (ex: ATIS IIF, ETSI, DVB, IETF).

A ATIS IPTV *Interoperability Forum* (IIF) desenvolve *standards* de IPTV em colaboração com o ITU-T, tem uma participação activa na Arquitectura IPTV, DRM, Metadados e Interoperabilidade.

A ETSI-TISPAN desenvolve *standards* para as NGN, incluindo IPTV. Tem uma participação activa no IMS, na arquitectura, no QoS, e na gestão de recursos IPTV.

O *Digital Video Boadcast Project* (DVB) está a desenvolver os *standards* de codificação de vídeo baseados em MPEG, e nos protocolos IETF. As especificações de IPTV do DVB são rectificadas pela ETSI.

O IETF desenvolve protocolos para IPTV para transporte, e controlo (IGMP, RTSP, RTP/RTCP, SIP,..), usados pelos *standards* mencionados anteriormente.

Segundo o ITU-T existem quatro domínios, sobre os quais assenta um sistema de IPTV, são eles:

Content Provider

Entidade que está autorizada a disponibilizar conteúdos, sendo estes de autoria própria ou de terceiro

[13] *International Telecomunication Union*

Service Provider

Entidade que fornece serviços de telecomunicações a clientes finais ou a outros fornecedores de serviços mediante uma tarifa ou contrato para o efeito. Esta entidade pode não operar a infra-estrutura de rede.

Network Provider

Entidade que opera e mantém os componentes duma rede, para que seja possível existir a funcionalidade do IPTV. Na maior parte dos casos o *Network Provider*, e o *Service Provider* são a mesma entidade.

End User

Pode ser uma pessoa ou uma organização que acede à rede, e utiliza os serviços existentes na mesma.

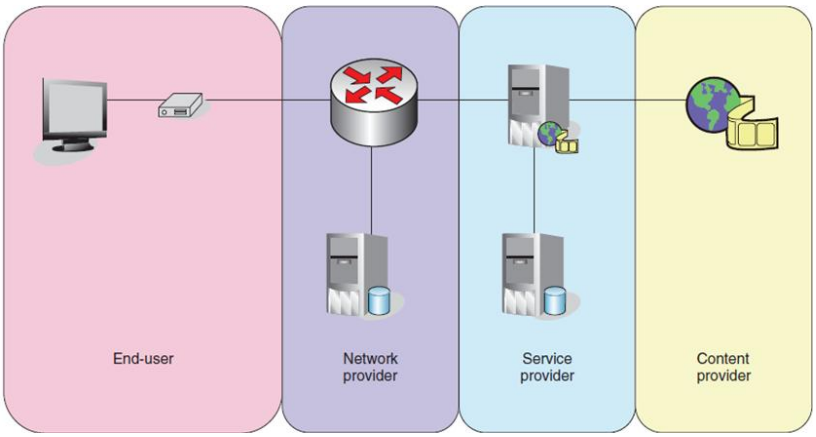

Figura 4 - Domínios Funcionais do IPTV

Com base nos domínios funcionais do IPTV, surgiu a *framework* da arquitectura funcional do IPTV, que assenta em sete grupos de funções (Figura 5), sendo estes os seguintes:

End-User Functions

As funções do utilizador final fazem a interligação entre o utilizador final, e a intra-estrutura do IPTV;

Application Functions

As funções de aplicação permitem que o utilizador final compre um conteúdo.

Content Delivery Functions

As funções de entrega do conteúdo facilitam a entrega dos conteúdos das *Application Functions* ao utilizador final, usando as capacidades das *Network Functions*. O conteúdo é distribuído para as *Content Delivery Functions,* através da *Application Functions* durante a oferta de serviço. O *Content Delivery Functions* é que suporta as funções de *playback control* do *End-User Functions* (ex: *trick mode* com o VoD e *Network* PVR[14]).

Service Control Functions

As funções de controlo de serviços proporcionam as funcionalidades de reservar, e libertar os recursos da rede, necessários para os serviços do IPTV. Este bloco é responsável pela reserva de largura de banda, que permite uma *stream* ser transmitida.

Management Functions

Este bloco permite a gestão global de todo o sistema, desde a configuração à monitorização.

[14] *Personal Video Recorder*

Content Provider Functions

Este bloco é responsável por controlar a venda das licenças dos conteúdos disponibilizados.

Network Functions

As funções de rede fornecem conectividade IP entre os componentes do serviço do IPTV, e as *End-User Functions*. As *Network Funtions* contribuem para o fornecimento de Qualidade de Serviço (QoS) requerido pelos serviços do IPTV.

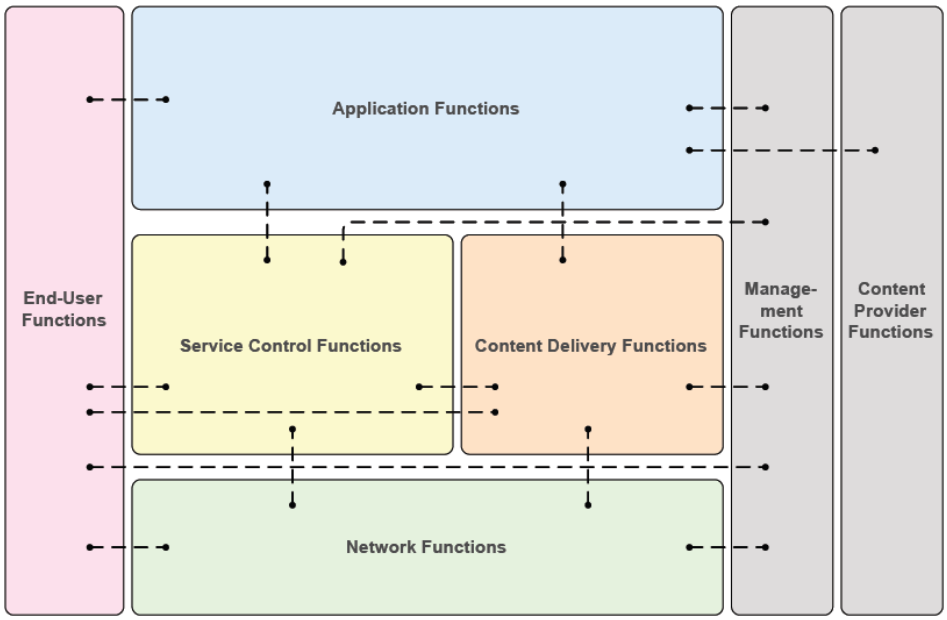

Figura 5 - *Framework* da Arquitectura Funcional do IPTV

Surgem três abordagens referentes as arquitecturas de IPTV, definidas pelas entidades atrás mencionadas.

As três abordagens são:

- "*Non*-NGN IPTV *Functional Architecture*" – Descreve uma arquitectura baseada em protocolos de controlo convencionais para as redes existentes;

- "NGN *Non*-IMS IPTV *Functional Architecture*" – Descreve uma arquitectura baseada nas Redes de Nova Geração com protocolos de controlo convencionais, mas sem IMS (IP *Multimedia Subsystem*);

- "NGN IMS IPTV *Functional Architecture*" – Descreve uma arquitectura de IPTV baseada nas Redes de Nova Geração utilizando protocolos de controlo IMS.

2.3 Protocolos de Rede

Como foi mencionado anteriormente, os sistemas de IPTV permitem distribuir essencialmente dois tipos de conteúdos, são eles a TV em directo (*live* TV), e o vídeo que já se encontra armazenado (*Video on Demand* - VoD).

O IP *multicast* é a forma de distribuir o mesmo conteúdo para múltiplos equipamentos em simultâneo, sem sobrecarregar toda a rede. Antes de existir, sempre que um conteúdo é distribuído para vários receptores em simultâneo, é efectuada uma "cópia"do mesmo para cada receptor, o que sobrecarrega o servidor e diminuí o desempenho geral da rede. Surge então o IP *multicast* com tecnologia inovadora que conserva a largura de banda, e diminui o processamento do servidor que emite apenas uma *stream* para todos os receptores. Optimizando a performance da rede, e a sua escalabilidade.

O funcionamento do IP *multicast* é baseado em árvores, criadas com base em apenas dois protocolos, o IGMP e o PIM. O IGMP é usado dentro do segmento de rede no qual se encontram as STB. Enquanto o PIM é usado pelos *routers* que interligam os segmentos de rede.

Se vários receptores em localizações diferentes fazem um pedido de ligação a uma *stream multicast*, usam o IGMP para comunicar com a rede. A seguir, o *router* da rede interpreta esse pedido, e cria ligações com os outros *routers,* criando uma ligação em cadeia, até à fonte da *stream multicast* através do protocolo PIM construíndo uma árvore. De seguida, em cada nó da árvore é feita

uma replicação dos pacotes da *stream* original até esta chegar aos receptores (Figura 6).

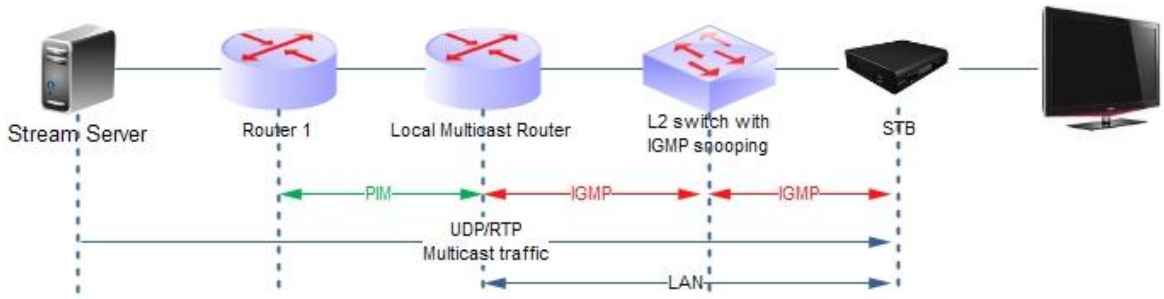

Figura 6 - *Multicast Network*

Quando é difundida a TV em directo, é usado o IGMP v2/v3 para recepção da *stream* que é enviada em MPEG *transport stream* através do IP *multicast*.

Sempre que queremos ver um conteúdo que já se encontra armazenado, (ex: Alugar um Filme), é usado o *Real Time Streaming Protocol* (RTSP) através de IP *unicast*. O IP *unicast* é um método sobre o qual é possível enviar informação para um único computador. Tendo em conta, que se dois ou mais computadores acedem ao mesmo conteúdo, o número de sessões abertas no servidor, é igual ao número de computadores que estão ligados.

Este assunto é detalhado no capítulo 3.

2.4 Codificação de Vídeo e Áudio

A televisão surge graças à tecnologia analógica. Os programas de televisão são gravados com uma câmara analógica que capta a realidade, e em tempo real, cria um sinal de vídeo, que posteriormente é codificado, modulado e difundido através de emissores, até às nossas casas. Finalmente, nas nossas casas, o sinal é desmodulado e descodificado no televisor que apresenta o vídeo.

Os tipos de codificação analógica para televisão são o PAL, NTSC, e o SECAM (ver ref. e extensos). Todos estes *Codecs* actuam sobre as três componentes de vídeo RGB.

A figura que se segue mostra a largura de banda utilizada para cada um destes *codecs* analógicos:

System	*Country*	*Bandwidth (MHz)*
NTSC	United States, Japan, Canada, Mexico	4.2
PAL	Great Britain	5.0
PAL	Austria, Germany, Italy	5.5
SECAM	France, Russia	6.0

Tabela 1 - Largura de Banda ocupada por cada sistema de codificação da TV analógica

Segundo a Tabela 1, a largura de banda necessária para codificar um canal analógico é de 5MHz no caso do PAL, mas para transportar este sinal é preciso uma Largura de Banda de 8MHz. Uma linha ADLS 2+ tem uma largura de banda de 2,2MHz, e consegue fornecer 2 canais em simultâneo + internet + voz. Se tivermos em conta esta pequena comparação, não há dúvida de que a codificação digital é mais eficiente no que toca ao aproveitamento do espectral, logo é possível ter cinco ou mais canais, onde anteriormente cabia apenas um.

O MPEG2 é um dos *codecs* usados no IPTV. Este codec é um padrão na codificação de vídeo e áudio digital, pois permite o armazenamento, e transporte dum filme, utilizando a largura de banda/capacidade disponíveis actualmente, recorrendo à compressão do áudio e do vídeo. É evidente que existe uma perda de qualidade do vídeo e do áudio devido à elevada compressão a que estão sujeitos. No entanto, se não for usada compressão, um DVD (4,37GB) apenas armazena 3,7 minutos de vídeo, tendo em conta que para codificar um segundo de vídeo sem compressão são necessário 21Mbytes.

A resolução máxima do MPEG2 é de 720x480 por *frame* com uma cadência de 30 *frames* por segundo no NTSC. No sistema PAL é de 720x576, com uma cadência de 25 *frames* por segundo.

O MPEG2 pode ser codificado de duas formas, ou com CBR (*Constant Bit Rate*) no qual é necessário ter um *bit rate* de 6 Mbps para não haver perda de qualidade perceptível, ou então VBR (*Variable Bit Rate*), no qual é necessário um bitrate médio de 4Mbps para que a qualidade do vídeo seja semelhante à do original.

O H264 é também usado pelo IPTV, utilizando outras técnicas de compressão mais avançadas, que exigem cerca de 10 vezes mais processamento para codificar, e cerca de 4 vezes mais para descodificar em relação ao MPEG2, necessita de cerca de metade do *bit rate* de transmissão relativamente ao MPEG2.

Seguem alguns dos pontos que aumentam a taxa de compressão do H264, relativamente ao MPEG2.

No H264 as *frames* podem ser de três tipos: I (*Intra – frame* completo), P (*Predictive – frame* diferencial em relação à ultima *frame* visualizada), e B (Bi-*Predictive* – é uma *frame* que não é um diferencial em relação à ultima *frame* visualizada, mas também à *frame* que ainda vai ser visualizada.

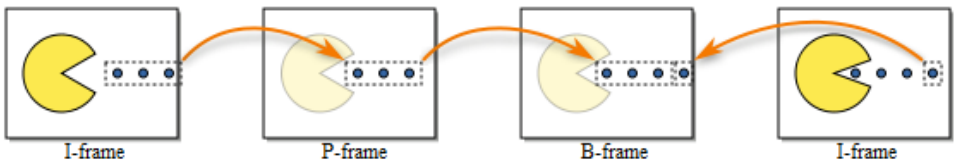

Figura 7 – Tipos de *frames*

As *frames* são divididas em macroblocos de dimensões distintas, enquanto no MPEG2 são fixos.

Actualmente, estes dois *codecs* são os mais usados nos sistemas de IPTV. Existem outros, tais como, o WMV (*windows media video*), e o FLV (*flash video*) que são menos usados porque são *codecs* proprietários, ou não suportados pela maioria das *set-top box*, e dispositivos móveis.

Este assunto é detalhado no capítulo 4.

2.5 QoS

Os sistemas de IPTV requerem um elevado nível de Qualidade de Serviço (QoS), de forma a serem implementados a larga escala. O QoS é dividido em duas categorias, qualidade de áudio, e qualidade de vídeo.

Pode ser definida por qualidade de áudio, a capacidade que o sistema tem de recriar as características principais do sinal de áudio original. Esta pode ser afectada por diversos factores, tais como: *codecs* de áudio (compressão), rede de transmissão ou limites de velocidade.

Quanto mais comprimido o áudio for, menor será a sua qualidade. Hoje em dia, os *codecs* de áudio permitem fazer uma maior compressão sem que as características principais sejam perdidas, obtendo assim uma ocupação de largura de banda menor.

A qualidade do vídeo padece dos mesmos factores condicionantes da qualidade do áudio, sendo que o impacto dos mesmos, manifesta-se através da "*pixelização*" da imagem, arrastamento de blocos, retenção de blocos, entre outros.

O QoS é definido no ITU-T E.800, como efeito global de performance que determina o grau de satisfação dum utilizador, quando usa um serviço. Em telecomunicações, QoS é medido através do desempenho da própria rede. Os mecanismos de QoS incluem qualquer mecanismo, que contribui para melhorar o desempenho geral da rede, de forma a aumentar a experiência do utilizador

final. Estes mecanismos são implementados a diferentes níveis, por exemplo, a camada de rede inclui mecanismos de gestão de tráfego tais como, *buffering* e escalonamento de filas de espera, para permitir diferenciar o tráfego proveniente de cada aplicação. Todos os mecanismos de QoS têm como objectivo final garantir a QoE (*Quality of Experience*).

A Qualidade da Experiência está definida no Apêndice I do ITU-T P.10/G.100, como uma aceitação duma aplicação ou serviço, dum modo subjectivo por parte do utilizador final.

Na Figura 8 podemos ver um resumo dos factores objectivos, e subjectivos que afectam a qualidade da experiência.

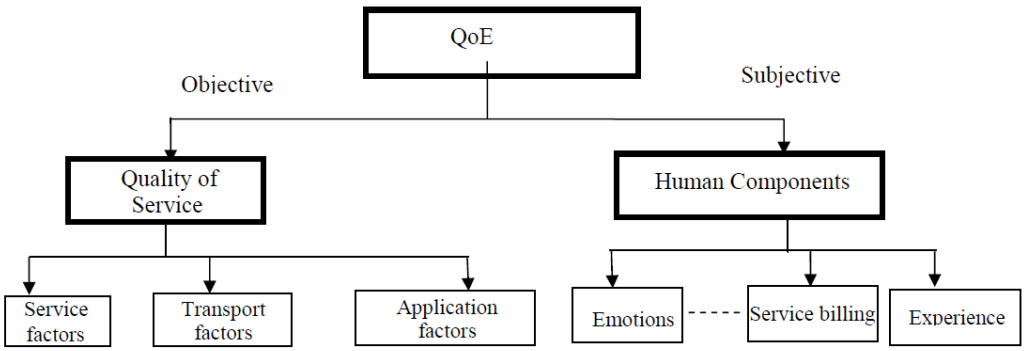

Figura 8 - QoE (*Quality of Experience*)

Este assunto é detalhado no capítulo 5.

3. Protocolos de Rede

Neste capítulo são descritos os protocolos de rede necessários para uma rede de IPTV.

3.1 IGMP (Interior Gateway Multicast Protocol)

As mensagens do IGMP são encapsuladas num Datagrama IP, com o IP *protocol number* 2. Os *routers* enviam as *queries* IGMP para todas as máquinas através do endereço 224.0.0.1, para solicitar grupos *multicast* que têm clientes activos.

O IGMP tem actualmente três versões, IGMP v1, v2 e v3.

No IGMP v1 (RFC1112) são apenas definidas dois tipos de mensagens IGMP:

Membership query

Esta mensagem é enviada periodicamente pelo "*Querier router"* que está no segmento de rede, para descobrir se algum dos *hosts* ainda está interessado em receber tráfego de um grupo *multicast*.

Membership report

Esta mensagem é enviada pelos *hosts* para informar *o "Querier router"* que pretendem ligar-se a uma fonte específica de tráfego *multicast*. No IGMP v1 não existe um processo para sair do grupo *multicast*, a única forma de o fazer é o *host* deixar de enviar *reports* ao "*router Querier*" até este detectar, através do mecanismo de *timeout* que não recebe *reports* dos *hosts*.

Nesta versão não existe o mecanismo de eleição do "*Querier router*" possibilitando vários *routers* enviarem "membership query" para a rede.

No IGMP v2 (RFC2236) é adicionada uma mensagem relevante além das anteriores definidas na versão 1:

Leave group

É usada pelos *hosts* para manifestar a sua intenção de sair do grupo *multicast*, desta forma, o *router* que está a enviar o tráfego suspende a sua emissão, sem ter de esperar pelo tempo de *timeout*, evitando assim tráfego desnecessário na rede.

Foram também adicionadas as seguintes funções:

Eleição do *router Querier*

Numa rede multi-acesso (com vários pontos de acesso ao exterior), é eleito um *"router Querier"* baseado no seu endereço IP, para que apenas um *router* por segmento de rede envie as mensagens de *"membership query".*

Group-specific query

O *router* envia uma mensagem de *"group-specific query"* antes de executar o *timeout* para o grupo *multicast*, desta forma assegura-se que não existe nenhum *host* no segmento, que ainda está interessado em receber a emissão do grupo *multicast*.

No IGMP v3 (RFC3376) são usados os mesmos tipos de mensagens: "*membership query*" e "*membership report*".

No entanto as mensagens de *"membership Querier"* podem ser de três tipos:

General

Esta é enviada pelo *Querier router* que está no segmento de rede periodicamente, para descobrir se algum dos *hosts* ainda está interessado em receber tráfego de um grupo *multicast*.

Group-specific

Definição idêntica à anterior, mas direccionada a um determinado grupo.

Group-and-source-specific

O mesmo que o *general*, mas direccionado a um determinado grupo, e a determinadas fontes.

Existe também o **IGMP *snooping***, este mecanismo consiste em analisar as negociações IGMP existentes na rede, para só deixar passar tráfego *multicast* para os *hosts* que o queiram receber. Resumindo, este sistema actua como "interruptor *multicast*" evitando que os outros *hosts* do segmento de rede não recebam tráfego *multicast* que não solicitaram.

Associado ao **IGMP *snooping*** existe outro mecanismo denominado **IGMP *fast leave***. Os *swicths* com este mecanismo conseguem interpretar as mensagens de IGMP, e sempre que detectam um *leave group*, interrompem de imediato a transmissão da *stream* que o cliente está a receber. Removendo o cliente do grupo *multicast* a que está ligado, evitando que a mensagem se propague até ao *router Querier*. Desta forma, não existe tanto atraso sempre que é feito um *zapping* de canais por parte do utilizador, pois o *router Querier* não está a enviar *queries* de confirmação para todas as STBs. Sempre que uma STB sai de um grupo *multicast*, esta operação é feita pelo *switch* que está mais perto da STB.

A melhor forma de entender este mecanismo é demonstrar as fases de negociação entre a STB e o *router Querier*, sempre que é feito um *leave group*, com e sem *fast leave*.

As duas figuras seguintes (Figura 9, figura 10) representam as etapas de negociação, sempre que uma STB sai do grupo *multicast*. Para efeitos demonstrativos, assume-se que é a STB da porta 1, que vai enviar a mensagem

de *IGMP leave group*, tendo em conta que as STB das portas 2 e 3 não fazem nenhuma operação.(IGMP Fast Leave)

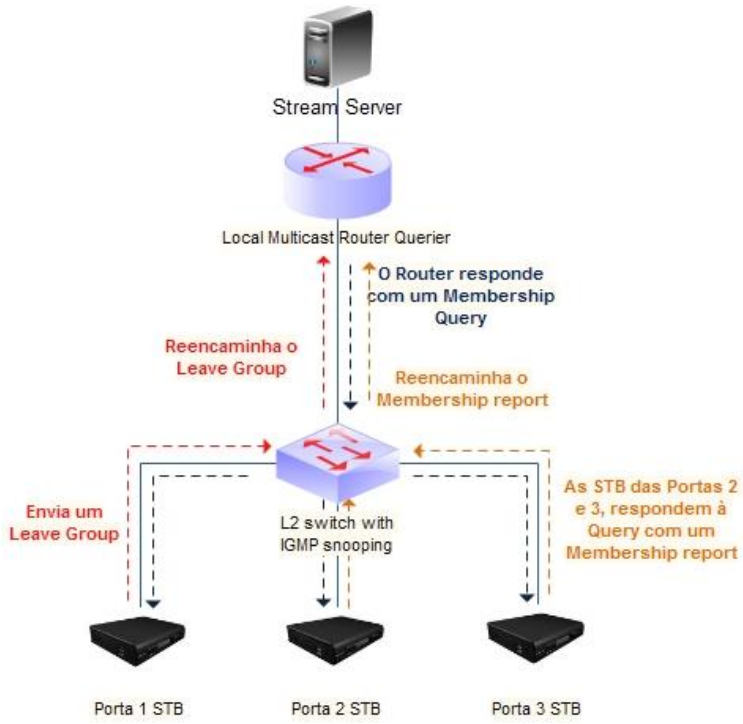

Figura 9 - *IGMP Leave Group*, sem o mecanismo de *Fast Leave*

Nesta figura, podemos verificar que a STB da porta 1 manifesta a intenção de sair do grupo *multicast* no qual se encontra. De seguida, são enumerados os passos até à operação estar concluida.

1) Quando o *switch* de acesso recebe na porta 1 a mensagem de *"leave group"* proviniente da STB, encaminha-a para o *router* de *multicast*.
2) Assim que o *router* de *multicast*, recebe a mensagem de *"leave group"*, envia uma mensagem de *"membership query"* para o *switch* de origem da mensagem. O *switch* por sua vez reencaminha a *query* para todas as STB que estão no mesmo grupo *multicast* (no mesmo canal de TV).
3) Todas as STBs do *switch* respondem à mensagem de *"membership query"* com um *"membership report"* para o *switch*, e a seguir este encaminha-as para o *router multicast*.

