



Departamento de Ciências e Tecnologias da Informação

MQV3D: Monitor de Qualidade de Vídeo 3D

Ricardo Jorge Macedo Mota João

Dissertação submetida como requisito parcial para obtenção do grau de
Mestre em Engenharia de Telecomunicações e Informática

Orientador:
Doutor Rui Miguel Neto Marinheiro, Professor Auxiliar,
ISCTE- Instituto Universitário de Lisboa

Setembro, 2015

Resumo

Devido à crescente evolução das tecnologias 3D, nos próximos anos o IPTV 3D pode vir a tornar-se um serviço largamente distribuído pelos operadores, desta forma é importante que estes possam ter uma forma fácil e barata de monitorizar a qualidade do serviço que oferecem. A instalação de um monitor de qualidade de experiência de vídeo 3D em qualquer ponto da rede vai permitir aos operadores adaptarem o seu serviço e infra-estrutura de forma a poderem garantir uma determinada qualidade de experiência aos seus utilizadores, por exemplo para uma aplicação de IPTV

Esta dissertação de mestrado apresenta o desenvolvimento do protótipo de um monitor de qualidade de vídeo 3D que permite a avaliação de vídeo H.264 para diferentes tipos de modelos de avaliação de qualidade de vídeo. O protótipo oferece uma arquitetura que permite uma fácil integração de novos modelos de avaliação, flexibilidade de gestão dos dados de entrada, execução pela linha de comando, gestão de *Logs* e monitorização em tempo real.

Foi ainda implementado um modelo de avaliação de qualidade de vídeo V+P com o qual foram realizados testes, onde se introduziram erros com o simulador G.1050. Com o monitor desenvolvido obtiveram-se resultados de qualidade com uma forte correlação com a qualidade medida dos vídeos utilizados. Estes resultados permitiram validar o modelo de avaliação de qualidade mesmo com as considerações tomadas para que este funcionasse num cenário operacional.

Abstract

Due to the increasing evolution of 3D technology in the coming years, the 3D IPTV may prove to become a service widely distributed by the operators. So it is important that they can have an easy and inexpensive way to monitor the quality of service they offer. The installation of a 3D video experience quality monitor anywhere in the network will enable operators to adapt their service and infrastructure so that they can ensure a certain quality of experience for their users, for example to an IPTV application.

This dissertation presents the prototype developed of a 3D video quality monitor. This prototype enables H.264 video assessment of multiple video evaluation models. The prototype provides an architecture that allows an easy integration of new evaluation models, support for multiple input data, command line mode, log management and monitoring in real time.

It was also implemented a video-plus-depth quality assessment model with which tests were performed. We have introduced errors with G.1050 simulator. With the developed monitor we've obtained quality results with a high correlation with the measured quality values. These results allowed us to validate the evaluation model even with the assumptions made so that it worked in a real scenario.

Agradecimentos

Quero agradecer especialmente aos professores do ISCTE Rui Marinheiro que orientou todo o meu trabalho, ao professor Tomás Brandão que me ajudaram com a codificação de vídeos e ao professor Luís Ducla Soares que indiretamente apoio o meu trabalho com a sugestão de alguma bibliografia. Quero ainda agradecer todo o apoio dado pelo professor Luís A. Da Silva Cruz da Faculdade de Ciências e Tecnologias da Universidade de Coimbra.

Quero agradecer a todos os colegas que me ajudaram ou contribuíram de alguma forma para este trabalho. Em particular quero destacar a ajuda dos colegas Bruno Feitor da Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria e o Hugo Alves do ISCTE- Instituto Universitário de Lisboa.

Este trabalho foi apoiado pelo projeto do Instituto de Telecomunicações: 3DVQM, IT/LA/P01131/2011 e pela Fundação para a Ciência e Tecnologia (FCT), Portugal no âmbito do projeto UID/EEA/50008/2013.