4) Só depois do processamento dos passos anteriores, é que a STB sai do grupo *multicast* no qual se encontra, e o tráfego é interrompido na porta 1.

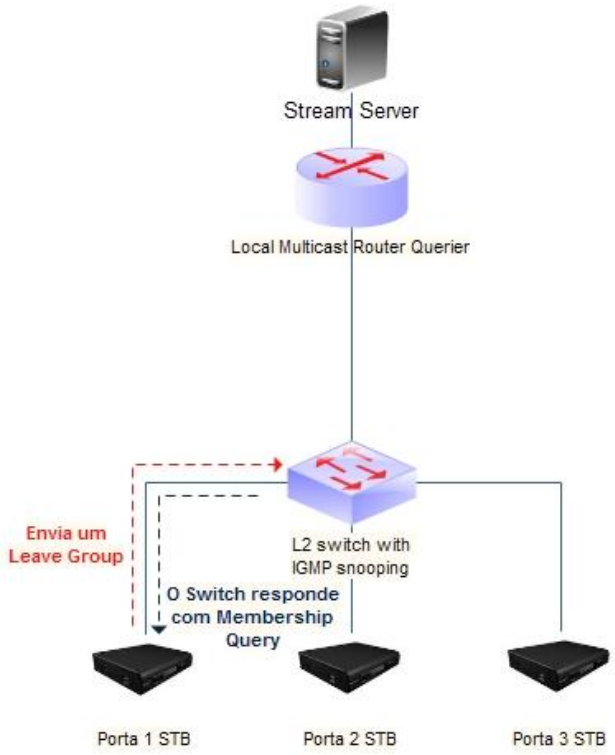

figura 10 - IGMP *Leave Group*, com mecanismo de *Fast Leave*

Nesta figura verificamos que a STB da porta 1, manifesta a intenção de sair do grupo *multicast* no qual se encontra. De seguida, são enumerados os passos até à operação estar concluida.

1) Quando o *switch* de acesso, recebe na porta 1 a mensagem de *"leave group"* proviniente da STB, envia imediatamente uma mensagem de *"membership query"* para a STB que fez o pedido, evitando desta forma o reencaminhamento de mensagens até ao *router Querier*.

2) Se a STB não responder em tempo útil, o *swicth* remove-a imediatamente do grupo *multicast* a que esta ligada.

Como é possível verificar, o "*IGMP fast Leave*" vem diminuir o tempo que uma STB leva a sair de um grupo *multicast*, aumentando a velocidade de transição entre canais, intensificando a *QoE* na função de *zapping*, principalmente em ambientes hoteleiros, nos quais podem existir várias dezenas de utilizadores a interagir com as STB em simultâneo.

3.2 PIM (*Protocol Independent Multicast*)

O PIM (*Protocol Independent Multicast*) é um protocolo que permite encaminhar de forma eficiente grupos *multicast* através de vários segmentos de rede. Existem essencialmente dois modos de difundir *multicast* com o PIM, *Sparce-Mode* e o *Dense-Mode*. Tanto num como noutro, são construidas as árvores de *multicast* com base nos *Rendezvous Points*. A diferença reside na forma como é feita a distribuição dos conteúdos. No modo *Sparse-Mode*, só é enviado conteúdo, quando este é solicitado, logo, se não existirem receptores "*interessados*" em receber *multicast*, não existe nenhum fluxo de *multicast* na rede, apenas são contruídas as árvores. (RFC4601, 2006)

No modo *Dense-Mode*, a rede é "inundada" com o conteúdo de todos os grupos *multicast*, mesmo que nenhum receptor esteja "*interessado*" em receber, e só de seguida é que começa a parar a emissão por cada *link* que não tenha pedido *multicast*. Mas para tal, é enviada uma mensagem de "*Prune*" pelos *routers* que não têm receptores *multicast* activos.

O *Sparse-Mode* é mais usado, por ser mais eficiente que o *Dense-Mode*, na gestão de recursos de rede ocupados. Nos sistemas de IPTV é o modo aconselhado, existem inclusivamente alguns equipamentos (exemplo: *Microtick*), que não trazem suporte para PIM-DM, apenas trazem o modo *Sparse-Mode*.

A RFC mais actual do PIM SPARSE-MODE é a RFC4601 publicada a Agosto de 2006, substituindo a anterior RFC2362.

De seguida, são descritos os termos importantes para entender o funcionamento do PIM-SM.

***Rendezvous Point* (RP)**

Um **RP** é um *router* que está configurado para ser usado como raíz de uma árvore de distribuição sem fonte específica *(non-source-specific distribution tree)* para um grupo *multicast*. As mensagens de *"JOIN"* provinientes dos receptores com destino a um grupo *multicast*, são enviadas para o **RP**, e o conteúdo é enviado desde o emissor até ao **RP**, sendo depois replicado por outros receptores que querem fazer *"JOIN"* ao mesmo grupo.

***Designated Router* (DR)**

Sempre que numa rede local (LAN) existam vários *routers PIM-SM* conectados, um deles é eleito **DR**, isto é, todos os receptores vão comunicar com o **DR**, que vai servir de intermediário para os outros *routers* com PIM-SM. Por *interface*, apenas é eleito um DR, é um processo de eleição muito simples, ou seja, o *router* com o IP mais elevado é eleito **DR**.

***Multicast Routing Information Base* (MRIB)**

MRIB é essencialmente uma tabela que representa a topologia de rede *multicast*, construida através da tabela de encaminhamento *unicast*, ou então, através de protocolos como *Multiprotocol Bonduary Gateway Protocol* (MBGP), sempre que são *routers* que se situam nas fronteiras dos *Autonomous-Systems* entre operadores. No PIM-SM esta informação é usada para decidir para onde devem ser encaminhadas as mensagens de *JOIN/PRUNE*. A segunda função do MRIB fornece métricas de encaminhamento para os endereços de destino.

Reverse Path Forwarding* (RPF) *Neighbor

Este mecanismo evita *loops* no encaminhamento dos pacotes *multicast*. No *multicast* o encaminhamento é feito com base na origem, e não no destino. Isto é, sempre que um pacote *multicast* entra numa *interface* de um *router*, este verifica na sua tabela quais são as redes que consegue atingir através dessa *interface*, neste caso o *Reverse-Path* do pacote. Se o *router* encontrar uma correspondência para o IP de origem do pacote, a verificação do RFP vai dar "OK", e o *router* encaminha o pacote para as outras interfaces que estão a participar naquele grupo de *multicast*. Os *routers* RFP só encaminham pacotes que provêm da *interface*, cujo IP encontra-se na tabela de encaminhamento.(Reverse-Path-Forwarding, 2009)

***Tree Information Base* (TIB)**

Esta é uma tabela que contém os estados de tudo o que está relacionado com o PIM no *router*. Esta tabela é criada através da recepção das mensagens de PIM *JOIN/PRUNE,* IGMP, ou através da informação dos *hosts Multicast Listener Discovery* (MLD). Com este conjunto de informações, o *router* fica com os estados de todas as árvores de distribuição existentes no *router*.

***Multicast Forwarding Information Base* (MFIB)**

A TIB contém toda a informação necessária para o encaminhamento dos pacotes multicast, contudo, o encaminhamento de pacotes apenas com a TIB é pouco eficiente. Para colmatar este problema, na implementação dos *routers*, é construída uma outra base de dados denominada MFIB, com base nos estados existentes na TIB, e outras informações de encaminhamento do *router*.

3.2.1 Breve descrição sobre o PIM-SM

O PIM-SM está desenhado de forma a optimizar o envio de tráfego *multicast* através das estruturas de redes existentes, tem em conta os seguintes objectivos:

a) Manter o modelo tradicional do serviço de IP *multicast*, ou seja, implica que o receptor inicie sempre a negociação com o grupo *multicast*. Neste modelo, as fontes apenas enviam os pacotes de *multicast* até ao *first-hop Ethernet*, sem qualquer tipo de sinalização. Os receptores tratam da sinalização, com o objectivo de juntarem-se ao grupo *multicast* do qual querem receber dados.

b) O modelo de *leave* do sistema *multicast* mantém-se o mesmo, devido ao PIM-SM ser um protocolo *router-to-router*, o que significa que os *hosts*, não têm que sofrer nenhuma actualização, apenas é aplicado na rede.

c) PIM-SM suporta tanto *shared trees,* como *source distribution trees*. Para as *shared trees*, o PIM-SM utiliza um *router* central denominado *Rendezvous Point* (RP), que funciona como raiz da árvore partilhada. Todas as fontes de tráfego *multicast* enviam o seu tráfego para o RP, que por sua vez, encaminha os pacotes através da árvore *multicast* comum a todos os *routers* que são membros da mesma. Existem também as *source trees*, que ligam directamente as fontes aos receptores, e é criada uma árvore por cada fonte *multicast*. As *source trees* são consideradas as *shortest-path trees* do ponto de vista das tabelas de encaminhamento *unicast*. O PIM-SM pode usar qualquer um dos tipos de árvores, ou ambas em simultâneo.

d) Manter a independência de qualquer protocolo *unicast*.

e) Usar mecanismos *soft-state*, para se adaptar às mudanças na topologia de rede, e à dinâmica do *multicast*. *Soft-state* significa que, o *router* está actualizado, o estado da configuração deste é mantido por um curto período, e expira ao fim de um tempo definido.

Para melhor entender o protocolo PIM-SM, está dividido em diversas partes, enumeradas a baixo, e as quais estão resumidas mais à frente:

- Mensagens de *Hello*
- Encaminhamento de pacotes *multicast*
- Como se faz o JOIN a uma *shared-tree*
- *Shortest-path tree (SPT) switching*
- Registo no *Rendezvous Point*
- *Pruning interfaces*
- *Assert messaging*
- Eleição do RP

Como exemplo, assume-se à partida que o sistema é estável, isto é, já está feita a eleição do RP. Antes de explicar cada ponto anterior, é necessário abordar os seguintes conceitos, de forma resumida, imprescindíveis ao bom entendimento do PIM-SM:

- *Flooding and reverse path forwarding (RPF)*
- *Shortest-path trees*
- *Shared trees*

3.2.2 Flooding e Reverse Path Forwarding

Flooding é um mecanismo de encaminhamento usado nos *routers multicast*, para evitar que estes não dependam de nenhuma informação de encaminhamento. O princípio deste mecanismo consiste na transmissão de um pacote para todas as *interfaces* do *router*, excepto para aquela de onde veio o pacote.

Para limitar o número de vezes que o pacote é replicado, este tem um contador (*time-to-live*), que é decrementado cada vez que o pacote passa num *router*, até chegar a zero, altura em que o pacote é descartado.

O problema do *flooding* é que cria um número exponencial de cópias de cada pacote, mas por outro lado, garante que entrega da cópia do pacote em cada nó da árvore *multicast*.

Reverse Path Forwarding (RPF) é uma técnica de encaminhamento, introduzida por Yogen Dalal - um dos mentores de *Silicon Valey*, que consiste na optimização do *flooding*, no qual o *router* aceita o pacote da *source* **S** através da *interface* **I**, mas apenas se esta *interface* é a usada pelo *router* para alcançar a *source* **S**. Esta verificação é feita através da consulta da tabela de encaminhamento *unicast* do *router*.

Esta técnica optimizou o tamanho do *overhead* comparado com o do *flooding*, porque o *router* só aceita os pacotes de apenas um *neighbor*, e apenas o replica uma vez, fazendo com que cada pacote seja transmitido apenas uma vez em cada *link*.

3.2.3 Shortest-Path Trees

Shortest-path trees (SPT) são também denominadas *source-based trees.* A política de encaminhamento é baseada no caminho mais curto até à source, através da informação da tabela de encaminhamento de *unicast.* Se o cálculo da métrica é baseado no número de saltos entre *routers*, significa que os ramos da SPT têm o número mínimo de saltos. Se a métrica é baseada na comparação ao atraso, então os ramos vão ser os que têm menos atraso.

Nas SPT, para cada *source multicast*, existe uma árvore *multicast* correspondente, que liga a *source* a todos os receptores. Uma vez criada uma árvore para uma *source* associada a um grupo *multicast*, todo o tráfego dos membros desse grupo passará ao longo de toda a àrvore. As STPs têm tabelas

com entradas (**S**,**G**), com uma lista de *interfaces* de saída, na qual **S** é o *source address* e o **G** é o grupo *multicast*.

Algums dos protocolos que usam STPs são DVRMP (*Distance Vector Multicast Routing Protocol* – RFC 1075) e MOSPF (*Multicast Open Shortest Path First* - RFC 1584) que são protocolos *DENSE-MODE*.

3.2.4 Shared Trees

As *shared trees* são denomidas RPT (*Rendezvous Point Trees*) no PIM-SM, pois a raíz destas árvores tem origem num *router* central denominado *Rendezvous Point* (RP). O *Rendezvous Point* recebe todo o tráfego das *sources*, e encaminha-o para os receptores, para tal, os membros do RP, enviam mensagem de *join* até ao RP. Ao contrário do PIM *Dense-Mode*, não se assume que todos os membros são receptores do tráfego *multicast*, isto é, podem existir membros, que não estejam interessados em receber dados das *sources* existentes no RP.

Com as RPT, apenas existe uma árvore por cada grupo *multicast*, não interessando o número de *sources* que lá existam. Apenas os *routers* que estão na árvore conhecem o grupo *multicast*, e o tráfego só é enviado para os receptores que o pretendem.

Estas árvores contêm tabelas com a informação (*,**G**), nas quais **G** é o grupo *multicast*.

Com o *Rendezvouz Point*, os receptores têm um local para onde enviar as mensagens de *join*, mesmo que ainda não exista a *source* pretendida.

As *shared-trees* são unidireccionais, o que significa que o fluxo de dados é apenas do RP para os receptores, o que torna o PIM-SM tão vantajoso para distribuir IPTV.

Para uma *source* enviar dados para o RP e enviá-los para a árvore, os dados têm que ser primeiro enviados para o RP através de um "túnel", e só depois é que este efectua o *multicast* para os receptores interessados. Isto significa que se

um receptor de multicast quiser ser também uma *source* do RP, não pode usar a árvore da qual é receptor para enviar os dados para o RP.

Após a introdução destes conceitos, é abordado o protocolo de PIM-SIM, através dos seguintes tópicos, já enumerados no capítulo 3.2.1:

3.2.5 Mensagem de *Hello*

As mensagens de *Hello* servem para os *routers* PIM descobrirem os seus vizinhos. Estas mensagens são tocadas pelo grupo *multicast* 224.0.0.13 (*All-PIM-Routers group*). Na mensagem de *Hello*, existe um campo *"Holdtime"* que define a validade da informação da mensagem.

Não é enviado qualquer tipo de *acknowledgement* sempre que é recebida uma mensagem de *Hello* num *router*. Após a recepção de uma mensagem de *Hello* por uma *interface*, esta não é imediatamente adicionada à lista de *outgoing interfaces* para encaminhar *multicast*. Para tal o PIM-SM usa uma mensagem de *join* específica. Isto é, um receptor sempre que pretende receber *multicast*, deve enviar uma mensagem de *join* para o *router* PIM, e só de seguida é que a *interface* do *router* PIM fica disponível na lista de *outgoing interfaces* do mesmo.

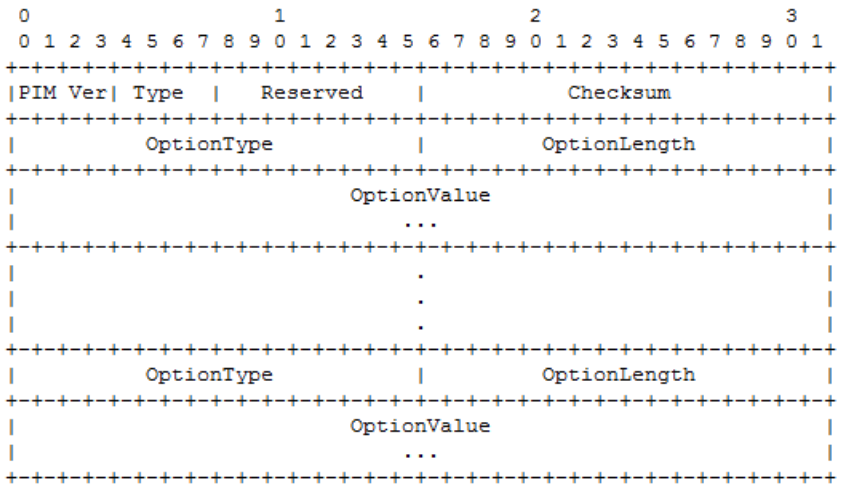

Figura 11- formato da mensagem de *Hello* do PIM-SM

3.2.6 Encaminhamento de pacotes *multicast*

Os *routers* PIM-SM, encaminham o tráfego *multicast* para todas as *interfaces*, que têm como destino receptores que previamente, manifestam o interesse por se juntar ao grupo *multicast*.

Os receptores manifestam essa intenção através do envio de uma mensagem de IGMP *Membership Report*, a cada grupo que pertencem. Estas mensagens de IGMP têm um TTL=1 (*time-to-live*) de forma a não sairem de dentro do segmento de rede, apenas chegam ao *router* que tem acesso à WAN, este de seguida, converte os pedidos de IGMP para PIM. Este *router* começa por efectuar o RPF antes de encaminhar o pedido de PIM *Join*. O método de RPF *check* que o *router* efectua, depende do tipo de árvore (RPT ou SPT). Se for RPT, o RPF *check* verifica o IP do *Rendezvous Point* (RP). Se for STP, o RPF *check* verifica o endereço da *source*.

3.2.7 Join Shared-Tree

Sempre que um cliente efectuar um *join* a um grupo *multicast*, ele envia uma mensagem de IGMP para o *upstream router*, isto significa, que esse *router* vai aceitar tráfego *multicast* desse grupo. Para que possa efectuar esta operação, notifica o *Rendezvous Point*, para se juntar à RPT. Esta notificação é feita através do envio de uma mensagem de PIM *Join* (*****,**G**), para o PIM *neighbor* mais próximo. As mensagens de *Join*, são distribuidas em *multicast* no endereço 224.0.0.13, de *router* em *router* até chegar ao RP. Isto significa que numa rede multi-acesso, todos os PIM *neighbors* têm conhecimento do *join*, mas só o *upstream* PIM *neighbor* é que efectuará o *join*.

3.2.8 *Designated Router*

Sempre que existem vários *routers* ligados no mesmo segmento rede com acesso ao exterior, a rede denomina-se rede multi-acesso. Nesta situação um dos *routers* tem que ser eleito para agir como *designated router*. O *designated router* (DR) é responsável por tratar, e enviar as mensagens de *join/prune* para

o *Rendezvous Point*. A eleição do DR é feita através das mensagens de *Hello*, o *router* que tiver o IP mais elevado fica com a função de DR.

3.2.9 Mensagem *Assert*

Nas redes multi-acesso podem existem vários caminhos alternativos para a *source*, ou para o RP. Esta situação pode originar a recepção de pacotes duplicados por parte dos membros do grupo *multicast*, pois podem existir vários *routers* a enviar a mesma informação por caminhos diferentes.

Para resolver esta situação o PIM-SM, passa por uma selecção prévia dos caminhos, através das mensagens de *Assert*. Desta forma, sempre que existirem dois ou mais *gateways* para a mesma *source* ou RP, um deles é designado "*designated forwarder*".

3.3 RTP (Real Time Transport Protocol)

O *Real Time Transport Protocol* (RTP) é um protocolo desenvolvido para transporte de áudio/vídeo. Este protocolo criado por um grupo de trabalho do IETF, e a sua primeira publicação é em 1996 na RFC1889, que em 2003 é substituída pela RFC3550. Actualmente, a RFC5761 é a que se encontra em vigor.(Perkins, Glasgow, Westerlund, & Ericsson, RFC5761 Multiplexing RTP Data and Control Packets on a Single Port, 2010)

O RTP fornece funções de transporte *end-to-end*, tipicamente utilizadas para transporte de dados em tempo real. É usado para transportar áudio no VoIP (bidireccional), e áudio+vídeo nos sistemas de IPTV (unidireccional), sobre o protocolo UDP. Com a vantagem de não ser sensível aos atrasos, mas com o problema de não garantir entrega, logo, o controlo de congestionamento é efectuado pelos protocolos das camadas superiores.

Este protocolo não oferece qualquer tipo de controlo, qualidade de serviço, ou mesmo garantia de entrega, sendo essa função entregue ao RTCP (*Real-Time*

Transport Control Protocol), que vem sempre associado ao RTP. A sua principal função é enviar informação de controlo, e parâmetros de QoS periodicamente.

O RTCP em conjunto com RTP permite compensar o *jitter,* e detectar a alteração da sequência dos pacotes, comum durante as transmissões numa rede IP. O grupo de trabalho do RTP está também a trabalhar em conjunto com o grupo do *Datagram Congestion Control Protocol* (DCCP), para garantir que o RTP seja transportado eficientemente em cima do protocolo DCCP. O DCCP opera na camada de transporte, que implementa ligações bidireccionais em *unicast*, e controla o congestionamento. Tem como objectivo aumentar o desempenho do transporte de *streams* de áudio e vídeo, porque permite controlar os atrasos, e garante a ordem de entrega. Confere funções idênticas às do TCP em relação à garantia de entrega, e controlo de congestão, só que é mais rápido, diminuindo os atrasos inerentes ao processo de controlo. (Kolher, Handley, & Floyd, 2006)

Outro ponto forte é a capacidade de transferir dados para vários destinos através do *multicast*.

Uma das funções adicionais do RTP, especificadas na RFC3551 que foi substituída pela RFC5761, é a definição de perfis associados ao *payload* de áudio e vídeo que o RTP transporta, denominando-se assim RTP/AVP (*audio and video payload types*). Seguem alguns exemplos dos tipos de *payload* do RTP/AVP:

Payload type	Name	Type	No. of channels	Clock R(Hz)	Description	Ref
3	GSM	audio	1	8000	European GSM	RFC3551
33	MP2T	A/V	1	90000	MPEG2 TS video	RFC2250
…	…	…	…	…	…	…
dynami	H264	video	Nd	90000	H.264 MPEG-4 Part 10	RFC3984

Tabela 2 - Tipos de Payload RTP/AVP

3.4 RTCP (*Real Time Control Protocol*)

O RTCP, definido incialmente na RFC3550, funciona em conjunto com o RTP. Enquanto o RTP faz a entrega dos dados, o RTCP envia periodicamente pacotes de controlo, para aferir a qualidade da entrega dos dados por parte do RTP. Com

base no *feedback* é possível identificar problemas de atraso, ou mesmo de qualidade de serviço, e saber se o problema é global, ou parcial. Tipicamente a largura de banda ocupada pelo RTCP deve representar 5% do volume do tráfego RTP. O RTCP é responsável por atribuir nomes canónicos (CNAMES) aos participantes (entenda-se por participante um emissor, ou receptor de dados RTP).

Para que o RTCP consiga controlar o RTP usa cinco tipos de mensagens: (Schulzrinne, Casner, Frederick, & Jacobson, 2003)

SR (*Sender Report*)

Esta mensagem contém um relatório de envio, e recepção de pacotes RTP por participantes que sejam fontes activas.

RR (*Receiver Report*)

Esta mensagem contém um relatório com a recepção por parte dos participantes passivos, que apenas recebem os dados.

SDES (*Source Description Items*)

Esta mensagem descreve o participante, e inclui a informação do seu CNAME.

BYE

Esta mensagem que indica a saída do participante da comunicação.

APP

Esta mensagem contém funções específicas da aplicação, o nome escolhido deve ser baseado na entidade que a representa para ser único.

O RTCP com base nas mensagens anteriores efectua quatro funções para garantir o bom funcionamento do RTP.

Essas funções são:

a) Fornecer *feedback* sobre a qualidade da distribuição de dados RTP, a principal função, pois assegura de alguma forma a função de protocolo de transporte (análogo ao TCP), efectuando as funções de controlo de fluxo, e congestão. Este *feedback* pode ser útil no caso de existir codificação adaptativa, com o propósito de controlar o *bit rate* que mais se adequa ao estado da rede. Nos ambientes *multicast* este protocolo é essencial para reportar eventuais falhas na distribuição.

b) O RTCP é reponsável por atribuir um CNAME *(canonical name)* a cada fonte RTP, este nome é transportado no RTCP, e actua como identificador da fonte. No caso de existir um conflito, ou um programa ser reiniciado, os receptores tentam encontrar de novo a fonte com base no CNAME. Este identificador serve também para sinalizar um grupo de *streams*, no caso de associação do Áudio + Vídeo, muito usado nos sistemas de IPTV.

c) As funções anteriores exigem que todos os participantes enviem pacotes RTCP. Para que não existam conflitos, deve ser calculada uma taxa com base no número de participantes, para aumentar escalabilidade do RTP. Quanto maior for o número de participantes, maior é a latência introduzida na troca de mensagens RTCP.

d) A quarta função é opcional, e consiste em transportar o mínimo de informação de controlo. Mas esta opção só deve ser usada em ambientes nos quais não é necessário tanto controlo das sessões dos participantes.

As funções definidas entre o ponto 1 e 3 são exigidas sempre que o RTP opera num ambiente *multicast*.

3.5 RTSP (*Real Time Streaming Protocol*)

O RTSP é um protocolo de grande importância nos sistemas de IPTV na componente do VoD, pois permite que, em vez do cliente armazenar grandes ficheiros multimédia para depois os reproduzir, estes são enviados em *streams* na rede IP em tempo real.

O *Real Time Streaming Protocol* é um protocolo cliente-servidor, orientado à reprodução de conteúdos multimédia, que permite a entrega controlada de conteúdos multimédia através de *streams* sobre a rede IP.

Quando é feito um *streaming*, o conteúdo é "partido" em pequenas partes, para que essas partes possam ser transportadas em pacotes entre servidores, e clientes. Desta forma, o cliente pode estar a vizualizar o primeiro pacote, a descomprimir o segundo, e enquanto recebe o terceiro. Permitindo ao cliente usufruir do conteúdo sem tempo de espera. (Liu, 2000)

Com este protocolo é possível interagir com o servidor de *streams*, como se de um leitor de DVD se tratasse, possibilitando assim os comandos de *play*, *pause*, *fast forward*, *reverse*, *record*.

O RTSP establece e controla *streams* contínuas de áudio e vídeo entre servidores de contéudos multimédia, e clientes. Um servidor de conteúdos multimédia, fornece serviços de reprodução, ou gravação para as *streams* multimédia, enquanto o cliente pede continuamente dados multimédia ao servidor. Este protocolo funciona como um telecomando sobre a rede IP, tornando possível a interacção com o servidor de *streams*, como se de um leitor de DVD se tratasse, com os comandos de *play*, *pause*, *fast forward*, *reverse*, *record*.

Este protocolo opera ao nível da camada de aplicação, mas foi desenhado para trabalhar com protocololos de níveis mais baixos, tais como, RTP e RSVP, para permitir um serviço de *streaming* completo através da internet. Isto é, permite a

escolha dos canais de entrega do conteúdo entre servidor e cliente (quer seja por UDP, *multicast* UDP e TCP), e mecanismos de entrega baseados no RTP. É amplamente escalável, pode funcionar para muitos clientes em *multicast*, como também com apenas para um cliente em *unicast*.

Encontra-se definido na RFC2326 que foi publicado em Abril de 1998. (Schulzrinne, Rao, & Lanphier, Real Time Streaming Protocol (RTSP) RFC2326, 1998)

O objectivo do RTSP é fornecer os mesmos serviços em *streams* de áudio e vídeo, que o http fornece para texto e gráficos. Foi desenhado para ter uma sintaxe e operações, compatíveis com grande parte das extensões aplicadas ao HTTP.

No RTSP, cada apresentação e *stream* de multimédia é identificada por um rtsp url. A apresentação global e as propriedades do conteúdo são definidas num ficheiro com a descrição da apresentação, que contém o tipo de codificação, língua, RTSP urls, *destination address*, *ports*, entre outros parâmetros. O ficheiro com a descrição da apresentação pode ser obtido por um cliente através do protocolo HTTP.

Apesar de existirem semelhanças entre o RTSP, e o HTTP, são diferentes em muitos aspectos. Primeiro, o HTTP é um protocolo *stateless* (não é orientado à sessão), o servidor de RTSP é *statefull* (mantém o estado das sessões), para poder correlacionar o pedidos RTSP com a *stream*. Segundo, o HTTP é basicamente um protocolo assimétrico, no qual o cliente faz pedidos ao servidor, e este responde. Mas no RTSP, tanto o cliente como o servidor podem efectuar pedidos.

Actualmente, os serviços e as operações são suportados através dos seguintes métodos:

OPTIONS

O cliente ou o servidor designa as opções que pode aceitar.

Exemplo:

```
C->S:  OPTIONS * RTSP/1.0
       CSeq: 1
       Require: implicit-play
       Proxy-Require: gzipped-messages

S->C:  RTSP/1.0 200 OK
       CSeq: 1
       Public: DESCRIBE, SETUP, TEARDOWN, PLAY, PAUSE
```

DESCRIBE

O cliente pede a descrição da apresentação, ou o conteúdo identificado pelo URL no servidor.

Exemplo

```
C->S: DESCRIBE rtsp://server.example.com/fizzle/foo RTSP/1.0
      CSeq: 312
      Accept: application/sdp, application/rtsl, application/mheg

S->C: RTSP/1.0 200 OK
      CSeq: 312
      Date: 23 Jan 1997 15:35:06 GMT
      Content-Type: application/sdp
      Content-Length: 376

      v=0
      o=mhandley 2890844526 2890842807 IN IP4 126.16.64.4
      s=SDP Seminar
      i=A Seminar on the session description protocol
      u=http://www.cs.ucl.ac.uk/staff/M.Handley/sdp.03.ps
      e=mjh@isi.edu (Mark Handley)
      c=IN IP4 224.2.17.12/127
      t=2873397496
      a=recvonly
      m=audio 3456 RTP/AVP 0
      m=video 2232 RTP/AVP 31
      m=whiteboard 32416 UDP WB
      a=orient:portrait
```

ANNOUNCE

Quando enviado do cliente para o servidor, o *ANNOUNCE* envia a descrição da apresentação dum conteúdo multimédia representado por um url, para o servidor. No sentido inverso, o servidor actualiza a descrição da sessão em tempo real no cliente.

Exemplo:

```
C->S: ANNOUNCE rtsp://server.example.com/fizzle/foo RTSP/1.0
          CSeq: 312
          Date: 23 Jan 1997 15:35:06 GMT
          Session: 47112344
          Content-Type: application/sdp
          Content-Length: 332

          v=0
          o=mhandley 2890844526 2890845468 IN IP4 126.16.64.4
          s=SDP Seminar
          i=A Seminar on the session description protocol
          u=http://www.cs.ucl.ac.uk/staff/M.Handley/sdp.03.ps
          e=mjh@isi.edu (Mark Handley)
          c=IN IP4 224.2.17.12/127
          t=2873397496 2873404696 2873404696
          a=recvonly
          m=audio 3456 RTP/AVP 0
          m=video 2232 RTP/AVP 31

 S->C: RTSP/1.0 200 OK
       CSeq: 312
```

SETUP

O cliente pede ao servidor para reservar os recursos para uma *stream*, e dar início a uma sessão.

Exemplo:

```
C->S: SETUP rtsp://example.com/foo/bar/baz.rm RTSP/1.0
      CSeq: 302
      Transport: RTP/AVP;unicast;client_port=4588-4589

S->C: RTSP/1.0 200 OK
      CSeq: 302
      Date: 23 Jan 1997 15:35:06 GMT
      Session: 47112344
      Transport: RTP/AVP;unicast;
        client_port=4588-4589;server_port=6256-6257
```

PLAY

O cliente pede ao servidor para que este envie a *stream* alocada através do SETUP, enviado previamente.

Exemplo:

```
C->S: PLAY rtsp://audio.example.com/meeting.en RTSP/1.0
       CSeq: 835
       Session: 12345678
       Range: clock=19961108T142300Z-19961108T143520Z

 S->C: RTSP/1.0 200 OK
       CSeq: 835
       Date: 23 Jan 1997 15:35:06 GMT
```

***PAUSE*:**

O cliente suspende temporariamente a visualização da *stream* que esta a ver, sem que os recursos alocados no servidor se libertem.

Exemplo:

```
 C->S: PAUSE rtsp://example.com/fizzle/foo RTSP/1.0
       CSeq: 834
       Session: 12345678

 S->C: RTSP/1.0 200 OK
       CSeq: 834
       Date: 23 Jan 1997 15:35:06 GMT
```

RECORD

O Cliente inicia a gravação de um intervalo de tempo da *stream* de acordo com a descrição da sessão.

4. *CODECS*

Neste capítulo são descritos os *codecs* mais usados nos sistemas de IPTV, para tal, é feita uma introdução sobre o sistema visual do ser humano (HVS[15]). É com base nos estudos efectuados neste sistema que os *codecs* são desenvolvidos.

4.1 Conceito de *CoDec*

Segundo a definição, o acrónimo ***codec*** é um diminutivo de ***co***der/***dec***oder, e pode ser qualquer tecnologia que permite codificar/descodificar dados. Os *Codecs* são implementados em *software* ou hardware, ou com a combinação dos dois. Muitas vezes confunde-se o conceito de *codec* com o tipo de codificação, ou mesmo com o formato de compressão. Estes conceitos não devem ser confundidos. O formato é um ficheiro, apenas uma forma de guardar os dados em disco, enquanto o *codec* é uma implementação (tipicamente uma programa), que consegue escrever, ou ler o formato dos ficheiros a (ele) associados.

Relativamente à forma como descretizam as fontes analógicas, os *codecs* podem ser de dois tipos: *codecs* sem perdas (*lossless codecs*) ou com perdas (*lossy codecs*).

Os *lossless* conseguem codificar *streams* de vídeo sem perdas, não aplicam qualquer tipo de compressão ao conteúdo codificado. Este tipo de *codecs* é usado sempre que se pretende ficar com toda a informação do conteúdo original, o que é útil no caso de ser feita uma edição em estudio do conteúdo. Claro que o espaço de armazenamento de um conteúdo com um *codec lossless* é muito superior à de um *codec lossy*, mas como a tendência do preço dos dipositivos de armazenamento é de descer, e a largura de banda das redes IP é de subir, os *codecs lossless* são ser cada vez mais usados.

[15] HVS - *Human Vision System*

Os *lossy* codificam as *streams* com perdas na fase amostragem e quantização, de seguida, aplicam compressores de vídeo e áudio, para otber ficheiros muito pequenos, mas com tempo alargado de conteúdo. Certo é que estes *codecs* não são ideais para futuramente fazer edição de vídeo, mas conseguem ocupar pouca largura de banda na sua transmissão, e reduzir o custo de armazenamento.

Todos os *codecs* são implementados com base no estudo do sistema de visão humana, aproveitando-se das suas limitações para conseguirem retirar informação à imagem captada, considerada irrelevante pelo sistema de visão humana.

4.1.1 Sistema de Visão Humana

O sistema de visão humano é composto por dois olhos ligados ao cérebro, através dos nervos ópticos.

O olho humano é um órgão em forma de esfera com aproximadamente 2,5cm. É composto pela íris e pela retina. (figura XX) (HVS)

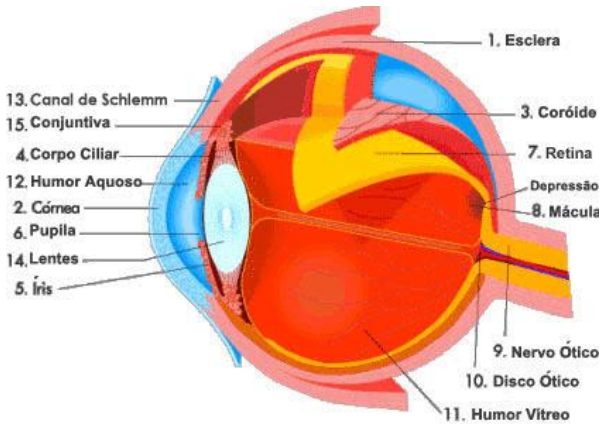

figura 12 - Olho Humano

O modo de funcionamento do olho humano é similar ao de uma máquina fotográfica, isto é, a luz começa por atravessar a córnea (lente), passa pela íris (diafragma), que é responsável por regular a quantidade de luz que entra

dentro da pupila. De seguida é focada pelo cristalino, e projectada na retina (HVS).

A retina é uma membrana fina que se encontra na parte mais interna do olho. Nesta encontram-se células fotoresceptoras denominadas os bastonetes, e os cones.

Segundo a teoria de Thomas Young (Físico Inglês) de 1802, o olho tem três tipos de receptores, cada um deles é sensível a uma parte do especto. Estes receptores denominam-se os cones. (Britannica, 2010)

Os bastonetes são responsáveis por detectar a luminusidade.

Existem certa de 125 milhões de bastonetes e cones dentro da retina, numa relação de 18:1, sendo os bastonetes os mais numerosos. Na figura abaixo, podemos ver um cone entre dois grupos de bastonetes.

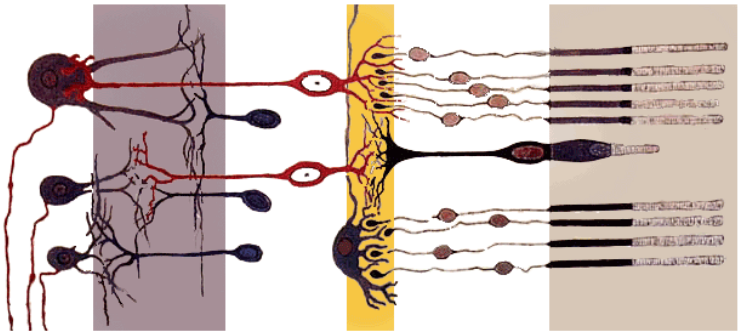

Figura 13 - Estrutura da retina 1 cone entre dois grupos de bastonetes(Britannica, 2010)

Os bastonetes são capazes de funcionar com pouca luz (conseguem detectar 1 único fotão), criando as imagens a preto/branco, quando existe pouca luz. Os cones só entram em acção com níveis de luz superiores, e são responsáveis pela percepção das cores e dos detalhes dos objectos.

O nervo óptico tem a capacidade de realizar diversas funções motoras e sensitivas. Este nervo capta as informações através dos cones, e dos bastonetes presentes na retina estimulados pela luz reflecida nos objectos visualizados. As informações visuais são captadas e enviadas ao lóbulo occipital do cérebro para

as áreas 17, 18 e 19, responsáveis de processar esta informação, gerando resultados de cor, forma, tamanho, distância e noções de espaço. (Wikipedia, 2010)

Tendo em conta a complexidade do sistema de visualização humana, são desenvolvidos modelos que permitem simplificar a sua complexidade, permitindo a criação de *codecs* que tiram partido das "falhas" do olho humano para aumentar a compressão dos conteúdos.

O olho humanano é mais sensível a variações bruscas de luminosidade (contrastes), do que à luminosidade absoluta. Os *codecs* nas situações de grandes contraste tiraram partido das limitações do olho humano, e retira a informação considerada não perceptível, enquanto o olho é iludido pelo contraste. O olho humano é mais sensível a frequências espaciais mais baixas, nas mais altas, actua como um filtro passa alto.

4.2 Codecs

As técnicas de codificação e compresssão de vídeo desempenham um papel importante no mundo das telecomunicações, e sistemas multimédia, nos quais a largura de banda continua a ser um recurso valioso. A principal função destas técnicas é reduzir a quantidade de informação necessária para uma sequência de imagens, sem degradar a qualidade do ponto de vista do sistema de visualização humana.

Na década de 70 é feita a primeira codificação de sinais de televisão, com codificação PCM[16], com um *bit rate* de 140Mbps. Mas o elevado *bit rate* restringe este tipo de codificação a sistemas de distribuição de TV e estúdios de edição. A televisão digital torna-se actractiva para distribuição por satélite, quando os sistemas evoluem para o método de codificação de canal TDMA[17]. Sistemas experimentais nos anos 80 usam *bit rates* de 45 Mbps para o *standard* NTSC, e 60 Mbps para o *standard* PAL.

[16] *Pulse Code Modulation*
[17] *Time Division Multiple Access*

No final dos anos 80, a recomendação do ITU-T para sistemas de videoconferência, é a combinação de codificação entre *frames* com DPCM[18] para que o atraso de codificação fosse mínimo, e de seguida DCT[19] aplicada a cada *frame* diferencial. Surge então o H.261 (Figura 14). (Ghanbari, 2003)

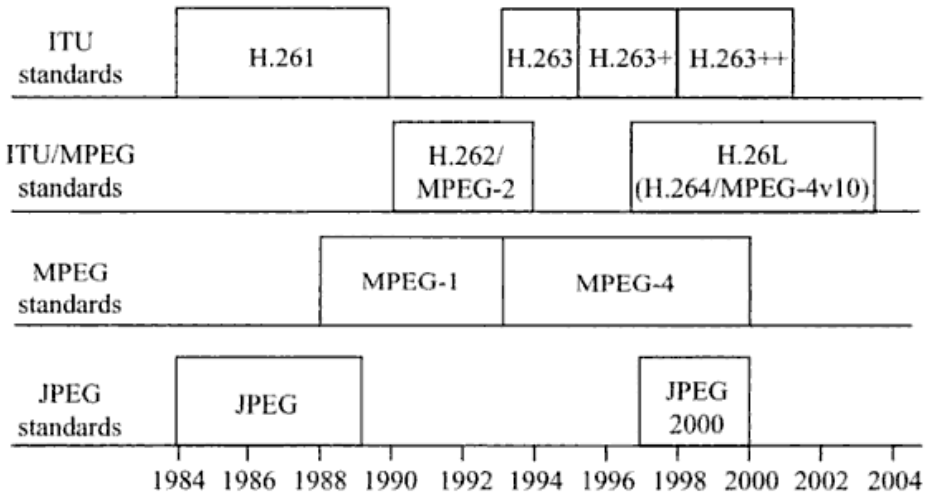

Figura 14 - Evolução dos *Codecs*

4.2.1 H.261 / MPEG1

O *CoDec* H.261 surge no final dos anos 80, como o primeiro *codec* implementável com base na tecnologia existente, usado para efectuar vídeo-conferências sobre RDIS (2x64kbps), ou circuitos primários (30x64kbps). Considera-se que o *bit rate* aceitável para vídeo-conferência ronda os 384kbps, para uma qualidade superior com uma taxa de fps[20] maior 1 Mbps. Este *codec* é denominado **p** x 64 kbps, no qual o **p** era o número canais de 64kbps para o *codec*, o **p** podia ter valores entre 1 e 30.(Ghanbari, 2003)

A definição final do H.261 fica completa em 1989, e categorizado um *standard*.

[18] *Differencial Pulse Code Modulation*
[19] *Discrete Cosine Transform*
[20] *Frames* por segundo

Este *codec* é o primeiro a estabelecer uma unidade básica de processamento, denominada macrobloco, este conceito é a base de todos os *codecs* actuais. Cada imagem (*frame*) é dividida em macroblocos de 16x16 pixeis, codificados num vector bidimensional de 16x16 pixeis de luminância (valor de luminância de cada pixel), e mais dois vectores crominância de 8x8, usando amostragem 4:2:0 e YCbCr.

A codificação DPCM entre *frames* reduz a redundância temporal, isto é, apenas as diferenças de *frame* para *frame* são processadas. Cada *frame* "diferença" é dividida em macroblocos de 16x16 pixeis, nos quais é aplicada a DCT de 8x8 pixeis, para reduzir a redundância espacial. Os coeficientes da DCT são calculados analizando os macroblocos em zig-zag.

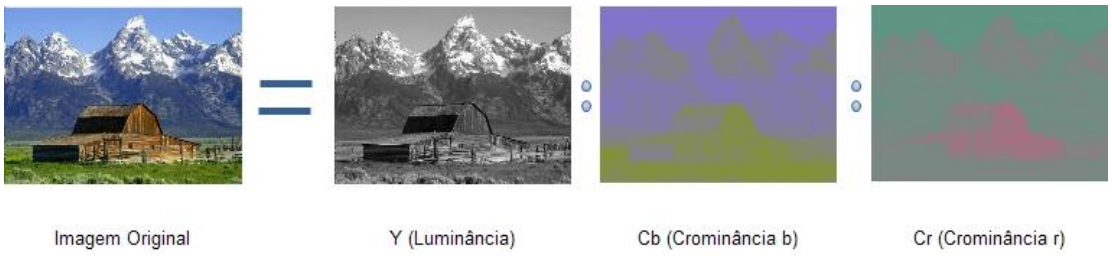

figura 15 - YCbCr

O sucesso do H.261 é um marco importante na história da codificação de vídeo com *bit rates* baixos. No início dos anos 90, o *Motion Picture Experts Group* (MPEG) começa a investigar técnicas de codificação para armazenamento de vídeo em suporte digital (ex: CD-ROM). A intensão é desenvolver um *codec* de vídeo capaz de comprimir vídeos com muito movimento (ex: filmes), em discos rígidos / CD-ROM, mas com uma performance idêntica à dos vídeos de cassetes VHS. O H.261 serve de base à primeira geração do novo *codec* do grupo MPEG (MPEG-1), que consegue atingir o objectivo com um *bitrate* de apenas 1,5Mbps. A evolução deste *codec* chama-se MPEG-1+, que introduz o vídeo entrelaçado. Com este *codec*, os *broadcasters*, que inicialmente estam resistentes ao uso de qualquer tipo de compressão de vídeo, rapidamente adoptam a nova geração do MPEG, denomidado MPEG-2.

4.2.2 H.262 / MPEG2

O MPEG-2 é a nova geração do *codec* que sucede o MPEG-1. A aplicabilidade deste *codec* é tão abrangente que tem um impacto significativo no mundo das telecomunicações. Oferecendo *bit rates* entre 4 a 9 Mbps para vídeo enterlaçado, este é aplicado na difusão de televisão digital terrestre/satélite, tv por cabo, DVDs, entre outros. Em Novembro de 1998 a *OnDigital* no Reino Unido começa a emitir a BBC e a ITV em MPEG-2. A *Sky-Digital* por satélite também muda a emissão para MPEG-2 na mesma altura.

No MPEG-2 o número de *frames* bidireccionais é definido pelo codificador, o único impacto deste parâmetro é o atraso na codificação, e o aumento de processamento, mas com o benefício de um *bit rate* mais baixo.

Devido à flexibilidade na parametrização do GoP, e das *frames* Bidireccionais, o ITU-T considera que este *codec* poderia ser muito usado nas telecomunicações, logo adopta-o sobre o nome de H.262.

A escalabilidade do MPEG-2/H.262 é o seu ponto forte, pois permite extrair de apenas numa *stream*, varias imagens com diferentes resoluções, e com qualidades distintas, o que é muito importante em aplicações nas redes IP.

4.2.3 H.263

Depois de tantos desenvolvimentos no MPEG-1/2 surge novamente a questão de desenvolver um *codec* que transmite vídeo com um *bit rate* abaixo 64Kbps, para a integração de serviços de vídeo-conferência sobre telemóveis, vídeo-vigilância, telemedicina, e tele-escola. Para satisfazer esta necessidade, o grupo MPEG inicia o desenvolvimento de um *codec* com capacidade de codificar, e comprimir vídeo com *bit rates* muito baixos com o nome de MPEG-4. Antes de atingir os valores de *bit rate* pretendidos, surgem novos requisitos, tais como: "*multiviewpoint scenes*", gráficos naturais, e virtuais com se fossem cenas naturais, vídeo interactivo, entre outros. O MPEG-4 não consegue dar resposta a tais requisitos, nem cumprir os *bit rates* inicialmente prometidos. Então o ITU-

T começa a trabalhar num novo *codec*, o H.263 que promete satisfazer os requisitos que o MPEG-4 não consegue.

Este *codec* é uma extensão do H.261, mas utiliza alguns conceitos do MPEG. Tem a capacidade de codificar pequenos vídeos com taxas de refrescamento baixas com *bit rates* entre os 10 e os 64Kbps. (H.263, 1996)

Este *codec* durante a sua evolução, é denominado H.263+, e H.263++. A recomendação deste *codec*, especifica também vídeo de alta definição.

O que permite o H.263 alcançar bit rates mais baixos é a introdução de 4 vectores de movimento por cada macrobloco, que pela primeira vez, apontam para fora do macrobloco. Enquanto os seus antecessores usam apenas um para estimar o movimento da imagem dentro do macrobloco. Usa também compensação de movimento de ½ *pixel*, uma codificação melhorada de tamanho variável, um cabeçalho reduzido, PB-*frames*, que combinam imagens diferentes, codificadas bidireccionalmente, como se fossem uma *frame* diferença normal (D.Gibson, 2001).