Um bem-haja a todos aqueles que me deram de alguma forma suporte para que conseguisse ter a disposição e a disponibilidade para realizar a escrita desta dissertação.

Índice

Resumo	i
<i>Abstract</i>	iii
Agradecimentos	v
Índice de Figuras	ix
Índice de Tabelas	xi
Índice de Abreviaturas	xiii
1 Introdução	1
2 Modelos de Avaliação de Vídeo	5
2.1 Formatos de representação de vídeo 3D	5
2.2 Métricas e Modelos de Avaliação de vídeo	7
2.3 Avaliação de vídeo MVC em redes broadcast	13
2.4 Avaliação de vídeo V+D em redes IP propensas a erros	17
2.5 Modelo de rede para avaliação de vídeo	22
3 Protótipo implementado	29
3.1 Arquitetura da solução	29
3.2 Descrição do protótipo existente	31
3.3 Generalização do Software	35
3.4 Cenários de Monitorização	39
3.4.1 Vídeo Multi-View-Coding em redes IP	40
3.4.2 Video-Plus-Depth em redes IP	44
3.5 Monitorização em tempo real e outras funcionalidades	51
4 Testes e resultados	57
4.1 Definição dos cenários de teste	57
4.2 SSIM estimado e medido	62

4.3	Apresentação dos resultados.....	63
5	Conclusão.....	71
	Referências	75
	Anexo A	79
	Anexo B.....	95
	Anexo C.....	101

Índice de Figuras

Figura 1 – Estrutura da predição MVC em duas vistas independentes a partir de uma vista de base [2].	6
Figura 2 – Estrutura interna da representação do formato vídeo plus depth [2]	7
Figura 3 – Pontos de rede onde é possível instalar cada um dos tipos de modelos existente [15].	9
Figura 4 – Classificação dos vários tipos de modelos VQA sem referência (NR) de acordo com o tipo de informação de transporte e vídeo que lhes estão associadas [16].	10
Figura 5 – Framework para o modelo de avaliação de qualidade de vídeo [9]	11
Figura 6 – Arquitetura de um modelo de avaliação para uma aplicação IPTV [10].	12
Figura 7 – estrutura do modelo packet layer utilizado para estimação de qualidade objetiva (SSIM) para um sistema broadcast [15].	15
Figura 8 – Arquitetura do modelo utilizado para a estimação do SSIM e DMOS [17].	19
Figura 9 – Modelo base de um elemento de rede [26].	23
Figura 10 – Modelo de rede core-to-LAN. Modelo utilizado por defeito nos cenários de teste incluídos no simulador [26].	24
Figura 11 – Estrutura lógica do monitor de vídeo 3D.	30
Figura 12 – Diagrama com fluxo de informação criado entre os dois processos utilizados no protótipo do monitor [15].	32
Figura 13 – Interface gráfica do protótipo com indicação dos vários componentes [15].	34
Figura 14 – Relação entre a classe Monitor e o cenário de monitorização concreto <i>MVC_BroadcastSystem</i> .	37
Figura 15 – Relação entre o cenário <i>MVC_BroadcastSystem</i> e o respetivo VQA concreto.	39