Os formatos suportado pelo H.263 são:

Picture Format	Luminance Pixels	Video Decoder Requirements H.261	H.263
SQCIF	128 × 96	Not defined	Required
QCIF	176 × 144	Required	Required
CIF	352 × 288	Optional	Optional
4CIF	704 × 576	Not defined	Optional
16CIF	1408 × 1152	Not defined	Optional

Tabela 3 - H.263 *Standard* Video Picture Formats (D.Gibson, 2001)

4.2.4 H.264/ MPEG-4 Part 10

Como é possível verificar até aqui, os *standards* de codificação de vídeo evoluem com dois nomes, H.26x e MPEG-x. Em 1997, o grupo de trabalhos ISO/IEC MPEG junta-se ao grupo de codificação de vídeo do ITU-T, formando a JVT (*Joint Video Team*), para trabalhar num *codec* de vídeo com um *bit rate*

mais baixo que os anteriores, formando assim o projecto H.26L, no qual L significa *Long-Term Objectives*.

A primeira versão do *standard* é concluída no final do ano 2002, com o nome de H.264 pelo ITU-T, e como MPEG-4 *version* 10 AVC (*Advanced Video Coding*), pelo grupo ISSO/IEC MPEG *group*.

O que destingue este *codec* dos anteriores é a capacidade de tratar as imagens como objectos, e gerar um *bit stream* escalável. Desta forma é possível interagir com o vídeo, como por exemplo, escolher as partes com mais interesse, ou mesmo fundir imagens reais com imagens virtuais. Este processo chama-se *virtual studio*. O MPEG-4 define um método de codificação basedo em modelos de objectos, para codificar objectos simétricos. Com todos estes argumentos o H.264 consegue atingir metade dos *bit rates* dos seus antecessoes (MPEG-2, H.263), para a mesma resolução de imagem sem aumentar a complexidade, ou custo de produção de equipamentos que o suportam. A flexibilidade deste *codec* permite a sua aplicação em cenário de baixa/alta resolução, baixo/elevado *bit rate*, redes RTP/IP, e nos sistemas de vídeo-conferência.

Este *codec* é usado em quase todos os *players* de contéudos multimédia, tais como *Microsoft Silverlight*, *BlueRay Discs*, *Youtube*, *iTunes*, *Adobe Flash Player*, DVB-S/T/C, Conferências, e mais recentemente pelos *browsers* de internet que interpretam HTML5 (ex: *Google Chrome*, *Safari*).

5. QoS/QoE em IPTV

Fornecer serviços de Televisão sobre redes IP até à casa do cliente, incumbe a grande responsabilidade de assegurar um serviço de TV, que proporcione uma experiência de utilização tão boa, quanto os serviços existentes na TV por cabo/satélite. Para assegurar a disponibilidade e robustez do serviço de IPTV, têm que ser criados mecanismos de reserva de recursos, e prioritização de dados através da implementação de políticas de QoS (*Quality of Service*).

Segundo a definição, QoS é a capacidade de fornecer diferentes níveis de prioridade a diferentes aplicações, utilizadores ou fluxos de dados. Estes necessitam de garantias ao nível de *bit rate*, atraso, *jitter*, descarte de pacotes, ou mesmo na taxa de erros. A QoS desempenha um papel importante sempre que a rede não tem capacidade suficiente para satisfazer todos os pedidos, tendo que existir um tratamento diferente para tráfego que realmente tem necessidades especiais, em detrimento de todo o resto.

A generalidade dos fornecedores de internet oferece neste momento um serviço do tipo *best-effort*, isto é, sem mecanismos de QoS que diferenciem os fluxos de dados. Desta forma, todas as aplicações *real-time* (VoIP, IPTV) sensíveis às variações de atraso, e que requerem *bit rates* constantes, não funcionam correctamente, se não existirem políticas de QoS implementadas na rede. As políticas são aplicadas a várias camadas de rede: camada de aplicação, camada de transporte e camada de rede. Enquanto os requisitos da camada de aplicação se centram no *bit rate* ao nível da aplicação ponto-a-ponto, os requisitos da camada de transporte focam-se na latência, *jitter* e perda de pacotes. Finalmente os requisitos da camada de rede centram-se na taxa de perda de pacotes em função do *bit rate* e no intervalo de tempo, entre a perda de dois pacotes consecutivos. (Paul, 2011)

5.1 Mecanismos de QoS

O protocolo IP quando foi definido não foi pensado para aplicações *real-time*, ou com tempos de resposta definidos, mas mesmo assim, na RFC791, foi definido um campo de ToS (*Type of Service*), com o objectivo de classificar a qualidade de serviço na internet. Este campo nunca foi utilizado, o que obrigou o protocolo IP posicionar-se sempre no modelo *best-effort*[21]. Mas com os novos serviços a lançar, novos desafios são lançados sobre o protocolo IP, nomeadamente a exigência de o IP proporcionar mecanismos de QoS. Estes assentam em duas abordagens, uma de serviços integrados (IntServ[22]), e outra de serviços diferenciados (DiffServ[23]), ambas definidas pelo IETF.

IntServ – Integrated Services Architecture and RSVP (Resource Reservation Protocol)

A abordagem do *IntServ*, inicialmente publicada na RFC1633, é fundamentada por um conjunto de RFCs, com o objectivo de implementar uma infra-estrutura para a internet, que possibilita o transporte de áudio, vídeo e dados em tempo real com robustez, sobre as infra-estruturas de rede actuais. A arquitectura *IntServ* visa a garantia de QoS através de mecanismos de reserva de recursos da rede, através do RSVP. Desta forma, os recursos são reservados antes do início da transferência de dados. As aplicações começam por solicitar os recursos necessários para cumprir o SLA definido à rede. Esta solicitação é feita através do protocolo de sinalização RSVP. Nesta solicitação, a aplicação começa por pedir a garantia de QoS à rede através da reserva de recursos ao *"reservation setup agent"*, que de seguida verifica se é possível fazer o controlo de admissão desejado. Se é possível, são feitas as alterações necessárias no classificador de tráfego do *router*, para intrepertar, e classificar o fluxo a aplicar o QoS, que

[21] *Best Effort* – O primeiro pacote a chegar é o primeiro a ser "atendido"
[22] *IntServ – Integrated Services Architecture*
[23] *DiffServ – Differentiated Services Framework*

interage com o escalonador de pacotes para encaminhar o fluxo de dados associado à reserva. (R. Braden D. C., June, 1994)

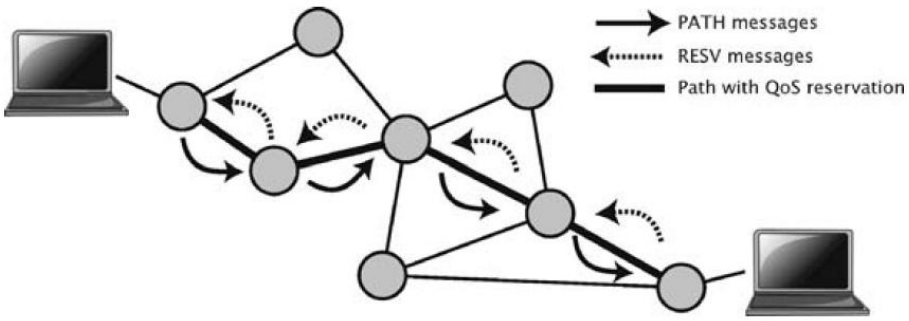

Figura 16 - RSVP - Reserva do caminho com QoS (Braun, 2008)

A tarefa de *"reservation setup agent"*, mencionada na RFC1633 do IntServ, é assegurada pelo protocolo RSVP, publicado 3 anos depois na RFC2205 em 1997. O RSVP não é por si só um protocolo de encaminhamento, mas foi desenhado para funcionar em conjunto com os protocolos de encaminhamento presentes, e futuros de *unicast* e *multicast*. O RSVP faz reservas de recursos, tanto para *unicast* como para *many-to-many multicast,* e consegue adaptar-se dinamicamente a mudanças de rotas (Figura 16 - RSVP - Reserva do caminho com QoS Este protocolo faz reservas para fluxos de dados unidireccionais, o que o transforma num protocolo simplex. É sempre o receptor do fluxo de dados que inicia e mantém a reserva de recursos para o fluxo pretendido, mas no caso de existir um nó da rede que não suporte RSVP, este passa de forma "transparente" por esse nó.(R. Braden L. Z., September 1997)

Nem tudo são vantagens, o IntServ requer um processo de reservas por aplicação, que sobrecarrega memória de cada *router* que tem que armazenar e gerir vários estados de múltiplas reservas. Este problema torna-o impossível de implementar na internet, limitando a implementação a redes de dimensão mais reduzida. Na sequência deste problema surge outra abordagem com o objectivo de garantir QoS, denominada DiffServ.

DiffServ – Differentiated Services Framework

A abordagem DiffServ é outra forma de garantir qualidade de serviço. Não utiliza nenhum mecanismo de reserva de recursos, a qualidade de serviço é assegurada através de mecanismos de prioritização de pacotes. A prioritização é feita através de classes de serviço, assim, os pacotes são classificados, marcados, e processados segundo o campo DSCP (*Differentiated Service Code Point*).

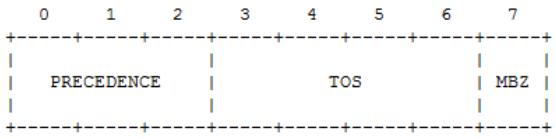

Figura 17 - *ToS Octet of IP Packet*(Almquist, July 1992)

O campo DSCP usa os seis *bits* mais significativos do acteto ToS do pacote IP representado na Figura 17, e os restantes *bits* não são usados. Com esta arquitectura, a ideia é reduzir o nível de processamento dos *routers* ao permitir que todo o tipo de fluxos de dados gerados pelas aplicações, serem agregados em poucas classes de serviço, em função do nível de qualidade de serviço especificado para cada fluxo de dados.

Qualquer fluxo de dados que é admitido numa rede com DiffServ, está sujeito a um processo de classificação, baseada em diversos parâmetros (endereço de origem; endereço de destino; tipo de tráfego; protocolo; etc). De seguida, o fluxo é atribuido a uma classe de serviço marcada no campo DSCP. Após o tráfego ser encaminhado para outro *router*, este intreperta o campo DSCP, e coloca o fluxo na respectiva classe, sem ser necessária nova classificação. Desta forma, a eficiência dos *routers* aumenta por diminuir o processamento do mesmo.(Grossman, April 2002)

5.2 QoS nas redes *wireless*

A norma 802.11 definida pelo IEEE especifica as redes *wireless lan*, que operam na sua maioria, sem mecanismos de QoS. Com o objectivo de "*tapar*" esta lacuna, surge em 2005 a norma 802.11e (*enhanced*), que fornece suporte para serviços sensíveis a atrasos como a voz e o vídeo. Este suporte é conseguido recorrendo à implementação de mecanismos QoS adiante detalhados. Esta norma corresponde também à certificação WMM (*Wi-Fi Multimedia*™) da entidade *Wi-Fi Alliance* (Alliance), responsável por certificar os fabricantes de equipamentos Wi-Fi que implementam a norma 802.11e. (STACEY, 2008)

Como qualquer outra norma 802.X, a norma 802.11 abrange a definição do MAC (*Medium Access Control*), e da Camada Física (*Physical Layer*), neste caso o rádio. Nas primeiras versões do 802.11 é apenas definido um MAC, que interage com três PHY[24], cada um deles com um débito entre 1 e 2 Mbps. Nas redes *wireless* o MAC executa funções que normalmente estão associadas a camadas protoculares mais elevadas, tais como, fragmentação, retransmissão de pacotes e confirmação de entrega (*acknowledges*).

Ao nível de métodos de acesso, a camada de MAC define o DCF (*Distributed Coordination Function*), e o PCF (*Point Coordination Function*). É aqui que a norma 802.11e intervém, adicionando novas extenções às funções de coordenação do MAC, para permitir essencialmente dois métodos de acesso ao *medium*, acessos com e sem contenção. (Prof. Rathnakar Acharya, WLAN QoS Issues and IEEE 802.11e QoS Enhancement, February, 2010)

[24] PHY – *Physical Layer*, sendo neste caso o rádio

Distributed Coordination Function with QoS DiffServ (EDCA)

O método de acesso DCF é o procedimento de transmissão convencional adoptado pelos equipamentos *wireless lan* baseado no CSMA/CA[25]. Nativamente o CSMA/CA funciona da seguinte forma: sempre que uma estação quer transmitir, começa por escutar o meio. Se este está ocupado (no caso de outra estação estar a transmitir), então a estação adia a sua transmissão, mas assim que detecta que o meio está livre começa a transmitir. Isto pode levar a colisões, no caso de duas estações detectarem em simultâneo que o meio está livre, e transmitirem.

Para resolver esta situação, a norma 802.11 usa o mecanismo de *Collision Avoidance*, em conjunto com o mecanismo de *positive acknowledge*, passando a funcionar da seguinte forma: a estação que quer transmitir escuta o meio, se este está ocupado adia a transmissão, se está livre durante um período de tempo, denominado DIFS (*Distributed Inter Frame Space*), então a estação pode transmitir. A estação receptora verifica o CRC[26] da *frame*, e envia uma *frame* de ACK[27]. Se a estação que transmite a *frame* recebe o ACK, significa que não ocorreu colisão, se não recebe o ACK, temos colisão e a estação transmite novamente.

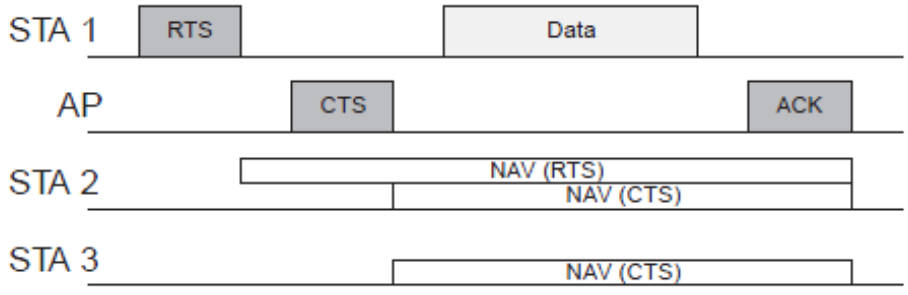

Figura 18 - RTS/CTS *exchange for hidden node protection* (STACEY, 2008)

[25] CSMA/CA – *Carrier Sense Multiple Access with Colision Avoidance*
[26] CRC – *Cyclic Redundancy Code*
[27] ACK – *Acknowledgement*

Pode ainda ocorrer outra situação: as estações não se "ouvem" uma à outra (Figura 18), daí estar definido o mecanismo de *Virtual Carrier Sense* no qual a estação que transmite, envia primeiro um pacote de controlo RTS (*Request to Send*) que inclui a *source*, *destination* e duração da transação, (a duração inclui o tempo de envio do pacote + ACK). A estação receptora responde com um pacote CTS (*Clear to Send*). Todas as estações vizinhas que recebem o pacote de RTS, e/ou CTS, registam no seu *Virtual Carrier Sense indicator* (ou NAV – *Network Allocation Vector*), o tempo que a transação vai durar. Deste modo usam esta informação em conjunto com *Physical Carrier Sense*, enquanto escutam o meio para prevenir colisões.

Como se pode verificar, as funções de MAC são muito complexas, e levam a um desperdício de largura de banda com a informação de controlo, mas evitam colisões que podem levar a disperdícios ainda maiores, com as retransmissões de informação. Apesar de todos estes mecanismos, o método DCF tem limitações que não permitem o uso de mecanismos de QoS, que a norma 802.11e vem colmatar.

Algumas das limitações do DCF são a não existência do conceito de prioridade para dados. Os atrasos ampliam-se com o aumento de estações e aplicações, isto porque, o meio é partilhado pelo número de estações existentes, o que leva à "fome" (*starvation*) de largura de banda. Com os melhoramentos introduzidos pela norma 802.11e, o DCF passa a chamar-se EDCA (*Enhanced Distributed Channel Access*).

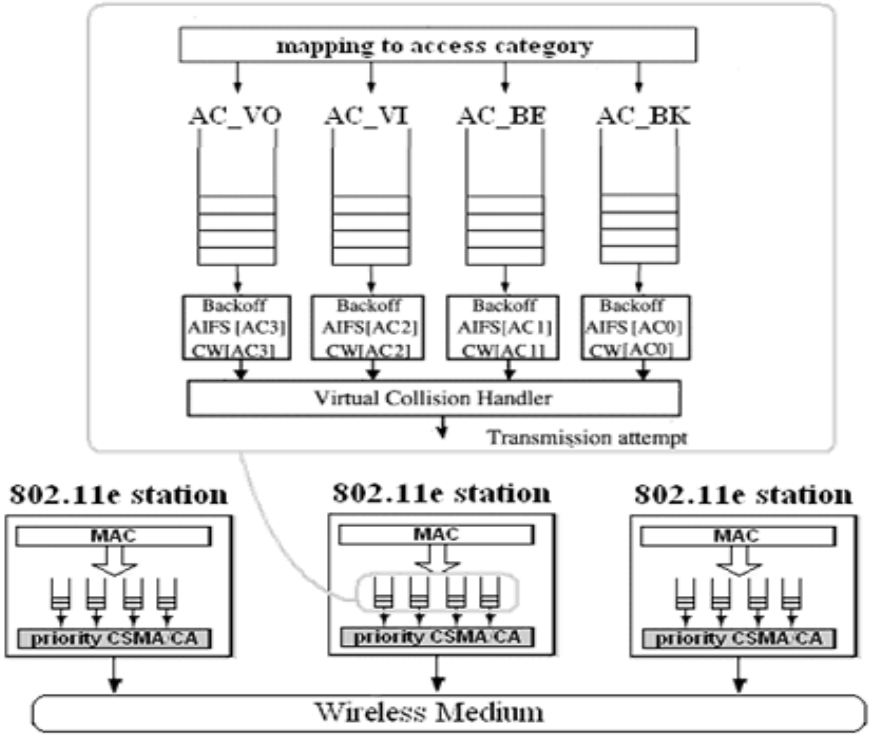

Figura 19 - 802.11e EDCA – (AC) *Access Categories* (Lin, 2009)

Access Category	Description	802.1d Tags
WMM Voice Priority	Highest priority Allows multiple concurrent VoIP calls, with low latency and toll voice quality	7, 6
WMM Video Priority	Prioritize video traffic above other data traffic One 802.11g or 802.11a channel can support 3-4 SDTV streams or 1 HDTV streams	5, 4
WMM Best Effort Priority	Traffic from legacy devices, or traffic from applications or devices that lack QoS capabilities Traffic less sensitive to latency, but affected by long delays, such as Internet surfing	0, 3
WMM Background Priority	Low priority traffic (file downloads, print jobs) that does not have strict latency and throughput requirements	2, 1

Figura 20 - Categorias de acesso 802.11e/WMM(Wi-Fi Alliance, September 1, 2005)

O EDCA permite fornecer diferentes tipos de prioridade a diferentes fluxos de dados, com base nos seus requisitos de QoS (Figura 19). Exitem 4 categorias de acesso predefinidas (Figura 20): voz, vídeo, *best effort*, e *background*. As categorias são parametrizadas por taxas de contenção (CW – *contention*

window), *arbitration inter frame space* (AIFS) e *Transmition Opportunity* (TXOP). A estação que consegue ter a taxa de contenção mais pequena, consegue estar mais tempo no meio, e desta forma realizar múltiplas transferências, enquanto o TXOP durar. O TXOP garante uma quantidade de largura de banda para cada categoria de acesso.

Point Coordination Function with QoS IntServ (HCCA)

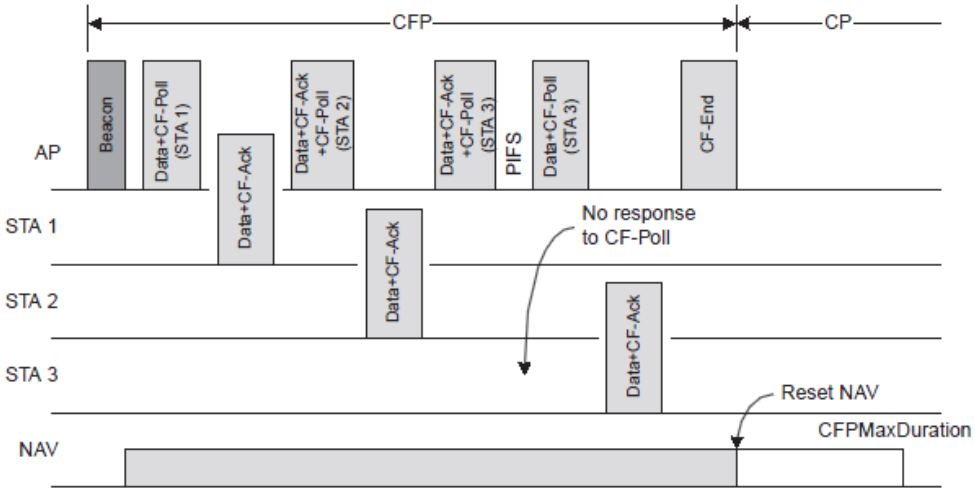

Figura 21 - Transferência de dados durante o CFP(STACEY, 2008)

PFC (*Point Coordination Function*) é um método de transmissão opcional, baseado em *polling*, que é adoptado nas redes *wireless lan* em conjunto com o método DCF. Existe um *access point*, (Figura 21), que desempenha a função de *point coordinator* (PC), estabelece um período sem contenção (CFP[28]) cíclico, no qual só entram as estações que manifestam a intenção de transmitir informação através da mensagem DTIM (*delivery traffic indication message*). O intervalo de tempo definido no CFP é multiplo do número de DTIM. Durante este período sem contenção, o NAV de todas a estações vizinhas, é configurado para o máximo de tempo do CFP das estações que estão a transmitir.(STACEY, 2008)

[28] CFP – Contention Free Period

Este método desperdiça menos largura de banda mas apresenta algumas limitações, uma delas é o *delay* que introduz no tráfego PCF e DCF, porque o período sem contenção (CFP), só ocorre quando existem mensagens de DTIM. Logo, se uma estação está a funcionar em modo PCF, e tem tráfego para enviar, mas o CFP está a terminar, tem de aguardar até ser colocada na *poll* do próximo CFP. Por outro lado, o tráfego que está para ser enviado no método DCF, mas chega durante o CFP tem de aguardar até este terminar, para ganhar de novo acesso ao canal. O algoritmo que distribui as stations no CFP é o *round-robin*, portanto não há qualquer mecanismo de selecção de prioritização de estações, e quando as estações têm acesso ao meio, apenas enviam uma *frame* (inteira ou um fragmento). Todo este conjunto de condicionantes afecta o tráfego sensível aos atrasos (ex: VoIP, *streaming*, tráfego CBR[29]). É de notar, que o método PCF nunca é implementado em larga escala, é muito pouco usado. (STACEY, 2008)

Com os melhoramentos introduzidos pela norma 802.11e, o sistema PCF passa a denominar-se HCCA (*Hybrid Coordination Channel Access*). As melhorias passam por dois aspectos importantes, um deles é a possibilidade de fazer *polling* duma estação, tanto no periodo de contenção, como no período sem contenção. O outro é a atribuição de uma TXOP durante a qual, a estação envia quantas *frames* quiser. São também negociados os parâmetros de QoS com o AP (*Function Coordinator*) (STACEY, 2008)

5.3 QoE

Complementar ao QoS, surge o termo QoE (*Quality of Experience*), uma forma subjectiva de avaliar um determinado produto e/ou serviço, por parte de um utilizador. Este tipo de avaliação é difícil de tratar de forma objectiva, porque é realizada através de questionários e entrevistas, e em contextos de avaliação sobre um serviço/produto que podem ser diferentes. Não obstante, grande parte das

[29] CBR – *Constant Bit Rate*

inovações dos serviços de televisão de nova geração, surgem da análise da QoE por parte dos utilizadores, na qual estes identificam problemas de usabilididade, demoras nos tempos de resposta, e também sugerem o que poderia ser integrado de forma a melhorar/diferenciar o serviço existente. Devido à importância da avaliação subjectiva sobre a qualidade de vídeo entregue ao utilizador final, é necessário entender os factores que a influenciam, para que esta seja avaliada com o máximo de precisão por parte do utilizador final. (Weijun Wang, Yan, Li, & Yang, 2007)

A percepção visual é formada por um complexo conjunto de interacções entre os olhos, e o cérebro, constituintes do sistema de visão humana (HVS). A percepção visual de qualidade é influenciada por dois factores. Um deles é a fidelidade espacial (com que clareza podem ser vistos todas as partes de uma *frame* apresentada na tv, e se existe alguma distorção obvia), e outro é a fidelidade temporal (verifica se o movimento é natural e regular). Existem outros factores que podem influenciar o juízo de qualidade das imagens visualizadas, tais como: o ambiente onde está a ser visualizado o vídeo, o estado de espírito do utilizador, a interacção com o conteúdo visualizado. O critério de avaliação "bom", pode ter diferentes patamares para o utilizador que está a ver o filme passivamente, e para o utilizador que está concentrado nas cenas onde podem existir falhas. Para o utilizador que está a avaliar a qualidade de vídeo, a sua avaliação é mais alta se está num ambiente sem distracções, e confortavelmente sentado com um "bom sistema de som". (Weijun Wang, Yan, Li, & Yang, 2007)

Todos os factores anteriormente referidos dificultam a avaliação da qualidade de vídeo duma forma precisa, e quantitativa.

5.4 Adaptative Streaming

O *adaptative streaming* é um método de transmissão de conteúdos de vídeo que ajusta a qualidade entregue ao utilizador de um sistema de TV sobre IP, existindo ou não mecanismos de QoS que falham, garantindo a melhor qualidade de

experiência ao utilizador. Esta abordagem está a ser adoptada pelos sistemas de IPTV que operam sobre a internet, denominando-se OTT (*Over the Top*). OTT significa que ao contrário dos sistemas tradicionais de IPTV, não necessita de nenhuma rede dedicada, ou de infra-estrutura fornecida pelo operador sendo os conteúdos transportados sobre a rede de internet comum.(Bringuier, 2010)

Como foi referido anteriormente, a maioria dos ISPs fornece um serviço do tipo *best effort* a terceiros, garantindo apenas QoS nos seus serviços, porque a configuração de parâmetros de QoS em todos os *routers* de um operador tem custos consideráveis. E, se o objectivo de um operador de internet TV é difundir os seus conteúdos para todo o mundo, os custos de implementação de QoS em todas as redes para os seus serviços são extremamente elevados. Todos estes factores levaram gigantes como a *Microsoft*, *Apple* e *Adobe* a desenvolver productos de *adaptative streaming* OTT. Os nomes das tecnologias são *Microsoft - IIS Smooth Streaming, Adobe – Flash Dynamic Streaming*, *Apple – http Adaptative Bitrate Streaming*. Todas estas tecnologias funcionam com *unicast streams*. (Bringuier, 2010)

O conceito base do *Adaptative Streaming* consiste em dividir a *source stream* em pequenos segmentos denominados *"chunks"*, e de seguida codificá-los no formato desejado. Os *chunks* podem ter entre 2 a 10 segundos de vídeo. Ao nível do *codec*, cada *chunk* corresponde a um GOP, e começa sempre numa *frame* I, não existindo nenhuma dependência entre *chunks*/GOP passados ou futuros. Desta forma é garantido que cada *chunk* é posteriormente descodificado. Independentemente dos outros, porque a transição entre *chunks* de diferentes *bit rates* é sempre feita na *frame* I.(Zambelli, March, 2009)

A maioria dos sites de partilha de conteúdos de vídeo VoD, incluindo *Youtube*, *Vimeo*, *MySpace* e *MSN Soapbox*, usam o método de *progressive download*, que começa a descarregar todo o conteúdo para uma cache no cliente, enquanto o vídeo é visualizado. O problema ocorre quando o utilizador apenas visiona uma

fracção do vídeo, e na realidade já o descarregou todo. O que representa custos directos na taxação de dados pela CDN (*Content Distribution Network*), que faz a distribuição dos conteúdos até aos ISPs na internet. Esta abordagem não funciona com *live streaming*, porque não existe um ficheiro inteiro do qual se possa fazer um *download*. (Zambelli, March, 2009)

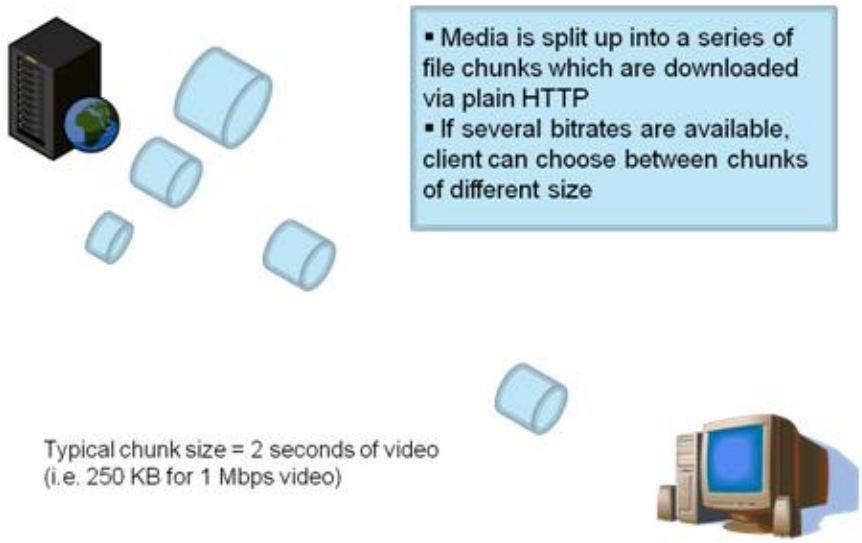

Figura 22 - *Adaptative Streaming* (Zambelli, March, 2009)

No método de *adaptative streaming*, o cliente nunca faz o *download* de um ficheiro com o vídeo inteiro, mas sim um pequeno número de *chunks* configurados em cada implementação. Assumindo que o número de *chunks* que cada cliente pode ter em *buffer* são dois. É exemplificado, a seguir, o funcionamento genérico do sistema sempre que um utilizador visiona um conteúdo.

O cliente quando quer aceder ao conteúdo, (ex: página *player* numa *web*), descarrega um ficheiro com a descrição, e endereços de todos os *bit rates* disponíveis, para efectuar o *download* dos 3 primeiros *chunks* na qualidade mais baixa. O primeiro começa a ser reproduzido enquanto é avaliada a velocidade dos três primeiros *chunks* que foram descarregados, e descarregar o *chunk* com um *bit rate* mais apropriado à ligação que o cliente tem (Figura 22). Sempre que

é comutada o *bit rate* (para mais ou menos qualidade) é descarregado um outro ficheiro de indexação dos próximos N *chunks* em todos os *bit rates*. Outra das grandes vantagens do *adaptative streaming* é a adaptação do *bit rate* do conteúdo, ao *player* com base em diversos parâmetros do equipamento no qual se encontra o cliente, tais como: ocupação de memória/CPU, tamanho da janela (se está ou não minimizada), e a velocidade do acesso.

Figura 23 - Exemplo de *Adaptative Streaming* - IIS *Smooth Streaming* (Microsoft, 2010)

Na Figura 23, está a captura de um exemplo de *adaptative streaming* da *Microsoft*, no qual é possível ver um gráfico no canto inferior esquerdo, que representa a comutação entre diferentes *bit rates* dependendo das condições anteriormente referidas.

6. Arquitectura da Solução

Este capítulo descreve a arquitectura da solução, desenhada com base em todo o conhecimento adquirido da investigação até agora. São abordados os requisitos que uma rede local com distribuição *wireless* deve cumprir, para suportar IPTV *Live*, e as limitações que os APs apresentam com o *multicast*. O cenário de testes assenta numa rede *wireless* de banda larga implementada na freguesia da Memória, concelho de Leiria, para futuramente ser implementado na mesma.

6.1 Introdução

O objectivo desta dissertação é investigar, e estudar o comportamento de um sistema de IPTV nas redes *wireless*. Nomeadamente o comportamento das *streams* de vídeo, através destas redes, que são alvo de um enorme número de perturbações, porque actuam num meio partilhado, sujeito a todo o tipo interferências que é o ar.

Ao longo deste estudo surge a oportunidade de integrar a investigação num caso concreto, o projecto *Memória Online*. Um projecto que já inspirou investigações anteriores, na área de redes de comunicações, tornando este projecto ainda mais aliciante.

Breve descrição sobre o projecto Memória Online

A freguesia da Memória é uma zona rural, com localidades dispersas geograficamente, onde o acesso à internet não chega a todo lado. A distância entre algumas localidades, e a central de ADSL ultrapassa na maioria dos casos os 5km. Para resolver este problema, está implementada uma rede de distribuição *wireless*, com o objectivo de distribuir o acesso à internet às localidades sem acesso, a partir de uma localidade com acesso. O acesso dos clientes à rede de distribuição de internet é via por *wireless*.

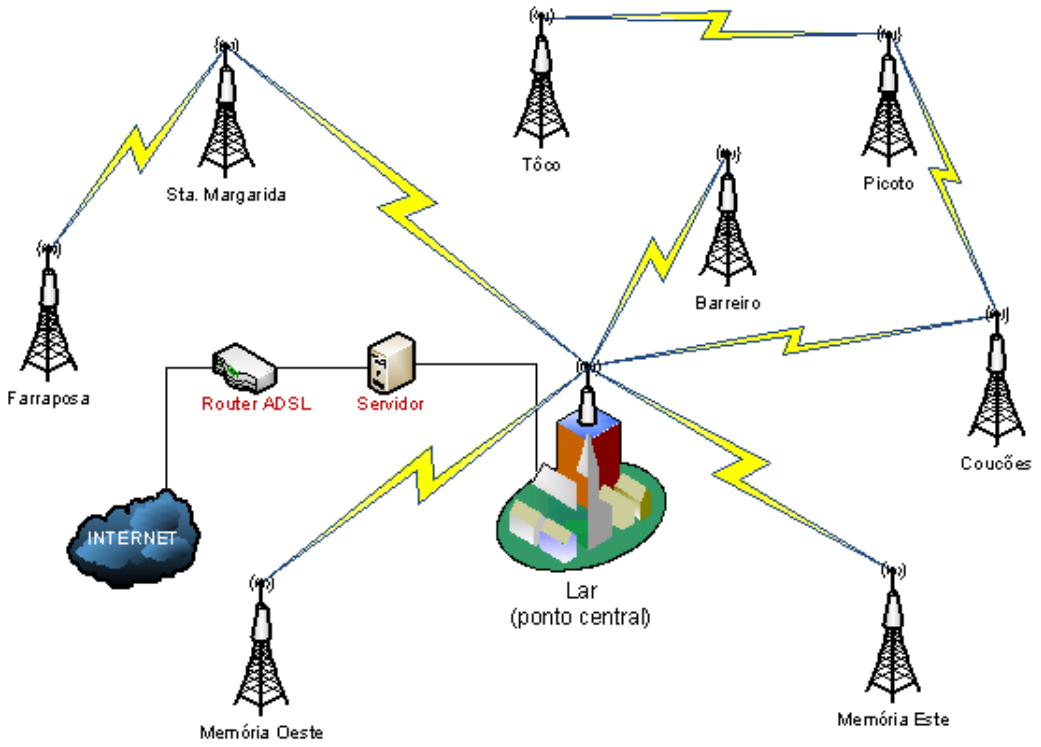

Figura 24 – diagrama da rede de distribuição *wireless* da Memória

A Figura 24 representa a rede de distribuição actual da freguesia da Memória, com um ponto central, o Lar onde chega o acesso adsl, e distribui para toda a freguesia. Os 9 pontos de distribuição usam *routers wireless microtik* e interligam-se com *links* até 300Mbps. A rede de acesso termina recorrendo a equipamentos *wireless* da *ubiquiti*, com *links* até 150 Mbps.

6.2 Cenário de Teste

Este cenário de teste pretende estudar o comportamento das transmissões baseadas em IP *multicast*, neste caso em concreto, *streams* de vídeo *multicast* num ambiente wireless. As aplicações *multicast* baseam-se no conceito *best effort delivery.* O que significa que não há garantia fiável de entrega dos dados a todos os pontos. Nas redes *wireless*, a situação agrava-se. Após o estudo da

norma 802.11-2007, relativamente ao *multicast*, estão identificados aspectos na implementação do MAC (IEEE, 12 June 2007), que levam à degradação da transmissão de *streams multicast*. A norma IEEE 802.11 suporta transmissões *multicast* emitindo-as sem implementar na camada MAC, mecanismo de confirmação de entrega. Significa que o emissor de *multicast* efectua o mecanismo de CSMA/CA antes de enviar os dados, não existindo qualquer tipo de RTS/CTS com ACK, no caso do campo To DS[30] ir a zero. Passo a sitar a secção 9.2.7-"*Broadcast and multicast MPDU transfer procedure*", na página 268 da norma 802.11-2007:

"There is no MAC-level recovery on broadcast or multicast frames, except for those frames sent with the To DS field set. As a result, the reliability of this traffic is reduced, relative to the reliability of individually addressed traffic, due to the increased probability of lost frames from interference, collisions, or time-varying channel properties."

Ainda no mesmo documento na secção 6.1.1-"*Data Service*", na página 51, é referenciado que o transporte de *broadcast*, e *multicast* fazem parte da camada de MAC. E devido às características *wireless medium*, não é possível garantir aos MSDUs[31] do *broadcast* e *multicast*, a qualidade de serviço é fornecida aos MSDU unicast. Passo a citar:

"[…]Broadcast and multicast transport is part of the data service provided by the MAC. Due to the characteristics of the WM[32], broadcast and multicast MSDUs may experience a lower QoS, compared to that of unicast MSDUs."

Todos os aspectos referidos anteriormente podem provocar muitos problemas na transmissão de *multicast* sobre redes *wireless*. No caso de existir uma colisão de uma *frame muticast*, esta é descartada sem retransmissão, porque a camada de MAC, definida na norma 802.11, não tem forma de saber se há ou não colisão por não receber ACK, fornecendo assim um serviço sem garantias. Para resolver

[30] DS – *Distribution System*
[31] MSDUs – *MAC service data unit*
[32] WM – *Wireless Medium*

esta situação, a maioria dos construtores de APs usam uma taxa de transmissão baixa, e fixa (*basic rates* mais baixa) para garantir que as maiorias das *frames* multicast são recebidas pelos seus destinatários com sucesso. Esta é a abordagem da *Microtik* e de outros fabricantes. (Microtik)

Pretende-se construir um cenário no qual é possível ver o comportamento das *streams multicast* através da implementação de um sistema de IPTV, e avaliar os limites da rede, referentes ao *multicast*.

O cenário de teste proposto recorre a equipamentos da mesma marca existentes na rede da Memória, *Microtik* (*Microtik RB433UAH c/RouterOS 4.13*, *routerboard R52n*), e *ubiquiti* (*Bullet5*, *PicoStation c/firmware v3.6.4*). São usados os seguintes equipamentos: um *Microtik*, no qual, liga o servidor de IPTV, dois clientes (*Bullet5* e *PicoStation*), um servidor para as aplicações de IPTV, dois computadores portáteis que se ligam à rede através das *station ubiquiti*, e uma *STB Amino 110*.

Actualmente, na rede da Memória é usada a norma 802.11a/n na rede de *core*/distribuição, e a norma 802.11n na rede de acesso. No entanto, o cenário de laboratório constuído, utiliza a norma 802.11a, por restrição dos equipamentos disponibilizados para os testes. É feito um esforço para usar ferramentas *open source*, que funcionam em *Linux*.

A elaboração do cenário de testes passa por três etapas. A primeira é a configuração do servidor de IPTV, e a selecção/instalação do *software* necessário. A segunda é a configuração e conectividade entre os equipamentos, nos quais são configurados os endereços de rede, as rotas necessárias, e os *links* de rádio. Ainda na segunda etapa, são feitas as medições dos débitos máximos dos *links*, entre o *microtik* e as *stations ubiquiti*. Na terceira etapa são abordadas as configurações necessárias para implementação do *multicast*, bem como, os problemas existentes na implementação do mesmo nas redes *wireless*.

6.2.1 Servidor de IPTV

O servidor do sistema de IPTV é um computador *Dell Optiflex 780*, disponibilizado pela ESTG[33], com as seguintes características:

Hardware	Caracteristicas
CPU	Quad Core Q9550 @2.83GHz
Memória RAM	4 GB DDR3 SDRAM at 1333MHz
Disco	500 GB
Placa da Rede	Integrated Intel® 82567LM Ethernet LAN 10/100/1000

Tabela 4 - Características do Servidor de IPTV

O sistema operativo a instalar é o *Ubuntu release 10.10 maverick*, com a versão *2.6.35-23-generic-pae* do *kernel Linux*, e com a versão do ambiente gráfico *GNOME 2.32.0*. A selecção desta distribuição baseia-se no facto de ser totalmente gratuido, (possui uma enorme comunidade que presta um excelente suporte quando ocorrem problemas), fácil de manobrar, e experiência acumulada nesta distribuição.

Selecção do *software* instalado no servidor

Este servidor é o core do sistema de IPTV, executa as funções de *streaming multicast server*, e *VoD server*. Executa também as funções de monitorização, e teste da capacidade da rede.

Live Streaming* / VoD *Server

Na componente de *live streaming* pretende-se um servidor que captura o contéudo de uma placa, ou de um ficheiro, e efectue o *streaming* para *multicast*. Na componente de VoD pretende-se um servidor que suporta RTSP, com funções de *Trick Play*.

É efectuada uma pesquisa de *software* que cumpre os requisitos exigidos, da qual, resultou a seguinte tabela:

[33] ESTG – Escola Superior de Tecnologia e Gestão de Leiria

Nome	Organização	Funcionalidades	Suporte Multicast	Free	Open Source	Plataforma
Live555 Media Server	Live 555(Live 555)	Servidor de RTSP com trick play	Não	Sim	Sim	Linux/ Windows/ FreeBSD/ MacOS
Darwin Streaming Server	Apple (Apple)	Servidor de RTP/RTSP	Sim	Sim	Sim	Linux/ Windows/ MacOS
Windows Media Server	Microsoft	Player/Servidor de sreaming	Sim	Não	Nao	Windows
VLC	VideoLan(VideoLan)	Player/Servidor de streaming	Sim	Sim	Sim	Linux/ Windows/ FreeBSD/ MacOS/ Solaris

Tabela 5 - Servidores de *Streaming*

Testadas as aplicações que constam na Tabela 5 - Servidores de *Streaming*, as que merecem maior destaque são o VLC, e o Live555, porque são de fácil utilização, produzem *streams* com boa qualidade, e têm suporte para um grande leque de plataformas.

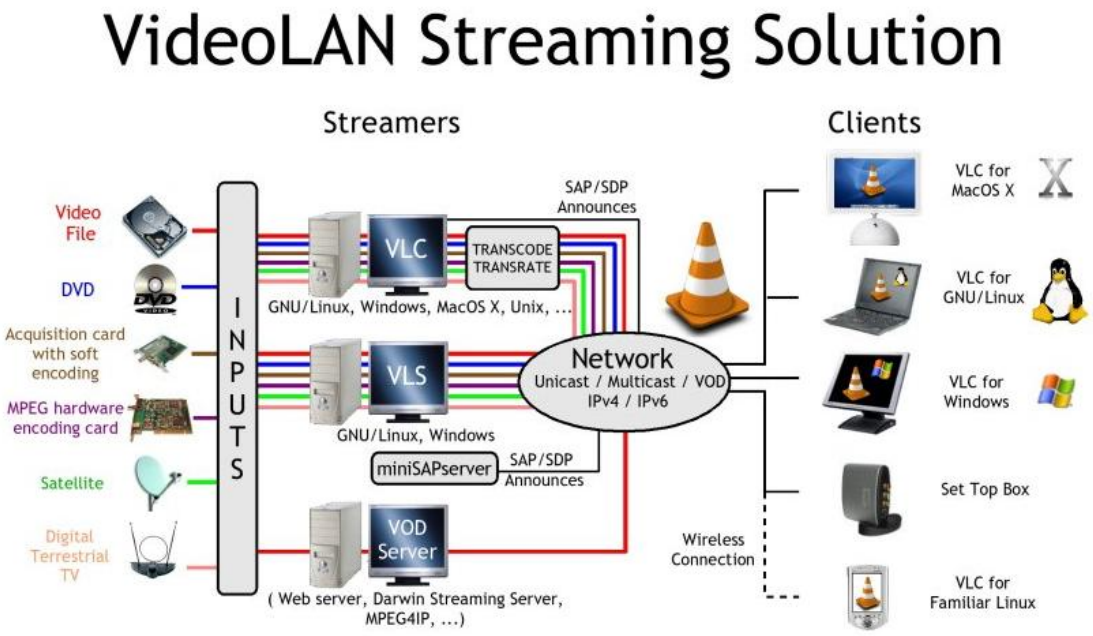

figura 25 - VLC *streaming solution*

O VLC destaca-se pelo número de funcionalidades, e de suporte para um grande número de protocolos de transporte, e *codecs* (figura 25). Permite capturar conteúdos de diversas fontes, tais como: ficheiros de vídeo, captura directa de vídeo/áudio de placas de vídeo através do V4L (*Video for Linux*), ou mesmo dum leitor de DVD. Desempenha a função de *transcoder* entre protocolos/*codecs* distintos, o que é bastante útil para o cenário em questão. O VLC também fornece as ferramentas de *debugging (Media information, Codec Information, Messages)*, uma ajuda valiosa quando é necessário fazer *troubleshooting.* (VideoLan)

O projecto VLC disponibiliza um *plugin* para integrar no *firefox*, permite imbutir vídeo numa página *Web*, disponibiliza uma APIs de desenvolvimento para Java e C++, e suporta interacção por linha de comandos ou *telnet*. Este *software* também fornece a função de *player*, e suporta grande parte do *codecs* existentes.(VideoLan)

O *Live555* destaca-se pela sua simplicidade, e pela boa qualidade que oferece no *streaming* VoD, é um servidor de RTSP completo. Pode fazer o *stream* de diversos tipos de ficheiros multimédia, para tal, basta executar o programa na pasta onde estão os ficheiros. Este servidor pode transmitir múltiplas *streams* concorrentes em simultâneo, com as funcionalidades de *trick play* nos ficheiros que sejam MPEG *transport stream*.

Por todas as razões anteriormente mencionadas, o VLC e o Live555 são as aplicações seleccionadas para integrar no servidor de IPTV.

Ferramentas de monitorização e medição de rede

Como ferramenta de monitorização de rede é seleccionado o analisador de protocolos *Wireshark,* uma aplicação multi-plataforma, na qual é possível monitorizar todo o tráfego que chega a uma *interface* de rede, desde que esta suporte o modo promíscuo. Outra vantagem desta aplicação é possibilidade de

construir filtros, que ajudam a filtrar a informação a capturar, ou capturada.(Wireshark)

A ferramenta de medição escolhida foi o *Iperf*, porque consegue medir a velocidade/*jitter*/perda de pacotes, com o transporte de dados UDP e TCP entre dois pontos da rede. Para o efeito, inicia-se o *iperf* como servidor num dos pontos, e como cliente no outro. Existe também uma GUI[34] escrita em *java* denominada *JPerf*. Este ferramenta é *free* e *open source*, e, disponível no *sourceforge.*

6.2.2 Cenário Proposto

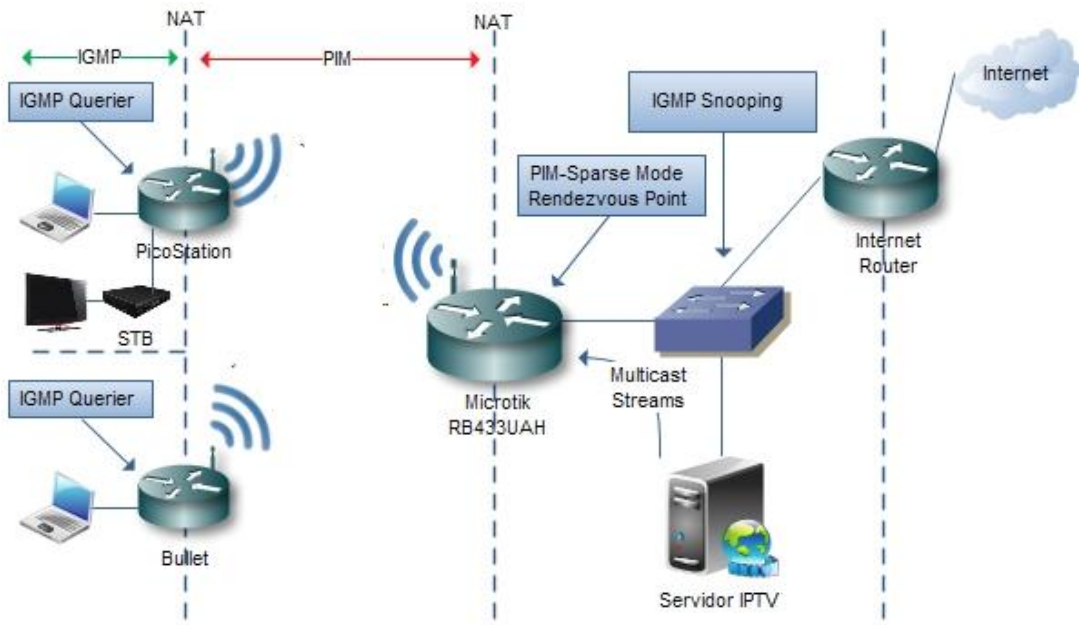

Figura 26 - Cenário Proposto

A arquitectura do cenário, na Figura 26, tem como objectivo testar o comportamento do *multicast* nas redes *wireless*, bem como as suas limitações, como base de referência, o ponto central da rede *wireless* implementada no Lar da freguesia da Memória.