Figura 16 – Estrutura interna de um ficheiro PCAP.	41
Figura 17 – Relação entre a classe abstrata <i>NetworkScenario</i> e as classes <i>MVC_BroadcastSystem</i> e <i>MVC_OverRTP</i>	43
Figura 18 – Fluxo de execução para configurar a proveniência dos vídeos em ambos os cenários.	43
Figura 19 – Principais métodos e atributos do cenário de monitorização <i>VPlusD_OverRTP</i>	46
Figura 20 – Relação entre o cenário <i>VPlusD_OverRTP</i> e o respetivo modelo <i>VQA</i>	48
Figura 21 – Estrutura do ficheiro de configuração para leitura de parâmetros do modelo <i>VQA</i> apresentado na seção 2.4.	49
Figura 22 – Interface gráfica do Monitor de qualidade de experiência.	50
Figura 23 – Diagrama com o fluxo de execução que ocorre quando se solicitam pacotes provenientes da interface de rede.	53
Figura 24 – SSIM e DMOS estimado e medido para um atraso fixo de 0.18s para o cenário <i>Dp3</i>	64
Figura 25 – SSIM e DMOS estimado e medido para um atraso fixo de 0.18s para o cenário <i>Du7</i>	65
Figura 26 – SSIM e DMOS estimado e medido para uma velocidade fixa de 36 Mbytes/s para o cenário <i>Dp3</i>	66
Figura 27 – SSIM e DMOS estimado e medido para uma velocidade fixa de 36 Mbytes/s para o cenário <i>Du7</i>	67
Figura 28 – Relação entre SSIM medido e estimado.	68
Figura 29 – Relação entre DMOS medido e estimado.	69
Figura 30 – Relação entre SSIM e DMOS estimado.	69

Índice de Tabelas

Tabela 1 – Descrição dos parâmetros de cabeçalhos indicados para a pilha protocolar TS/PES/NAL.....	14
Tabela 2 – QP's, PSNR e bitrate dos conjuntos de vídeo experimental.	16
Tabela 3 – Coeficientes do polinómio de grau 3 para as imagens P e B.....	17
Tabela 5 – Pesos de saída e tendências de entrada/saída utilizados n as equaações 6 e 7.	21
Tabela 6 – Estrutura e exemplo de relatório gerado pelo Módulo <i>Report</i>	49
Tabela 7 – Tabela com a caracterização do tráfego utilizado nos cenários de rede utilizados nas simulações.	60
Tabela 8 – Parâmetros de qualidade de rede que se fizeram varias nos cenários de teste.	61

Índice de Abreviaturas

ANN	<i>Artificial Neural Network, p.18</i>
CABAC	<i>Context-Adaptive Binary Arithmetic Coding, p.18</i>
CC	<i>Countinuity Counter, p. 14</i>
CSV	<i>Comma-Separated Values, p. 26</i>
DMOS	<i>Differential Mean Opinion Score, p. 18</i>
DSL	<i>Digital Subscriber Line, p. 24</i>
DVB	<i>Digital Video Broadcasting, p. 2</i>
FDR	<i>Frame Degradation Rate, p. 20</i>
FR	<i>Full Reference, p. 8</i>
GOP	<i>Group of Pictures, p. 3</i>
GPON	<i>Gigabit-capable Passive Optical Networks p. 24</i>

GUI	<i>Guide User Interface, p. 30</i>
HP	<i>Hight Profile, p. 18</i>
IDE	<i>Integrated Development Environment, p.31</i>
ILR	<i>Information Loss Rate, p. 20</i>
IPTV	<i>Internet Protocol Television, p. 1</i>
LAN	<i>Local Area Network, p. 24</i>
MOS	<i>Mean Opinion Score, p. 48</i>
NAL	<i>Network Abstraction Layer, p. 13</i>
NR	<i>No-Reference, p. 5</i>
PCAP	<i>Packet Capture, p. 2</i>
PES	<i>Packetized Elementary Stream, p. 13</i>
PLP	<i>Packet-Layer Parameters p. 19</i>

PLR	<i>Packet Loss Rate, p. 20</i>
QoE	<i>Quality of Experience, p. 1</i>
QoS	<i>Quality of Service p. 1</i>
SSIM	<i>Structural Similarity Index, p. 8</i>
TS	<i>Transport Stream, p. 1</i>
V+P	<i>Video Plus Depth, p. 35</i>
VCL	<i>Video Coding Layer, p. 13</i>
VQA	<i>Video Quality Assessment, p.5</i>

Capítulo 1

1 Introdução

Devido à crescente evolução das tecnologias 3D, nos próximos anos o IPTV 3D pode vir a tornar-se um serviço largamente distribuído pelos operadores [1], desta forma é importante que estes possam ter uma forma fácil e barata de monitorizar a qualidade do serviço que oferecem. Decorrente disso, surge a necessidade de os fornecedores de serviço conhecerem o impacto que o desempenho das suas infraestruturas têm sobre este tipo de serviço. Uma forma de o conseguir é descobrir de que forma um conjunto de métricas de qualidade de serviço (QoS - *Quality of Service*) influenciam a qualidade de experiência (QoE - *Quality of Experience*), permitindo garantir uma qualidade aceitável aos consumidores finais.