[34] GUI – *Graphical User Interface*

O Servidor IPTV desempenha as funções de *streaming multicast server* através do VLC. Desempenha as funções de monitorização de rede com o *Wireshark*, e de medição com o *IPerf+JPerf*. Este servidor é o ponto de origem das *streams multicast*. O *switch* que o liga ao *Microtik* deve suportar afunção de IGMP *snooping*, para que o *multicast* não "inunde" todas as portas do *switch*, degradando o desempenho do mesmo, e por conseguinte da rede.

O *Microtik* tem um papel importante a desempenhar, porque é o ponto central da rede que desempenha a função de IGMP *Querier* na *interface* de rede que fica do lado do Servidor de IPTV, para identificar as fontes de *multicast*. Para o *multicast* propagar-se no segmento de rede *wireless*, o PIM tem que ser configurado. O *rendezvous point* é o ponto mais próximo da *source* de *multicast*, logo, cabe ao *Microtik* assegurar essa tarefa.

Para que o *multicast* se propague até aos pcs dos clientes, os equipamentos *Ubiquiti* têm que suportar PIM, e serem *neighbors* do *rendevous point*. Têm também de assegurar a função de IGMP *Querier* no segmento de rede onde o cliente se liga, para atender os pedidos dos clientes *multicast*.

7. Implementação

Este capítulo destina-se à implementação do cenário proposto no capítulo anterior. A implementação passa por três fases, sendo a primeira a configuração do servidor de IPTV, a segunda a configuração do cenário, testes de conectividade, e medições dos *links*. Por fim a terceira, consiste na configuração do *multicast* na rede, e os respectivos testes.

7.1 Configuração do servidor de IPTV

O servidor de IPTV é o core do sistema de IPTV, executa as funções de servidor de *streaming multicast* com o VLC. As funções de monitorização são executadas com o *Wireshark*, e os de teste de capacidade da rede com o *IPerf*.

Instalação do sistema operativo

A primeira fase é a instalação do *Ubuntu* 10.10 (*maverick*) 32bits. De seguida, efectuam-se todos os *updates* disponíveis para esta distribuição.

Instalação do VLC

A instalação do VLC é feita com base no gestor de pacotes do *Ubuntu* (*Synaptic Package Manager*), que resolve todas as dependências necessárias à instalação, inclusive a instalação dos *codecs*, que até esta versão eram instalados manualmente. A versão instalada é a 1.1.4 de Novembro de 2010.

Instalação do Live555

A instalação do Live555 é bastante simples, descarrga-se o ficheiro para a pasta, no qual estão os vídeos que (queremos) servir em VoD no *Linux*, atribuem-se as permissões de execução, e executa-se o comando *"./live555MediaServer"*. No momento da execução do comando, o servidor de VoD por RSTP está pronto.

Para aceder ao vídeo abre-se o VLC em modo cliente, e digita-se no URL:

rtsp://<IP do Servidor>:8554/<Nome do ficheiro de vídeo>

A versão do *Live555* instalada é a 0.63 de Dezembro de 2010.

Instalação do *IPerf*

A instalação do *IPerf* efectua-se através do gestor de pacotes do Ubuntu. A versão instalada é a 2.0.4-5.

Para instalar o *JPerf*, é necessário a instalação do JRE[35], instalado com o gestor de pacote. O pacote a instalar é o *"default-jre"* com as respectivas dependências. De seguida, efectua-se o *download* do *jperf* no *sourceforge*, descompacta-se, e dão-se as permissões de execução ao ficheiro "*jperf.sh*". Executado o comando"*./jperf.sh*", aparece um GUI simpático que permite usar o *IPerf* de uma forma mais intuitiva.

Instalação do *Wireshark*

A instalação do *Wireshark* é efectuada através do gestor de pacotes do *Ubuntu*. A versão instalada é a 1.2.11.

7.2 Configuração e testes desempenho da rede proposta

Com o servidor configurado, a próxima etapa é a configuração do cenário de rede proposto. Na Figura 27 está representado o cenário a implementar com o devido endereçamento. Seguem-se as configurações base dos equipamentos, que garantem a conectividade de todo o cenário.

[35] JRE – *Java Runtime Environment*

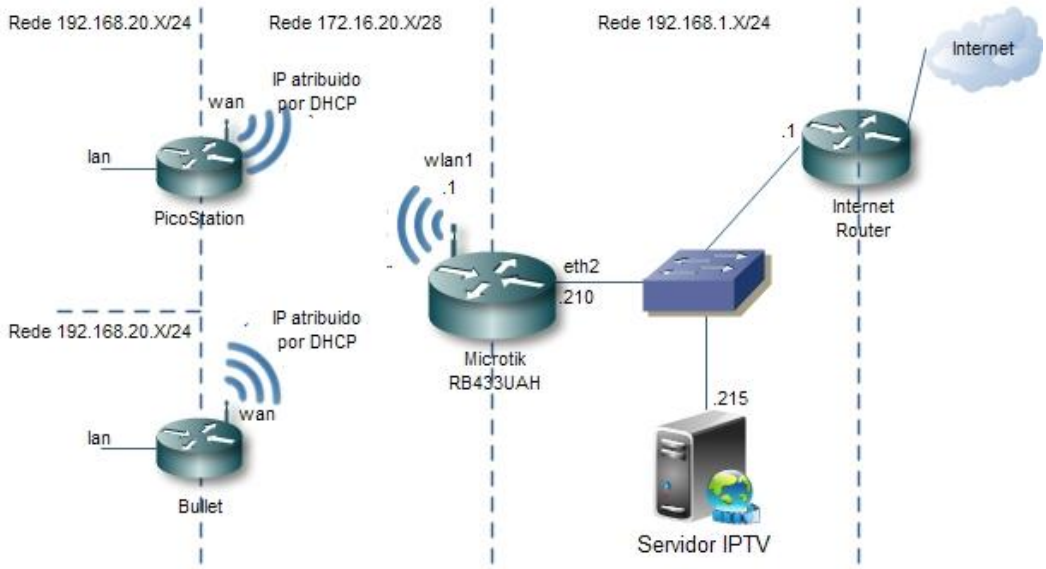

Figura 27 - Cenário de testes

Configuração do *Microtik* RB433UAH

O *Microtik* é configurado através da ferramenta *Microtik Winbox*. O primeiro passo é a configuração dos IPs das portas das *interfaces* de rede(figura 28).

Configuração dos endereços IP em IP -> *Addresses*:

Address:	172.16.20.1/28	Network:	172.16.20.0
Broadcast:	172.16.20.15	Interface:	wlan1
Address:	192.168.1.210/24	Network:	192.168.1.0
Broadcast:	192.168.1.255	Interface:	ether2

figura 28 - Endereços IP configurados no *Mickrotik*

Configuração da *interface wireless wlan1*:

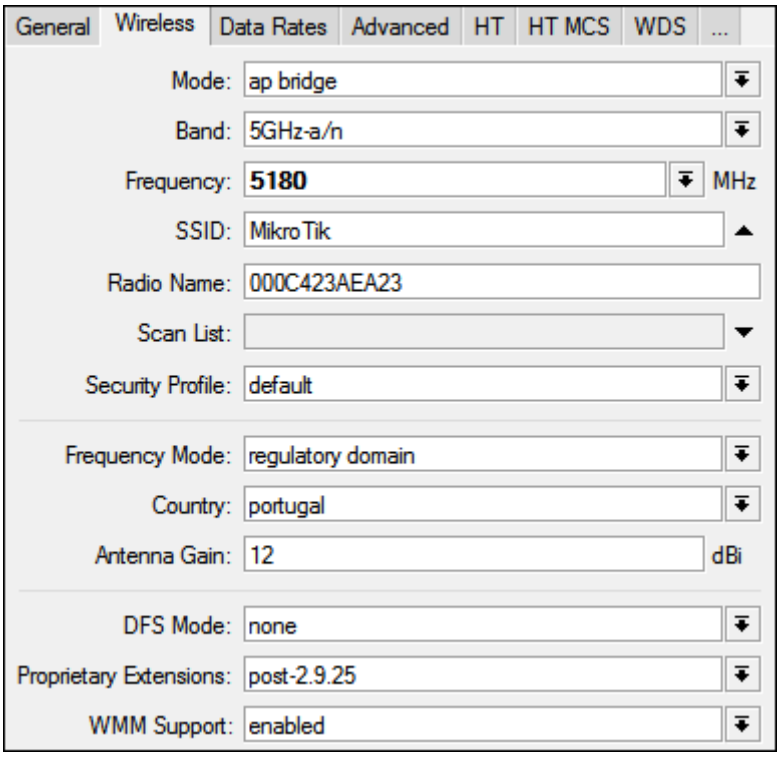

figura 29 - Configurações da *interface wireless* wlan1

Não está configurado qualquer tipo de segurança entre os links, o campo *Security Profile* na figura 29, está com a política *default* que não tem qualquer configuração de segurança.

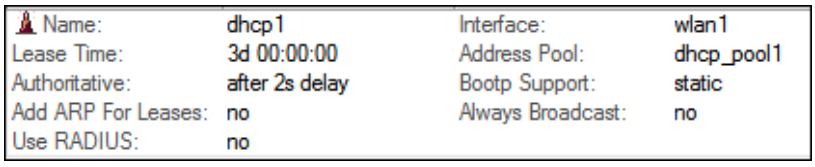

figura 30 - Configuração do servidor de DHCP

O servidor de DHCP configura-se em IP -> DHCP *Server*, para atribuir endereços IP dinâmicos às *stations* (figura 30).

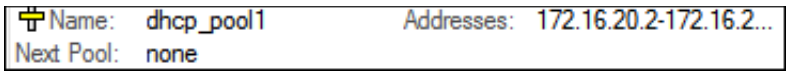

Figura 31 - Configuração de *pool* de endereços IP

A *poll* de endereços a atribuir pelo servidor de DHCP é 172.16.20.2-172.16.20.14,e configura-se em IP -> *Pool* (Figura 31).

Configuração das *Station Ubiquiti* (*Bullet5/PicoStation5*)

As configurações das duas estações são iguais. A *station* é configurada no modo de rede – *router*, como está na rede da Memória. O *link wireless* recebe o endereço por DHCP, e é feito o NAT para a rede interna. Os endereços IP são atribuídos por DHCP na rede interna, sendo a *pool* 192.168.20.2 – 192.168.20.10 (Figura 32).

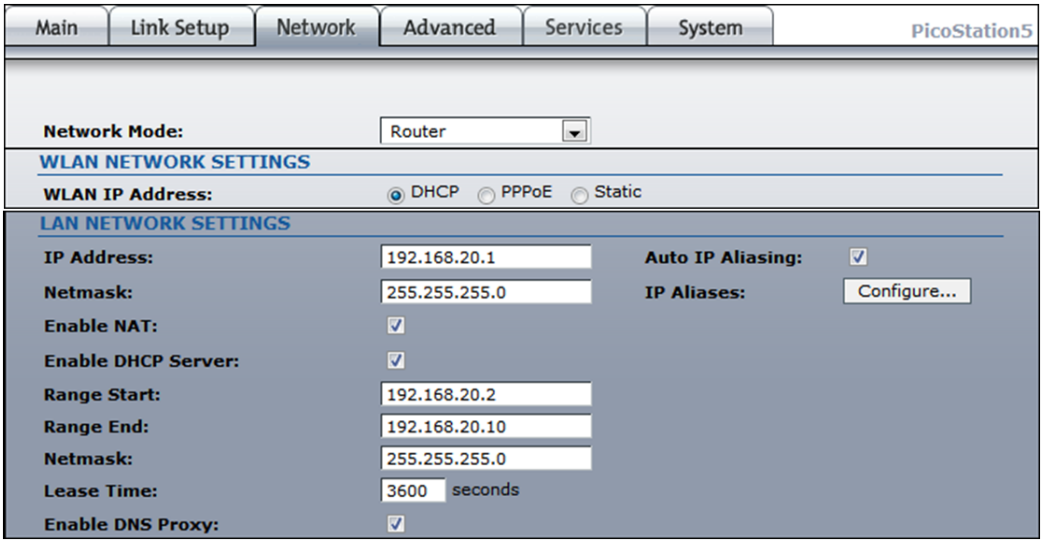

Figura 32 - Configurações de rede do *ubiquiti*

O *link wireless* é configurado no modo *Station*, quando se faz o scan das redes, a rede *wireless* do *microtik* aparece, e é efectuada a ligação à mesma (Figura 33).

MAC address	ESSID	Encryption	Signal, dBm	Noise, dBm	Frequency, GHz	Channel
00:0C:42:3A:EA:23	MikroTik	-	-42	-96	5.18	36

Figura 33 - resultado do *scan* de rede *wireless* do *ubiquiti*

Teste de conectividade

Segue-se o teste de conectividade do cenário. O teste consiste em fazer um *ping*, um *tracert*, e um *nslookup* a um dos endereços de DNS do *Google* (IP - 8.8.8.8),

de dum PC ligado à porta lan de cada uma das *station ubiquiti*. Se todos os comandos obtêm sucesso, toda a rede está operacional.

Resultado do *ping* (sucesso):

```
C:\>ping  8.8.8.8
A fazer ping para 8.8.8.8 com 32 bytes de dados:
Resposta de 8.8.8.8: bytes=32 tempo=67ms TTL=50
Resposta de 8.8.8.8: bytes=32 tempo=54ms TTL=50
Resposta de 8.8.8.8: bytes=32 tempo=53ms TTL=50
Resposta de 8.8.8.8: bytes=32 tempo=52ms TTL=50

Estatísticas de ping para 8.8.8.8:
   Pacotes: Enviados = 4, Recebidos = 4,
         Perdidos = 0 (perda: 0%),
Tempo aproximado de ida e volta em milissegundos:
   Mínimo = 52ms, Máximo = 67ms, Média = 56ms
```

Resultado do *tracert* (sucesso):

```
C:\>tracert 8.8.8.8
A rastrear a rota para google [8.8.8.8]
até um máximo de 30 saltos:
 1    1 ms    <1 ms    <1 ms  192.168.20.1
 2    5 ms     1 ms     1 ms  172.16.20.1
 3    2 ms     3 ms     1 ms  192.168.1.1
.
.
16    55 ms    54 ms    56 ms  google [8.8.8.8]
```

Resultado do *nslookup* (sucesso):

```
C:\>nslookup 8.8.8.8
Servidor:  UnKnown
Address:  192.168.20.1

Nome:    google
Address:  8.8.8.8
```

Conclui-se que toda a rede está operacional.

Teste dos *links wireless*

Pretende-se avaliar a largura de banda disponibilizada entre as *station ubiquiti*, e o *AP Microtik*. Apenas são efectuados testes unidireccionais, e com o protocolo

UDP, porque o tráfego gerado por um sistema de IPTV, apresenta as mesmas características. As *stations* deste cenário funcionam na norma 802.11a, cuja velocidade máxima é de 54Mbps, logo, os testes de capacidade são feitos à velocidade de 54Mbps.

O primeiro teste consiste na medição de velocidade de cada uma das *station* com o AP, averiguando a velocidade máxima obtida, e o valor do *jitter* durante um periodo de 300 segundos. O *IPerf* está configurado no servidor de IPTV, e nos computadores ligados às *stations* com configurações presentes na Tabela 6 – para mais informações sobre os parâmetros do *IPerf*, consultar **Anexo B**.

Parâmetros do IPerf no Servidor IPTV	Parâmetros do IPerf no Cliente
`Iperf -s -u -P 0 -i 1 -p 5010 -f m`	**`Iperf -c 192.168.1.215 -u -P 1 -i 1 -p 5010 -f m -b 54.0M -t 300 -T 3`**

Tabela 6 - Parâmetros do *IPerf* no cenário base – teste 1

Stations Ubiquiti	Largura de Banda Média	Jitter total
Bullet 5	**28,4 Mbps**	**10,933 ms**
PicoStation	**25,3 Mbps**	**14,334 ms**

Tabela 7 - Resultado do teste 1 – cenário base

Na Tabela 7 estão os resultados dos testes individuais efectuados, dos quais conclui-se que a *station Bullet5* tem melhor desempenho ao apresentar uma largura de banda média 3,1 Mbps, superior à *PicoStation*, e um *jitter* 3,401 ms mais baixo, na totalidade do teste.

O segundo teste consiste na medição de velocidade das duas *stations* com o AP em simultâneo, com os parâmetros do teste 2, exactamente iguais aos do teste 1.

Como é esperado, o débito baixa quando as duas *station* operam em simultâneo, mesmo assim a *Bullet5* continua a ter valor de *jitter* total mais baixo relativamente à *picostation (*Figura 34*)*.

Stations Ubiquiti	Largura de Banda Média	Jitter total
Bullet 5	**14,9 Mbps**	**5.817 ms**
PicoStation	**15,6 Mbps**	**14,611 ms**

Figura 34 - resultado do teste 2 – cenário base

7.3 Configuração do *multicast*

Devido ao facto das *stations Ubiquiti* não suportarem PIM, as configurações de *multicast* no cenário inicialmente proposto, têm que ser alteradas.

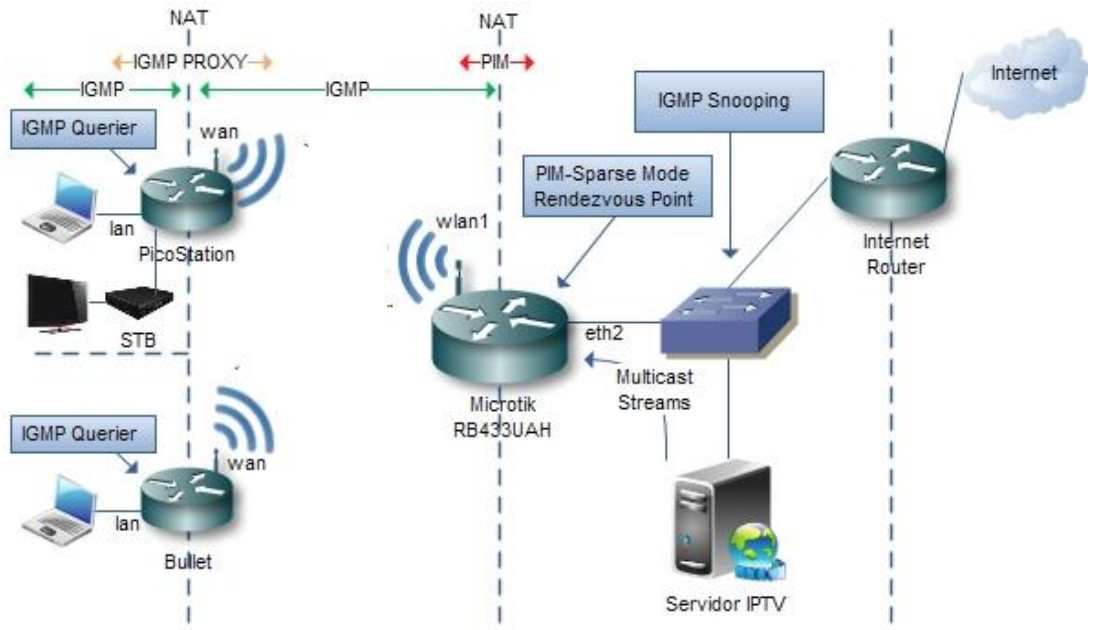

Figura 35 - Cenário de testes *multicast*

A alternativa ao PIM disponível nas *stations* é o IGMP *Proxy*, o que coloca a *interface wlan1* do *Microtik* a assumir o papel de *Querier* nesse segmento. Para efeitos de teste, o comportamento é semelhante, porque o *multicast* tem de ser transportado na rede *wireless* da mesma forma, mas através de um protocolo interno. O PIM continua configurado no *Microtik* para assegurar a tranferência de *multicast*, entre o segmento de rede no qual se encontra o servidor de IPTV, e o segmento *wireless*.

Configuração do *Multicast* no *Microtik*

Os parâmetros de PIM configuram-se em *Routing* -> PIM

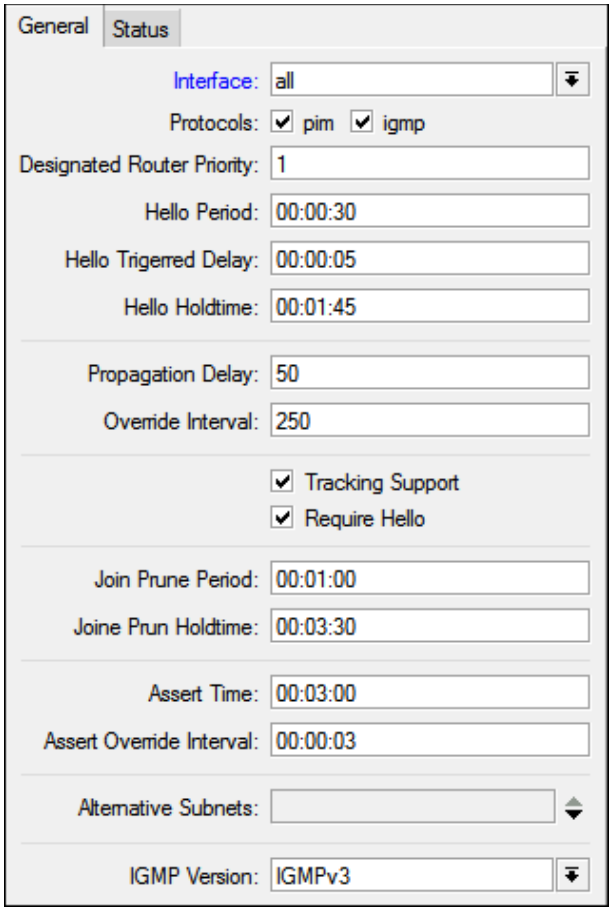

Figura 36 - configuração do PIM no *Microtik*

Todas as *interfaces* (wlan1, eth2) do *Microtik* estão associadas ao PIM, e ao IGMP (Figura 36). Na *interface* eth2, o IGMP está configurado para ocorrer interacção com o IP *multicast* do servidor de IPTV. Na *interface* wlan1 tem que existir IGMP por causa das *station ubiquiti* não suportarem PIM. A configuração do PIM serve para haver propagação de *multicast*, entre as *interfaces* eth2 e wlan1. O *rendezvous point* está configurado na *interface* mais próxima da origem do tráfego (eth2), como se pode ver na Figura 37.

Address:	192.168.1.210	Type:	static
Group:	224.0.0.0/4	Priority:	192
Hash Mask Length:	30	Active Groups:	2
Holdtime:	00:00:00	Timeout:	0

Figura 37 - Configuração do *Rendezvous Point*

Configuração do *multicast* nas *stations ubiquiti*

Como está referido anteriormente, as *stations ubiquiti* não suportam PIM. A alternativa é configurar o IGMP *proxy*, que encaminha os pedidos de IGMP (*lan* <-> *wan*), para o IGMP *Querier* (*interface wlan1* do *Microtik*). A replicação do *multicast* para a rede interna (*lan*) do *ubiquiti* é executada no próprio, ou seja, se estão duas, ou mais máquinas ligadas na *interface lan*, o tráfego *multicast* na *interface wan* não aumenta.

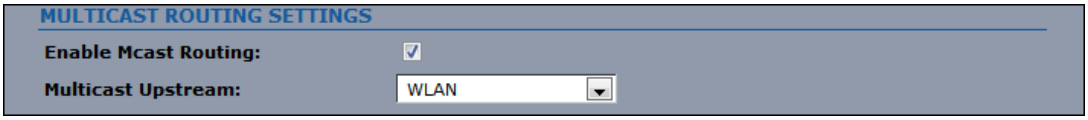

Figura 38 - Configuração do *multicast* nas *stations Ubiquiti*

Para activar o *multicast*, entra-se no menu de configuração *NETWORK*, e na opção *MULTICAST ROUTING SETTINGS* (Figura 38) Activa-se o *multicast routing*, e a *interface* de *upstream* que tem o papel de encaminhar os pedidos.