A avaliação de vídeo digital transmitido em redes propensas a erros tem vindo a obter maior importância para que os operadores possam oferecer serviços de qualidade. Atualmente ainda é difícil obter informação de qualidade percetual de vídeo 3D em diversos pontos da rede, principalmente se não existir o recurso a um vídeo de referência.

Com recurso a uma ferramenta que permita a monitorização destes serviços utilizando técnicas de monitorização não intrusivas, os fornecedores de serviço podem com um menor custo conhecer as suas necessidades para o correto fornecimento do seu serviço. Para o vídeo 2D generalizadamente já existem atualmente diversas soluções, o mesmo não se aplica o vídeo 3D.

A possibilidade de instalação de um monitor de qualidade de experiência de vídeo 3D em qualquer ponto da rede vai permitir aos operadores adaptar o seu serviço e infraestrutura de forma a garantir uma determinada qualidade de experiência aos seus utilizadores, por exemplo para uma aplicação de IPTV. E tornar a monitorização desse serviço mais barata e menos intrusiva.

Efetivamente já existe um protótipo capaz de avaliar a qualidade de vídeo 3D em formato TS (*Transport Stream*) e codificado em H.264/MVC [15]. Este protótipo permite monitorizar a

qualidade de vídeo 3D mas tem uma abrangência limitada a um único modelo de avaliação da qualidade de vídeo.

Tendo em conta que o modelo implementado no protótipo permite a monitorização de vídeo transmitido sobre redes *broadcast* e tendo em vista as necessidades futuras de monitorização de qualidade dos fluxos de vídeo que se prevê dominarem uma grande percentagem da capacidade das redes IP, fará todo o sentido expandir a abrangência do monitor para permitir a monitorização da qualidade do vídeo em redes IP para diversos métodos de codificação e transmissão de vídeo 3D.

Tendo em conta o problema e as limitações de monitorização de vídeo 3D, este trabalho tem como objetivo e desenvolver métodos para aceder à qualidade de experiência (QoE) de vídeo 3D, sem acesso a vídeos de referência, e que sejam válidos para diversos modelos de rede e codificação de vídeo.

Sem recurso à utilização de vídeos de referência, este trabalho também tem como objetivo verificar as métricas de qualidade de rede que afetam a qualidade de serviço (QoS) e a QoE.

Para alcançar estes objetivos partindo do protótipo já referido na seção anterior foram realizadas as seguintes etapas:

- A. Generalização de um protótipo de qualidade de vídeo TS desenvolvido para monitorizar vídeos 3D transmitidos nas tradicionais redes DVB (*Digital Video Broadcasting*). Este protótipo apenas permite monitorar a qualidade de vídeo H.264/MVC. Esta generalização passa a permitir uma integração simples de um novo modelo e cenário de monitorização.
- B. Criar um novo cenário de monitorização para avaliar a qualidade de vídeo TS transmitido em redes IP. Para o efeito foi utilizado o modelo de avaliação teórico já implementado no protótipo e que permite a avaliação de qualidade de vídeo TS codificado em H.264/MVC. O protótipo terá que permitir a avaliação de vídeo no formato PCAP (*Packet Capture*) permitindo a avaliação de vídeo capturado numa interface de rede.
- C. Criar de um novo cenário de monitorização que permite a avaliação de vídeo no formato codificado em H.264/AVC *video-plus-depth* sobre redes IP. Este foi criado com base num

modelo de avaliação teórico criado com recurso a uma rede neuronal e utiliza um conjunto de parâmetros que necessitam de uma análise mais complexa aos pacotes de vídeos.

- D. Criar um módulo que permita fazer monitorização em tempo real e integrá-lo no protótipo. No final será possível utilizar o modelo indicado no ponto C) para fazer avaliação em tempo real. Criar um conjunto de funcionalidades que permitam gerir de uma forma simples a proveniência dos dados dados de entrada (vídeos), a gestão da estrutura GOP, verificação de *logs* para deteção de erros/eventos, notificação de um broker gestor do tráfego e a execução a partir da linha de comando (sem recurso á interface gráfica).
- E. Com recurso a um simulador de rede G.1050 obter o impacto que as condições de rede definidas num conjunto de casos de teste podem ter na qualidade vídeo estimada para vídeos naquelas condições.

No final dos desenvolvimentos foram ainda realizados um conjunto de simulações que permitem obter vídeos resultantes do impacto de um conjunto de cenários que refletem várias condições de rede. Através dos resultados de qualidade estimados pelo protótipo para vídeo no formato *video-plus-depth*, foi possível descrever graficamente a variação da qualidade ao longo da taxa de erros e velocidade da tecnologia de acesso e o atraso da rede.

Este documento é constituído por cinco capítulos e 3 anexos. Neste primeiro capítulo foi apresentada a problemática envolvida em torno da monitorização da qualidade de vídeo 3D e que justifica a realização deste trabalho. São ainda apresentados os objetivos e descritas sucintamente todas as etapas necessárias para os alcançar.

No capítulo 2 é feita uma revisão da literatura que visa introduzir ao leitor os principais conceitos e modelos associados à monitorização de qualidade de vídeo não só para apresentar o estado da arte mas para que a compreensão de todo o desenvolvimento seja facilitada. Serão também apresentados com maior detalhe os modelos de avaliação de qualidade já desenvolvidos para o vídeo 3D. Será ainda apresentado o simulador de rede G.1050 utilizado para realizar um conjunto de simulações que permitirão introduzir um padrão de erros a vídeos introduzidos na simulação.

Serão feitas referências ao seu funcionamento e pressupostos de utilização para que seja melhor clarificada a fase de testes e obtenção de resultados.

No capítulo 3 será apresentado todo o processo de desenvolvimento do protótipo tendo como base um protótipo já existente. Este está dividido em várias seções onde é apresentado e justificado passo-a-passo todo o processo de generalização e criação de novos modelos de avaliação de vídeo 3D.

No capítulo 4 são apresentados os resultados obtidos para a qualidade dos vídeos resultantes das simulações realizadas com o simulador G.1050 e que permitem medir o desempenho do modelo de avaliação utilizado monitorizar vídeo em redes IP. No último capítulo será apresentada a conclusão e alguns aspetos que podem ser desenvolvidos no futuro com base neste trabalho.

Capítulo 2

2 Modelos de Avaliação de Vídeo

Neste capítulo é apresentado um enquadramento genérico dos modelos de avaliação de vídeo fazendo-se referência a metodologias de desenvolvimento de modelos de avaliação de qualidade de vídeo. Serão apresentados também os formatos de representação de vídeo 3D e alguns trabalhos desenvolvidos na avaliação da qualidade de vídeo 2D. Por fim serão apresentados o processo de desenvolvimento de dois modelos sem referência ou do termo internacional *no-reference* (NR) de avaliação de qualidade de vídeo 3D.

Na seção 2.3 será apresentado o modelo de avaliação de qualidade de vídeo também conhecido como VQA (*Video Quality Assessment*) implementado num protótipo de um monitor de qualidade de vídeo que será utilizado como base para o desenvolvimento de um novo cenário de monitorização, em que o mesmo modelo VQA é utilizado para monitorizar vídeo em redes IP. Na seção 2.4 será apresentado um modelo VQA que será desenvolvido de raiz nesse protótipo do monitor.