Só estas configurações não chegam, são necessárias configurações adicionais que não se encontram disponíveis no painel de configuração gráfico, encontram-se na linha de comandos. Entra-se na *station* por SSH, edita-se o ficheiro *"/tmp/system.cfg"* com o VI[36], e configuram-se os seguintes parâmetros:

igmpproxy.upstream.1.netmask=255.255.255.240
igmpproxy.upstream.1.network=172.16.20.0

Com esta configuração a *station ubiquity* está pronta a operar com *multicast*.

[36] VI – Editor de texto

7.4 Testes de IP *multicast*

Através destes testes pretende-se perceber os limites do *multicast* em ambiente *wireless*, devido ao facto, da implementação da norma 802.11 apresentar as limitações referidas no capítulo 6.2.

O primeiro teste consiste em colocar o VLC a produzir *streaming* de um vídeo em *Standard Definition* (720x480 *pixeis*), para validar o funcionamento do *multicast* na rede, bem como a taxa de erros sem outro tráfego na rede.

O segundo teste consiste em avaliar a capacidade da rede transportar *streams* nas resoluções SD (720x480 *pixeis*), HD (1280x720 *pixeis*), e FHD (1920x1080 *pixeis*) sem outro tipo de tráfego na rede.

O terceiro teste consiste em avaliar a capacidade da rede transportar as mesmas *streams*, com injecção de tráfego na rede *wireless*, até ocorrer degradação do vídeo (*frames* perdidas).

7.5 Video usado para os testes

O vídeo usado para teste foi o *Big Buck Bunny*, uma animação produzida pela subsidiária "*Blender Foundation*". O vídeo encontra-se disponível para *download* gratuito em www.bigbuckbunny.org, na resolução *Full*HD (1920x1080). Este vídeo convertido com o VLC para as resoluções SD, HD, e FHD, em *transport stream* com o *codec* de vídeo h264 *part 10* e com o *codec* de áudio *mpeg* AAC (Figura 39).

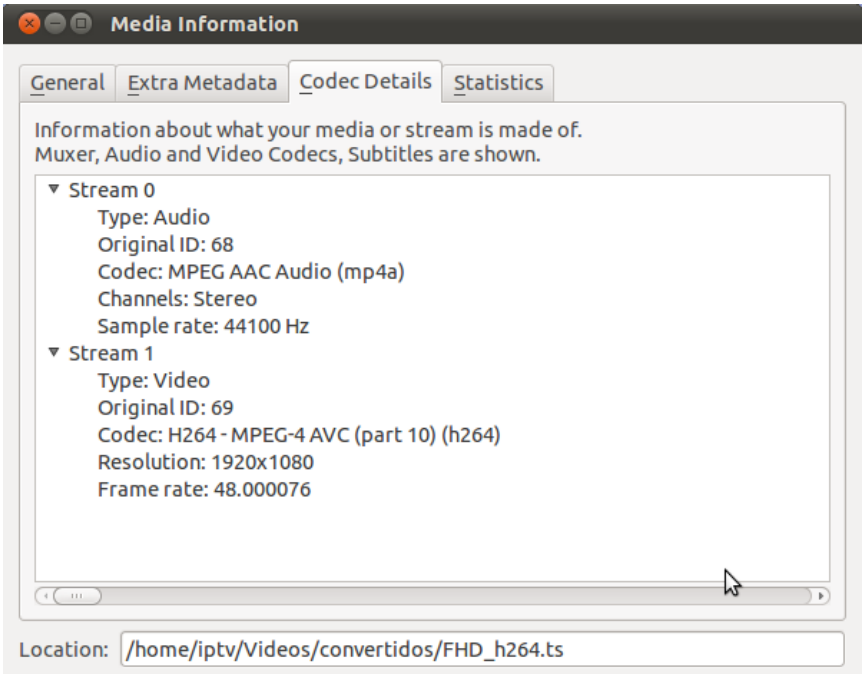

Figura 39 – Exemplo de converção para *transport stream*

O nome de cada *stream* corresponde à conversão do *Big Buck Bunny* para as diferentes resoluções. Na Tabela 8, estão definidos os grupos *multicast* associados a cada um dos vídeos.

Nome do vídeo	Resolução	Bit Rate da codificação variable bit rate	Grupo Multicast
SD_h264.ts	**SD**	**1,5 Mbps**	**224.0.10.1**
HD_h264.ts	**HD**	**2,5 Mbps**	**224.0.10.2**
FHD_h264.ts	**FHD**	**6 Mbps**	**224.0.10.3**

Tabela 8 - Associação do grupo *multicast* aos vídeos

Para o servidor emitir os vídeos em *multicast*, estão configurados os vlc através da *interface* gráfica, no entanto, existe uma limitação na definição do ttl (time-to-live), que sai sempre a 1 independentemente do valor configurado. A solução passa por usar o vlc através da linha de comandos. São então criados *scripts* de *bash* para cada um dos vídeos.

Nome do script	Conteúdo do script
SD_h264.sh	**cvlc -vvv SD_h264.ts --sout '#udp{dst=224.0.10.1:1234}' --ttl 5 --loop**
HD_h264.sh	**cvlc -vvv HD_h264.ts --sout '#udp{dst=224.0.10.2:1234}' --ttl 5 --loop**
FHD_h264.sh	**cvlc -vvv FHD_h264.ts --sout '#udp{dst=224.0.10.3:1234}' --ttl 5 --loop**

Tabela 9 - Conteúdo dos *scripts* para o vlc

A Tabela 9 representa o conteúdo de cada um dos *scripts*. Para mais informações sobre os parâmetros do vlc consultar ANEXO D.

7.6 Teste 1 – verificação do funcionamento do multicast

Este teste consiste na execução do *script* "SD_h264.ts" para o servidor de IPTV emitir no grupo *multicast* 224.0.10.1, o vídeo em SD. O próximo passo é a conexão dos clientes a este grupo através das *station Ubiquiti* com o vlc. É feita a monitorização do *multicast* no *Microtik*. Se o *multicast* está a funcionar correctamente, o *bit rate* de entrada na *interface* eth0 é igual ao de saída na *interface wlan1*.

1º passo: Execução do *script* "SD_h264.sh" no Servidor de IPTV

Group	Source	RP	Join State	Join Reg...	Timeout ...
224.0.0.0	192.168.1.210	192.168.1.210	not joined	unknown	0
224.0.10.1	192.168.1.215	192.168.1.210	not joined	unknown	0

Figura 40 - *Joins* no *Microtik* após execução do *script* SD_h264.sh

Após a execução do *script*, verifica-se na Figura 40, correspondente ao menu *Routing* -> PIM -> *Joins* do *Microtik*, que aparece o grupo *multicast* 224.0.10.1 com valor *join state* = *not joined*. O que significa que ainda não se ligou nenhum cliente.

2º passo: Ligação de um cliente ao grupo *multicast* 224.0.10.1

O vlc cliente conecta-se ao grupo *multicast* 224.0.10.1, através do menu *Media* -> *Open Network Stream*, e coloca o url da *stream*, que neste caso é "udp://@224.0.10.1:1234".

Group	Source	RP	Join State	Join Reg...	Timeout ...
224.0.10.1	192.168.1.215	192.168.1.210	joined	unknown	20

Figura 41 - tabela de *Joins* após a ligação de um cliente

Após a ligação do cliente, a tabela de *Joins* do *Microtik* (Figura 41) passa o Join *State = joined*, significa que o cliente está ligado com sucesso.

Name	Type	L2 MTU	Tx	Rx	Tx Pac...	Rx Pac...
ether1	Ethernet	1526	0 bps	0 bps	0	0
ether2	Ethernet	1522	4.0 kbps	1642.8 k...	1	154
ether3	Ethernet	1522	0 bps	0 bps	0	0
wlan1	Wireless (Atheros 11N)	2290	1639.7 k...	0 bps	150	0

Figura 42 - tabela de *interfaces* do *Microtik* com 1 cliente

3º Passo, Acesso ao grupo 224.0.10.1 nas duas *stations* em simultâneo

Name	Type	L2 MTU	Tx	Rx	Tx Pac...	Rx Pac...
ether1	Ethernet	1526	0 bps	0 bps	0	0
ether2	Ethernet	1522	17.6 kbps	1503.9 k...	2	143
ether3	Ethernet	1522	0 bps	0 bps	0	0
wlan1	Wireless (Atheros 11N)	2290	1499.5 k...	0 bps	138	0

Figura 43 - tabela de *interfaces* do *Microtik* com 2 clientes

Com os dois clientes ligados através das *stations*, o tráfego nas *interfaces* mantém o mesmo comportamento, ou seja, o tráfego que entra na *ether2*, é o mesmo que sai na *wlan1* do *microtik* (Figura 43). As taxas de *frames* perdidas são as seguintes:

Cliente VLC na station	*Frames* reproduzidas	*Frames* Perdidas	Taxa de perda
Bullet 5	**22348**	**1**	**0,00%**
PicoStation	**22345**	**0**	**0,00%**

Tabela 10 - *frames* perdidas em cada cliente c/2 clientes em simultâneo

A taxa de perda de *frames* nos clientes é (quase, ou praticamente) desprezível, não chegando a 1%, o que é bastante bom. Outro aspecto é que, existem menos *frames* perdidas com as duas *station* a transferir a *stream multicast*. Logo, o

multicast está a funcionar, e os testes seguintes podem ser efectuados com os dois clientes em simultâneo.

7.7 Teste 2 – medição de *frames* perdidas sem tráfego na rede

Este teste subsiste na medição de *frames* perdidas em cada um dos grupos *multicast*, correspondentes às resoluções (SD, HD, FHD), sem outro tipo de tráfego na rede.

Teste do vídeo SD_h264.ts no grupo *multicast* 224.0.10.1 c/2 clientes:

Cliente VLC na station	*Frames* reproduzidas	*Frames* Perdidas	Taxa de perda
Bullet 5	**12144**	**0**	**0.00%**
PicoStation	**12109**	**0**	**0.00%**

Tabela 11 - *Frames* perdidas SD

A Tabela 11 corresponde à visualização do vídeo SD_h264.ts disponibilizado pelo servidor de IPTV no grupo *multicast* 224.0.10.1. Não existe nenhuma *frame* perdida em ambos os clientes.

Teste do vídeo HD_h264.ts no grupo *multicast* 224.0.10.2 c/2 clientes:

Cliente VLC na station	*Frames* reproduzidas	*Frames* Perdidas	Taxa de perda
Bullet 5	**12162**	**1**	**0,00822%**
PicoStation	**12144**	**1**	**0,00823%**

Tabela 12 - *Frames* perdidas HD

A Tabela 11 corresponde à visualização do vídeo HD_h264.ts disponibilizado pelo servidor de IPTV no grupo *multicast* 224.0.10.2. Apenas existe uma *frame* perdida, o que não interfere na percepção de qualidade do vídeo.

Teste do vídeo FHD h264.ts no grupo *multicast* 224.0.10.1 c/2 clientes:

Cliente VLC na station	*Frames* reproduzidas	*Frames* Perdidas	Taxa de perda
Bullet 5	**12502**	**122**	**0,97584%**
PicoStation	**12486**	**147**	**1,17731%**

Tabela 13 - *Frames* perdidas FHD

A Tabela 13 corresponde à visualização do vídeo HD_h264.ts disponibilizado pelo servidor de IPTV no grupo *multicast* 224.0.10.3. Neste teste perdem-se muitas *frames*, e após a visualização do vídeo, em simultâneo com o painel de estatísticas do *Microtik*, constata-se que sempre que o *bit rate* vídeo atinge valores próximos de 6Mbps, a perda de *frames* aumenta significativamente, levando à degradação do vídeo, como se pode ver na Figura 44.

Figura 44 - Degradação do vídeo *Full* HD

Sempre que o *bit rate* fica abaixo dos 6Mbps, a imagem não apresenta qualquer problema como se pode ver na Figura 45.

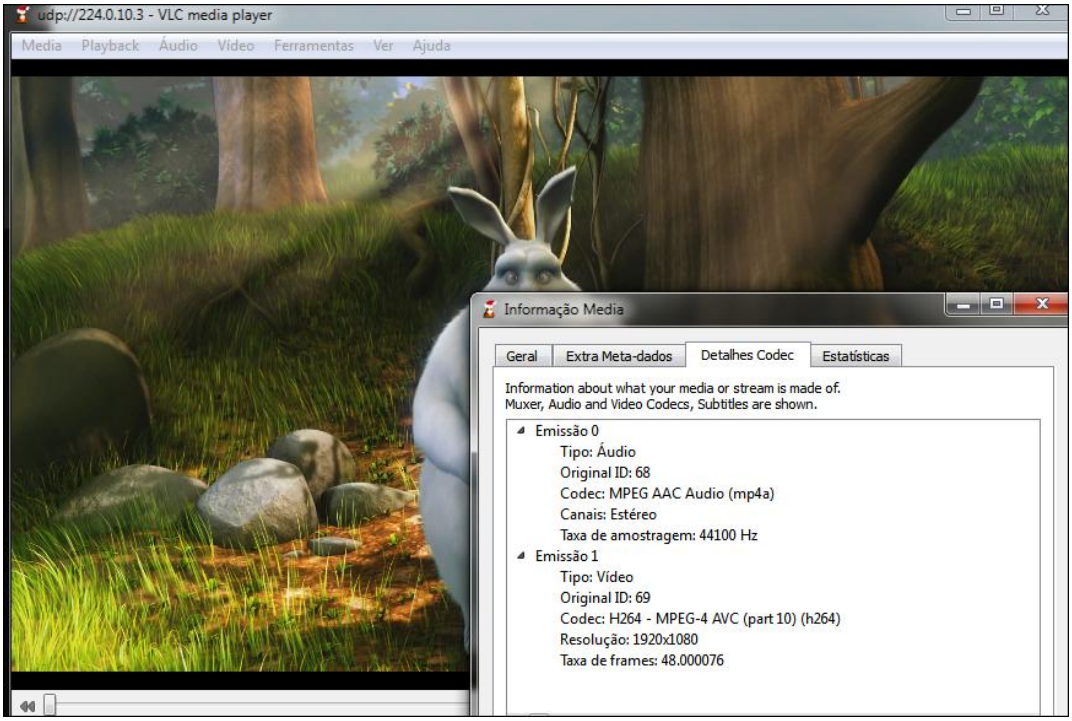

Figura 45 - vídeo *Full* HD sem degradação

7.8 Teste 3 – medição de *frames* perdidas com tráfego na rede

Este teste baseia-se na medição de *frames* perdidas em cada um dos grupos *multicast*, correspondentes às resoluções (SD, HD, FHD), com a injecção de tráfego na rede no sentido das *streams multicast*, e verificar para cada uma das resoluções, o limiar de tráfego *unicast* que é transferido até degradar o vídeo.

Na Tabela 14 estão representados os limiares de tráfego udp a injectar, aquando da recepção das *streams*. O máximo de tráfego a injectar é 54 Mbps.

Neste teste, há uma alteração na configuração do *IPerf*, porque, para o tráfego injectado seguir a mesma direcção do tráfego *multicast*, os clientes das *stations ubiquiti* assumem o papel de servidores de *IPerf*. Para ser possível esta alteração é necessária a configuração do *Port Forwarding* nas duas *station* para que o tráfego do *Iperf* passe no NAT.

O teste é feito para cada uma das resoluções. A avaliação da degradação do vídeo numa escala é de 0 – 5:

0 – Não existe degradação
5 – Degradação total

Limiar	Bit Rate Gerado	Comando Cliente IPerf
A	**1 Mbps**	`Iperf –c <IP da station> –u –P 1 –i 1 –p 5010 –f m –b 1M –t 500 –T 3`
B	**2 Mbps**	`Iperf –c <IP da station> –u –P 1 –i 1 –p 5010 –f m –b 2M –t 500 –T 3`
C	**4 Mbps**	`Iperf –c <IP da station> –u –P 1 –i 1 –p 5010 –f m –b 4M –t 500 –T 3`
D	**8 Mbps**	`Iperf –c <IP da station> –u –P 1 –i 1 –p 5010 –f m –b 8M –t 500 –T 3`
E	**16 Mbps**	`Iperf –c <IP da station> –u –P 1 –i 1 –p 5010 –f m –b 16M –t 500 –T 3`
F	**32 Mbps**	`Iperf –c <IP da station> –u –P 1 –i 1 –p 5010 –f m –b 32M –t 500 –T 3`
G	**54 Mbps**	`Iperf –c <IP da station> –u –P 1 –i 1 –p 5010 –f m –b 54M –t 500 –T 3`

Tabela 14 - Limiares de tráfego *unicast* a injectar em cada cliente

Teste do vídeo SD h264.ts no grupo *multicast* 224.0.10.1 c/2 clientes + injecção de tráfego

No resultado dos testes apresentado na Tabela 15, é possível verificar que o vídeo SD nunca sofre degradação mesmo com injecção de pacotes *unicast* em simultâneo. No entanto, a perda de pacotes injectados é maior nos limiares, os quais, somados com o *bit rate* do vídeo resultam num *bit rate* superior a 29Mbps na *interface* TX do *Microtik*.

Nos testes do limiar E e G existem perdas de *frames* que não estão relacionadas com a injecção de tráfego na rede.

		Medições VLC Cliente				Medições Injector de tráfego IPERF		Medições Microtik
Limiar	**Cliente VLC na station**	***Frames* reprod.**	***Frames* Perdidas**	**% Perda de *frames***	**Degradação**	**Jitter total do tráfego injectado (ms)**	**% Perda de Pacotes inject.**	**BitRate Máx de TX wlan1 (Mbps)**
A	**Bullet 5**	**12216**	**0**	**0,00**	**0**	**0,106**	**0,00**	**3,8**
	PStation	**12218**	**0**	**0,00**	**0**	**1,941**	**0,00**	
B	**Bullet 5**	**12315**	**0**	**0,00**	**0**	**0,372**	**0,00**	**5,8**
	PStation	**12316**	**0**	**0,00**	**0**	**1,721**	**0,00**	
C	**Bullet 5**	**12253**	**0**	**0,00**	**0**	**0,364**	**0,00**	**9,3**
	PStation	**12257**	**0**	**0,00**	**0**	**1,847**	**0,00**	
D	**Bullet 5**	**12218**	**0**	**0,00**	**0**	**0,117**	**0,13**	**17,9**
	PStation	**12219**	**0**	**0,00**	**0**	**1,219**	**0,12**	
E	**Bullet 5**	**12486**	**1**	**0,01**	**0**	**0,382**	**23**	**29**
	PStation	**12488**	**0**	**0,00**	**0**	**0,907**	**23**	
F	**Bullet 5**	**12494**	**0**	**0,00**	**0**	**0,564**	**61**	**28,9**
	PStation	**12491**	**0**	**0,00**	**0**	**0,291**	**62**	
G	**Bullet 5**	**12488**	**3**	**0,02**	**0**	**0,452**	**74**	**29,3**
	PStation	**12488**	**0**	**0,00**	**0**	**0,697**	**74**	

Tabela 15 - Resultado dos teste na qualidade SD

Teste do vídeo HD h264.ts no grupo *multicast* 224.0.10.2 c/2 clientes + injecção de tráfego

No resultado dos testes apresentado na Tabela 16, é possível verificar que o vídeo HD raramente sofre degradação, mesmo com injecção de pacotes *unicast* em simultâneo.

No entanto, a perda de pacotes injectados é maior nos limiares, os quais, somados com o *bit rate* do vídeo resultam num *bit rate* superior a 29Mbps na *interface* TX do *Microtik*.

Não existe relação entre a perda de *frames*, e o valor de tráfego injectado.

Surgem alguns episódios esporádicos de pixelização nos testes, nos limiares F e G. No entanto, não se perdem *frames*. O que sugere erros nos pacotes recebidos.

		Medições VLC Cliente				Medições Injector de tráfego IPERF		Medições Microtik
Limiar	**Cliente VLC na station**	***Frames* reprod.**	***Frames* Perdidas**	**% Perda de *frames***	**Degradação**	**Jitter total do tráfego injectado (ms)**	**% Perda de Pacotes inject.**	**BitRate Máx de TX wlan1 (Mbps)**
A	**Bullet 5**	**12341**	**0**	**0,00**	**0**	**0,532**	**0,00**	**4,6**
	PStation	**12333**	**0**	**0,00**	**0**	**2,010**	**0,00**	
B	**Bullet 5**	**12429**	**1**	**0,00**	**0**	**0,093**	**0,00**	**6,9**
	PStation	**12421**	**1**	**0,00**	**0**	**2,320**	**0,00**	
C	**Bullet 5**	**12475**	**1**	**0,01**	**0**	**0,412**	**0,01**	**10,8**
	PStation	**12467**	**1**	**0,01**	**0**	**1,718**	**0,06**	
D	**Bullet 5**	**12443**	**7**	**0,06**	**0**	**0,883**	**1,8**	**18,9**
	PStation	**12416**	**5**	**0,04**	**0**	**1,519**	**2,1**	
E	**Bullet 5**	**12416**	**2**	**0,01**	**1**	**0,764**	**33**	**28,9**
	PStation	**12409**	**2**	**0,00**	**1**	**1,103**	**33**	
F	**Bullet 5**	**12463**	**0**	**0,00**	**1**	**12,320**	**66**	**29**
	PStation	**12455**	**0**	**0,00**	**1**	**0,669**	**67**	
G	**Bullet 5**	**12451**	**0**	**0,00**	**2**	**6,293**	**76**	**28,8**
	PStation	**12438**	**0**	**0,00**	**2**	**37,995**	**77**	

Tabela 16 - Resultado dos testes na qualidade HD

Figura 46 - Teste 3 com resolução HD - degradação Limiar E

Durante um espaço de tempo, no limiar E, a imagem está pixelizada (Figura 46).

Teste do vídeo FHD h264.ts no grupo *multicast* 224.0.10.3 c/2 clientes + injecção de tráfego

Figura 47 - Teste 3 com resolução FHD – degradação nível 5

No resultado dos testes apresentado na Tabela 17, é possível verificar que o vídeo FHD sofre degradação total em qualquer limiar de injecção de pacotes *unicast*.

A perda de pacotes injectados existe em todos os limiares de teste, sendo maior nos limiares somados com o *bit rate* do vídeo resultem num *bit rate* superior a 18Mbps na *interface* TX do *Microtik*.

Existem também longos períodos de congelamento do vídeo, nos quais a degradação é máxima (Figura 47).

		Medições VLC Cliente				Medições Injector de tráfego IPERF		Medições Microtik
Limiar	**Cliente VLC na station**	***Frames* reprod.**	***Frames* Perdidas**	**% Perda de *frames***	**Degradação**	**Jitter total do tráfego injectado (ms)**	**% Perda de Pacotes inject.**	**BitRate Máx de TX wlan1 (Mbps)**
A	**Bullet 5**	**13201**	**67**	**0,50**	**3**	**0,066**	**0,28**	**7,9**
	PStation	**13181**	**98**	**0,74**	**4**	**0,781**	**0,01**	
B	**Bullet 5**	**12893**	**101**	**0,78**	**4**	**0,609**	**0,01**	**9,7**
	PStation	**12910**	**87**	**0,67**	**4**	**2,178**	**0,11**	
C	**Bullet 5**	**13893**	**163**	**1,17**	**5**	**0,442**	**0,34**	**12,8**
	PStation	**13973**	**89**	**0,63**	**4**	**1,687**	**0,36**	
D	**Bullet 5**	**13989**	**217**	**1,55**	**5**	**0,987**	**18**	**19,7**
	PStation	**14071**	**134**	**0,95**	**4**	**1,787**	**18**	
E	**Bullet 5**	**13986**	**253**	**1,81**	**5**	**12,829**	**53**	**29,4**
	PStation	**14136**	**97**	**0,68**	**5**	**1,065**	**52**	
F	**Bullet 5**	**13404**	**230**	**1,72**	**5**	**11,489**	**76**	**29,5**
	PStation	**13526**	**116**	**0,86**	**5**	**4,837**	**76**	
G	**Bullet 5**	**13507**	**227**	**1,68**	**5**	**5,212**	**85**	**29,4**
	PStation	**13644**	**92**	**0,67**	**5**	**6,824**	**85**	

Tabela 17 - Resultado dos testes na qualidade FHD

A situação de degradação total do vídeo é semelhante à que ocorre no teste efectuado no capítulo 7.7, "Teste do vídeo FHD h264.ts no grupo *multicast* 224.0.10.1 c/2 clientes". No qual, sem injecção de tráfego é possível verificar a

degradação máxima do vídeo FHD, sem a existência de qualquer tipo de tráfego em paralelo na rede.

8. Conclusão

Neste capítulo da dissertação é efectuada a conclusão do trabalho efectuado, com a realização dum resumo sobre o trabalho de investigação, e a percepção da dimensão de alcance dos objectivos inicialmente propostos. São também indicadas algumas considerações relativas a trabalho futuro, baseadas na análise do trabalho desenvolvido.

8.1 Conclusões

Esta tese propõe o estudo dos conceitos e tecnologias usadas nas arquitecturas IPTV, tais como, *codecs*, protocolos, QoS, QoE, e componentes. Com base neste estudo pretende-se estudar o comportamento da IPTV nas redes *Wi-fi.* Nomeadamente, o comportamento das *streams* de vídeo *multicast* através destas redes, que são alvo dum enorme número de perturbações por trabalharem num meio partilhado, sujeitos a todo o tipo de interferências que é o ar.

Como ponto de partida, é efectuado o levantamento de referências bibliográficas sobre IPTV e estudo das mesmas, que resultam no "Capítulo 2 – Estado da Arte". A maior parte dos assuntos abordados no "Capítulo 2" são aprofundados nos capítulos seguintes.

No "Capítulo 6 – Arquitectura da Solução", são definidos os requisitos a cumprir numa rede *wireless* para suporte de *streams multicast*. É feita a identificação dos problemas que as redes *wireless* têm com IP *multicast*. No final, e com base em todo o estudo efectuado nesta dissertação, é proposta a arquitectura a implementar para efectuar os testes de comportamento do IPTV, em redes *wireless*.

Na implementação são feitos dois tipos de testes com *streams multicast*.

O primeiro tipo, consiste em avaliar a capacidade da rede transportar *streams* nas resoluções SD (720x480 *pixeis*), HD (1280x720 *pixeis*), e FHD (1920x1080 *pixeis*), com diferentes *bit rates* e sem outro tipo de tráfego na rede. É possível concluir que só existe degradação sempre que o *bit rate* do vídeo atinge valores superiores a 6 Mbps, o que acontece com o vídeo FHD, cuja codificação é feita a 6 Mbps VBR. A degradação nestas condições é de certa forma previsível, devido à implementação da camada de MAC, definida na norma 802.11, que adverte para problemas no tráfego *multicast* e *broadcast*.

A solução apresentada por alguns fabricantes, passa por transmitir este tipo de tráfego *basic rate* mais baixa, que no caso da norma 802.11a, é de 6 Mbps, o que explica a degradação do vídeo sempre que este passa do 6 Mbps.

O segundo tipo de teste consiste em avaliar a capacidade da rede transportar as mesmas *streams*, mas com injecção de tráfego na rede *wireless*, até ocorrer degradação do vídeo (*frames* perdidas). Para perceber quando existe tráfego *multicast* na rede, toda a rede fica a funcionar apenas na *basic rate* mais baixa, o que paralisa-se todo o restante tráfego, ou se admite tráfego até ao limite de TX identificado nos teste de desempenho iniciais (Capítulo 7.2) com tráfego *unicast*.

Verifica-se que o tráfego *multicast* desde que não passe o limite da *basic rate* mais baixa, o que acontece com os vídeos SD e HD, não sofre qualquer tipo de degradação, independentemente da saturação da rede com tráfego unicast. No entanto, quando o tráfego multicast passa o limite da basic rate mais baixa, o que acontece no vídeo FHD, a degradação do vídeo é total, e o tráfego *unicast* apresenta perda apresenta perda de pacotes em todos os limiares de teste.

Em ambos os testes, conclui-se que de alguma forma existe prioritização do tráfego *multicast* em relação ao tráfego *unicast*. E a rede com ou sem *multicast*, chega sempre à taxa de transmissão medida nos testes de desempenho com tráfego *unicast*.

Conclui-se que é possível utilizar IP *multicast* em redes *wireless*, desde que os equipamentos cumpram os requisitos definidos nesta tese, e o *bit rate do* vídeo não ultrapasse o limite da *basic rate* mais baixa. Uma das formas de permitir mais tráfego *multicast*, é desactivar a *basic rate* mais baixa do AP, para que esta fique com um patamar mais alto. É óbvio que esta alteração pode implicar outros problemas na rede, sendo um deles a perda de alcance.

8.2 Trabalho Futuro

Como trabalho futuro, é importante efectuar testes com outras normas 802.11, e desenvolver, talvez averiguar a possibilidade de integrar *adaptative streaming* com o IP *multicast.* Tal combinação poderia resolver os problemas problemas de degradação evidenciados nos testes. Outro aspecto a considerar para trabalho futuro, são o desenvolvimento de um *midleware* com integração de CAS e DRM, para operar com STB. Outro tópico interessante é investigar mecanismos que aumentem a interação do sistema com o utilizador. Como por exemplo, perfis de utilizadores personalizáveis, nos quais, o sistema de alguma forma reconheceria o utilizador para carregar o perfil do mesmo.

Esta área está em contante evolução, o que a torna muito interessante e cativante para o desenvolvimento de projectos inovadores.

Bibliografia

Agency, D. A. (1981). *Internet Protocol - DARPA INTERNET PROGRAM - Protocol specification.* 1400 Wilson Boulevard, Arlington, Virginia 22209: Defense Advanced Research Projects Agency.

Ahmed Helmy, D. T. (1997). *Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification.*

Alliance, W.-F. (s.d.). *Articles*. Obtido em 1 de 12 de 2010, de Wi-Fi Alliance: http://www.wi-fi.org/knowledge_center/wmm

Almquist, P. (July 1992). *RFC1349 - Type of Service in the Internet Protocol Suite.*

Apple. (s.d.). *Darwin Streaming Server* . Obtido em 1 de 10 de 2010, de Darwin Streaming Server : http://dss.macosforge.org/

Braun, T. S. (2008). *End-to-End Quality of Service over Heterogeneous Networks.* Berlin: Springer.

Bringuier, L. (2010). *OTT Streaming.* France: Anevia.

Britannica, E. (19 de 07 de 2010). *Young-Helmholtz three-colour theory*. Obtido de Encyclopedia Britannica: http://www.britannica.com/EBchecked/topic/654047/Young-Helmholtz-three-colour-theory

Christophe Diot, B. N. (2000). Deployment Issues for the IP Multicast Service and Architecture. *IEEE Network* .

Chunglae Cho, I. H. (2007). Improvement of Channel Zapping Time in IPTV Services Using the Adjacent Groups Join-Leave Method. *Network Technology Laboratory, ETRI* .

CONE (célula). (07 de 2010). Obtido em 19 de 09 de 2010, de Wikipedia: http://pt.wikipedia.org/wiki/Cone_%28c%C3%A9lula%29

D.Gibson, J. (2001). *Multimedia communications: Directions and inovations.* United States of America: Academic Press.

Damodar Banodkar, K. R. (2008). *Multicast Instant Channel Change in IPTV Systems.* Rensselaer Polytechnic Institute (RPI), AT&T Labs Research.

Geert Van der Auwera, P. T. (2008). *Traffic Characteristics of H.264/AVC Variable.*

Ghanbari, M. (2003). Standard codecs: image compression to advanced video coding. In M. Ghanbari, *Standard codecs: image compression to advanced video coding* (pp. 1-22). Institution of Electrical Engineers.

Grossman, D. (April 2002). *RFC3260 - New Terminology and Clarifications for Diffserv.*

H.263, I.-T. R. (1996). *Video Coding For Low Bit Rate Communication.* Geneva: ITU - International Telecomunication Union.

Held, G. (2006). *UNDERSTANDING IPTV.* New York: Auerbach Publicatio.

Hjelm, J. (2008). *WHY IPTV? Interactivity, Technologies, Services.* United Kingdom: WILEY.

HVS, S. d. (s.d.). *Como funcona o olho humano*. Obtido em 2010 de 08 de 1, de SAC: http://www.sac.org.br/APR_FOH.htm

IEEE. (12 June 2007). *802 Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.* 3 Park Avenue, New York, NY 10016-5997, USA: IEEE.

IGMP Fast Leave. (s.d.). Obtido de CTS: http://www.ctsystem.com/technology/MetroETechNote.aspx?tnno=9

J.Walko, I. C. (Dec. 2005). I love my IPTV. 16-19.

Kolher, E., Handley, M., & Floyd, S. (2006). *Designing DCCP: Congestion Control Without Reliability.* ND: ACM SIGGCOM.

Lin, C.-H. (25 de 04 de 2009). *Evaluation of video stream quality over IEEE 802.11e EDCF*. Obtido em 12 de 12 de 2010, de Evaluation of video stream quality over IEEE 802.11e EDCF: http://140.116.72.80/~jhlin5/ns2/802_11e/NS-2_80211e.htm

Liu, C. (2000). *Multimedia Over IP.* ND: CiteSeer.

Live 555. (s.d.). *Live 555*. Obtido em 2 de 10 de 2010, de Live 555: http://www.live555.com/

Mattila, J. (2003). *Real-Time Transport Protocol.* Department of Computer Science Helsinki. Helsinki: University of Helsinki.

Microsoft. (2010). *http://www.iis.net/media/experiencesmoothstreaming*. Obtido em 16 de 12 de 2010, de http://www.iis.net/media/experiencesmoothstreaming: http://www.iis.net/media/experiencesmoothstreaming

Microtik. (s.d.). *Manual:Multicast and wireless*. Obtido em 15 de 12 de 2010, de Microtik WIKI: http://wiki.mikrotik.com/wiki/Manual:Multicast_and_wireless

Paul, S. (2011). *Digital Video Distribution in Broadband, Television, Mobile and Converged Networks.* ND: WILEY.

Perkins, C., Glasgow, U. o., Westerlund, M., & Ericsson. (2010). *Multiplexing RTP Data and Control Packets on a Single Port.* IETF: Internet Engineering Task Force (IETF.

Perkins, C., Glasgow, U. o., Westerlund, M., & Ericsson. (Abril de 2010). *RFC5761 Multiplexing RTP Data and Control Packets on a Single Port*. Obtido de rfc-editor: http://www.rfc-editor.org/rfc/rfc5761.txt

Prof. Rathnakar Acharya, D. V. (2010). WLAN QoS Issues and IEEE 802.11e QoS Enhancement. *International Journal of Computer Theory and Engineering, Vol. 2, No. 1 February, 2010* .

Prof. Rathnakar Acharya, D. V. (February, 2010). *WLAN QoS Issues and IEEE 802.11e QoS Enhancement.* International Journal of Computer Theory and Engineering, Vol. 2, No. 1.

R. Braden, D. C. (June, 1994). *Integrated Services in the Internet Architecture: an Overview.*

R. Braden, L. Z. (September 1997). *RFC2205 - Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification.*

R.Braden, D. S. (1994). *RFC-1633: Integrated Services in the Internet Architecture: an Overview.*

Reverse-Path-Forwarding. (Novembro de 2009). Obtido de http://en.wikipedia.org/wiki/Reverse-path_forwarding

RFC1112, I. (s.d.). Obtido de RFC 1112: http://tools.ietf.org/html/rfc1112

RFC2236, I. (s.d.). *RFC 2236*. Obtido de http://www.ietf.org/rfc/rfc2236.txt

RFC3376, I. (s.d.). *IETF RFC3376*. Obtido de IETF: http://tools.ietf.org/html/rfc3376

RFC4601. (2006). *Protocol Independent Multicast - Sparse Mode (PIM-SM)*. Obtido de http://www.rfc-editor.org/rfc/rfc4601.txt

Rizzo, L. (1998). Fast group management in IGMP. *Proceeding of Hipparc.*

Schulzrinne, H., & Casner, S. (Julho de 2003). *http://www.rfc-editor.org/rfc/rfc3551.txt*. Obtido de rfc-editor: http://www.rfc-editor.org/rfc/rfc3551.txt

Schulzrinne, H., Casner, S., Frederick, R., & Jacobson, V. (Julho de 2003). *http://www.rfc-editor.org/rfc/rfc3550.txt*. Obtido de rfc-editor: http://www.rfc-editor.org/rfc/rfc3550.txt

Schulzrinne, H., Rao, A., & Lanphier, R. (Abril de 1998). *Real Time Streaming Protocol (RTSP) RFC2326*. Obtido de http://www.ietf.org: http://www.ietf.org/rfc/rfc2326.txt

Siemens Communications and Juniper Networks, I. *High Quality and Resilient IPTV Multicast Architecture.* Siemens Communications and Juniper Networks.

STACEY, E. P. (2008). Next Generation Wireless LANs. In E. P. STACEY, *Next Generation Wireless LANs* (pp. IX - XXIV, 185 - 215). United States of America by Cambridge University Press, New York: Cambridge.

Systems, C. (2007). *The Exabyte Era*. Obtido em 2010, de Cisco Systems: http://www.cisco.com/en/US/solutions/collateral/ns813/white_paper_c11_480623_v2.pdf

Thomas Stockhammer, M. M. (2003). *H.264/AVC in Wireless Environments.* IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, VOL. 13, NO. 7, JULY 2003.

VideoLan. (s.d.). *VLC media player*. Obtido em 5 de 10 de 2010, de Video Lan Project: http://www.videolan.org/vlc/

Vít Novotný, D. K. (2008). *Large-Scale RTCP Feedback Optimization.* Brno, Czech Republic: Faculty of Electrical Engineering and Communications.

Weijun Wang, Y. L., Yan, L., Li, H., & Yang, X. (2007). *Integration and Innovation Orient to E-Society, Volume 2.* United States of America: Springer.

Wenger, S., Hannuksela, M., T.Stockhammer, & Westerlund, D. (2005). *RTP Payload Format for H.264 Video.* ND: ITU-T.

Wi-Fi Alliance. (September 1, 2005). *Wi-Fi CERTIFIED™ for WMM™ - Support for Multimedia Applications with Quality of Service in Wi-Fi® Networks.* Wi-Fi Alliance.

Wikipedia. (27 de 08 de 2010). *Nervo óptico*. Obtido de Wilkipedia: http://pt.wikipedia.org/wiki/Nervo_%C3%B3ptico

Wireshark. (s.d.). *Wireshark*. Obtido em 15 de 10 de 2010, de Wireshark: http://www.wireshark.org/

Y. Bernet, P. F. (November 2000). *RFC2998 - A Framework for Integrated Services Operation over Diffserv Networks.* ND: The Internet Society (2000).

Zambelli, A. (March, 2009). *IIS Smooth Streaming Technical Overview.* Microsoft Coorporation.

ANEXO A - Equipamentos

RouterBOARD 433UAH

The universal wireless access point.

The RB433UAH features two USB 2.0 ports which allow you to connect external storage devices, 3G modems and more.

The microSD card slot can be used for storing web proxy cache, log files, user manager and dude databases.

The 680MHz Atheros MIPs 24K CPU with a 64KB/32KB instruction/data cache is probably the fastest CPU used in low cost wireless access points.

The three Ethernet ports and three miniPCI slots give you ample data interfaces to put the big CPU power to work.

CPU	Atheros AR7161 680MHz network processor
Memory	128MB DDR SDRAM onboard memory
Boot loader	RouterBOOT
Data storage	512MB onboard NAND memory chip and microSD
Ethernet	Three 10/100 Mbit/s Ethernet ports with Auto-MDI/X
Expansion	2x USB 2.0 ports, max 480Mbit throughput. Max current 2A.
miniPCI	Three MiniPCI Type IIIA/IIIB slots
Extras	Reset switch, Beeper,Voltage monitor
Serial port	One DB9 RS232C asynchronous serial port
LEDs	Power, NAND activity, 5 user LEDs
Power options	Power over Ethernet: 10..28V DC (except power over datalines). Power jack: 10..28V DC. Voltage monitor.
Dimensions	10.5 cm x 15 cm, 137 grams
Power consumption	3W without extension cards, up to 10W when using USB. Maximum - 32W, 16W output to cards
Operating System	MikroTik RouterOS v3, Level5 license

RouterBOARD R52n

802.11a/b/g/n dual band miniPCI card

Key Features and Benefits

- Dual band IEEE 802.11a/b/g/n standard
- Output Power of up to **23dBm**
- Support for up to 2x2 MIMO with spatial multiplexing
- Four times the throughput of 802.11a/g
- Atheros AR9220, chipset
- High Performance (up to 300Mbps physical data rates and 200Mbps of actual user throughput) with Low Power Consumption
- 2 X U.FL Antenna Connector
- Modulations:
  OFDM: BPSK, QPSK, 16 QAM, 64QAM
  DSSS: DBPSK, DQPSK, CCK
- Operating temperatures: -50ºC to 60ºC
- Power consumption MAX 2.4W

The RouterBOARD R52n miniPCI network adapter provides leading 802.11a/b/g/n performance in both 2GHz and 5GHz bands, supporting up to 300Mbps physical data rates and up to 200Mbps of actual user throughput on both the uplink and downlink. Adding Wireless N to your Wireless device, it provides higher efficiency for everyday activities such as local network file transfers, Internet browsing, and media streaming.

802.11b	RX Sensitivity	Composite TX Power
1Mbit	-95	20
11Mbit	-91	21
802.11g		
6Mbit	-95	23
54Mbit	-81	19
802.11n 2.4GHz		
MCS0 20MHz	-95	21
MCS0 40MHz	-90	21
MCS7 20MHz	-78	17
MCS7 40MHz	-75	16

802.11a	RX Sensitivity	Composite TX Power
6Mbit	-95	21
54Mbit	-80	17
802.11n 5GHz		
MCS0 20MHz	-95	21
MCS0 40MHz	-92	19
MCS7 20MHz	-77	16
MCS7 40MHz	-74	13

Data Rates

802.11b	
	11Mbps; 5.5Mbps; 2Mbps; 1Mbps
802.11a/g	
	54Mbps; 48Mbps; 36Mbps; 24Mbps; 18Mbps; 12Mbps; 9Mbps; 6Mbps
802.11n	
20MHz	1Nss: 65Mbps @ 800GI, 72.2Mbps @ 400GI (Max.) 2Nss: 130Mbps @ 800GI, 144.4Mbps @ 400GI (Max.)
40MHz	1Nss: 135Mbps @ 800GI, 150Mbps @ 400GI (Max.) 2Nss: 270Mbps @ 800GI, 300Mbps @ 400GI (Max.)

ANEXO B - Testes de Largura de Banda

Parâmetros do IPerf:

```
Usage: iperf [-s|-c host] [options]
     iperf [-h|--help] [-v|--version]

Client/Server:
 -f, --format    [kmKM]   format to report: Kbits, Mbits, KBytes, MBytes
 -i, --interval  #        seconds between periodic bandwidth reports
 -l, --len       #[KM]    length of buffer to read or write (default 8 KB)
 -m, --print_mss          print TCP maximum segment size (MTU - TCP/IP header)
 -o, --output    <filename> output the report or error message to this specified file
 -p, --port      #        server port to listen on/connect to
 -u, --udp                use UDP rather than TCP
 -w, --window    #[KM]    TCP window size (socket buffer size)
 -B, --bind      <host>   bind to <host>, an interface or multicast address
 -C, --compatibility      for use with older versions does not sent extra msgs
 -M, --mss       #        set TCP maximum segment size (MTU - 40 bytes)
 -N, --nodelay            set TCP no delay, disabling Nagle's Algorithm
 -V, --IPv6Version        Set the domain to IPv6

Server specific:
 -s, --server             run in server mode
 -U, --single_udp         run in single threaded UDP mode
 -D, --daemon             run the server as a daemon

Client specific:
 -b, --bandwidth #[KM]    for UDP, bandwidth to send at in bits/sec
                          (default 1 Mbit/sec, implies -u)
 -c, --client    <host>   run in client mode, connecting to <host>
 -d, --dualtest           Do a bidirectional test simultaneously
 -n, --num       #[KM]    number of bytes to transmit (instead of -t)
 -r, --tradeoff           Do a bidirectional test individually
 -t, --time      #        time in seconds to transmit for (default 10 secs)
 -F, --fileinput <name>   input the data to be transmitted from a file
 -I, --stdin              input the data to be transmitted from stdin
 -L, --listenport #       port to recieve bidirectional tests back on
 -P, --parallel  #        number of parallel client threads to run
 -T, --ttl       #        time-to-live, for multicast (default 1)
 -Z, --linux-congestion <algo>  set TCP congestion control algorithm (Linux only)

Miscellaneous:
 -x, --reportexclude [CDMSV]   exclude C(connection) D(data) M(multicast) S(settings)
V(server) reports
 -y, --reportstyle C      report as a Comma-Separated Values
 -h, --help               print this message and quit
```

-v, --version print version information and quit

[KM] Indicates options that support a K or M suffix for kilo- or mega-

The TCP window size option can be set by the environment variable
TCP_WINDOW_SIZE. Most other options can be set by an environment variable
IPERF_<long option name>, such as IPERF_BANDWIDTH.

Report bugs to iperf-users@lists.sourceforge.net

Alguns Parâmetros do VLC

VLC media player 1.1.4 The Luggage (revision exported)
Usage: vlc [options] [*stream*] ...
You can specify multiple streams on the commandline. They will be enqueued in the playlist.
The first item specified will be played first.

Options-styles:
--option A global option that is set for the duration of the program.
-option A single letter version of a global --option.
:option An option that only applies to the *stream* directly before it
and that overrides previous settings.

Stream MRL syntax:
[[access][/demux]://]URL[@[title][:chapter][-[title][:chapter]]] [:option=value ...]

Many of the global --options can also be used as MRL specific :options.
Multiple :option=value pairs can be specified.

URL syntax:
[file://]filename Plain media file
http://ip:port/file HTTP URL
ftp://ip:port/file FTP URL
mms://ip:port/file MMS URL
screen:// Screen capture
[dvd://][device][@raw_device] DVD device
[vcd://][device] VCD device
[cdda://][device] Audio CD device
udp://[[<source address>]@[<bind address>][:<bind port>]]
UDP *stream* sent by a streaming server
vlc://pause:<seconds> Special item to pause the playlist for a certain time
vlc://quit Special item to quit VLC

No matching module found. Use --list or --list-verbose to list available modules.

Audio
--audio, --no-audio Enable audio (default enabled)
--volume <integer [0 .. 1024]>
Default audio volume

```
    --spdif, --no-spdif       Use S/PDIF when available (default disabled)
    --force-dolby-surround {0 (Auto), 1 (On), 2 (Off)}
                        Force detection of Dolby Surround
    --audio-replay-gain-mode {none,track,album}
                        Replay gain mode
    --audio-replay-gain-preamp <float>
                        Replay preamp
    --audio-replay-gain-default <float>
                        Default replay gain
    --audio-time-stretch, --no-audio-time-stretch
                        Enable time streching audio (default enabled)
    --audio-filter <string>    Audio filters
    --audio-visual <string>    Audio visualizations

Subpictures
  On Screen Display:
    --osd, --no-osd           On Screen Display (default enabled)
  Subtitles:
    --sub-file <string>       Use subtitle file
    --sub-autodetect-file, --no-sub-autodetect-file
                        Autodetect subtitle files (default enabled)
  Overlays:
    --sub-filter <string>     Subpictures filter module
  Track settings:
    --audio-language <string>  Audio language
    --sub-language <string>    Subtitle language
  Playback control:
    --input-repeat <integer>   Input repetitions
    --input-fast-seek, --no-input-fast-seek
                        Fast seek (default disabled)
    --rate <float>            Playback speed
  Default devices:
    --dvd <string>            DVD device
    --vcd <string>            VCD device
    --cd-audio <string>       Audio CD device
  Network settings:
    --server-port <integer>    UDP port
 -6, --ipv6, --no-ipv6         Force IPv6 (default disabled)
 -4, --ipv4, --no-ipv4         Force IPv4 (default disabled)

 Input
  Advanced:
    --prefer-system-codecs, --no-prefer-system-codecs
                        Prefer system plugins over VLC (default
                        disabled)
    --stream-filter <string>   Stream filter module

 Playlist
```

```
  Performance options:
 -Z, --random, --no-random      Play files randomly forever (default disabled)
 -L, --loop, --no-loop          Repeat all (default disabled)
 -R, --repeat, --no-repeat      Repeat current item (default disabled)
     --play-and-exit, --no-play-and-exit
                        Play and exit (default disabled)
     --play-and-stop, --no-play-and-stop
                        Play and stop (default disabled)
     --media-library, --no-media-library
                        Use media library (default enabled)
     --playlist-tree, --no-playlist-tree
                        Display playlist tree (default disabled)
     --open <string>            Default stream
     --auto-preparse, --no-auto-preparse
                        Automatically preparse files (default enabled)
     --album-art {0 (Manual download only), 1 (When track starts playing), 2 (As soon as track is
added)}
                        Album art policy
 -S, --services-discovery <string>
                        Services discovery modules
 -v, --verbose <integer>        Verbosity (0,1,2)
     --verbose-objects <string> Choose which objects should print debug
                        message
```