2.1 Formatos de representação de vídeo 3D

A representação do vídeo 3D inclui várias utilizações do formato de codificação H.264/AVC padrão e das suas extensões. No âmbito deste trabalho, os formatos de codificação considerados com maior potencial de utilização no futuro são o H.264/MVC e *vídeo plus depth* [1]. Estes exigem taxas de transmissão não muito elevadas, garantindo a qualidade do vídeo. Em cada um destes formatos, a informação de profundidade é representada e codificada utilizando abordagens completamente distintas.

Em [2] podemos ver uma descrição destes dois formatos e um estudo mais alargado para o formato H.264/MVC. Este formato de codificação fornece uma representação compacta de múltiplas vistas que são criadas simultaneamente por várias câmaras, permitindo a predição de vistas fornecendo uma significativa eficiência de codificação face à transmissão de todas as vistas em separado. É ainda possível a descodificação parcial de uma vista que corresponde ao vídeo 2D. Esta característica permite a compatibilidade deste formato com os sistemas de televisão 2D já existentes. A abordagem de codificação MVC pretende explorar a correlação temporal que existe entre as várias vistas e também entre imagens vizinhas, permitindo a redução da taxa de bits sem sacrificar a qualidade total dos vídeos reconstruídos. Na figura 1 é possível perceber a estrutura de codificação de um vídeo 3D contendo três vistas. São representadas 5 unidades de acesso, cada uma contendo um componente por vista [2]. Uma unidade de acesso ou *Acess Unit* corresponde a uma imagem 3D completa.

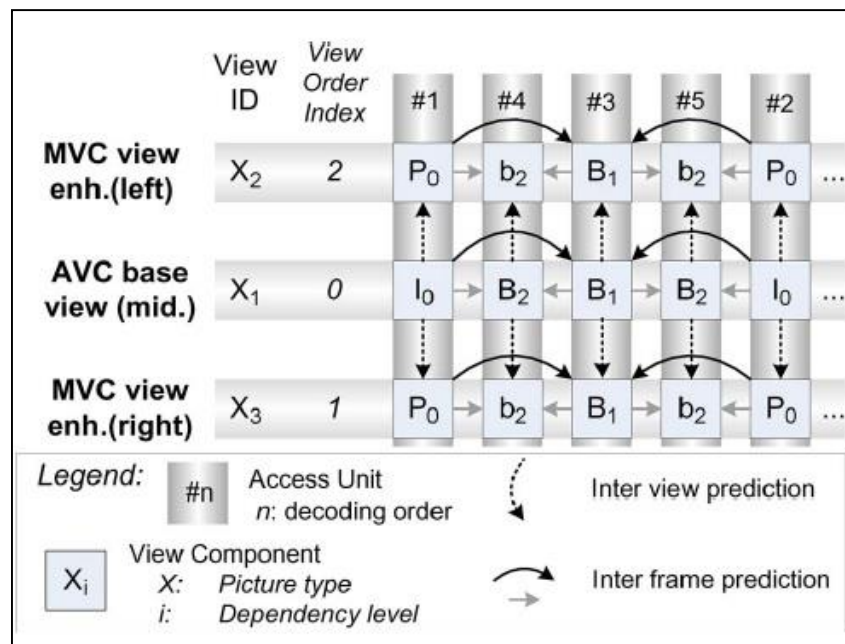


Figura 1 – Estrutura da predição MVC em duas vistas independentes a partir de uma vista de base [2].

Utilizando uma codificação assimétrica das vistas é possível diminuir a taxa de transmissão das aplicações de vídeo stereo. Neste tipo de codificação, cada uma das vistas é codificada com qualidades diferentes. Contudo, este método torna o processo de codificação mais complexo na medida em que o impacto visual 3D da vista degradada depende da percepção humana e não pode ser determinada com as medidas convencionais [3].

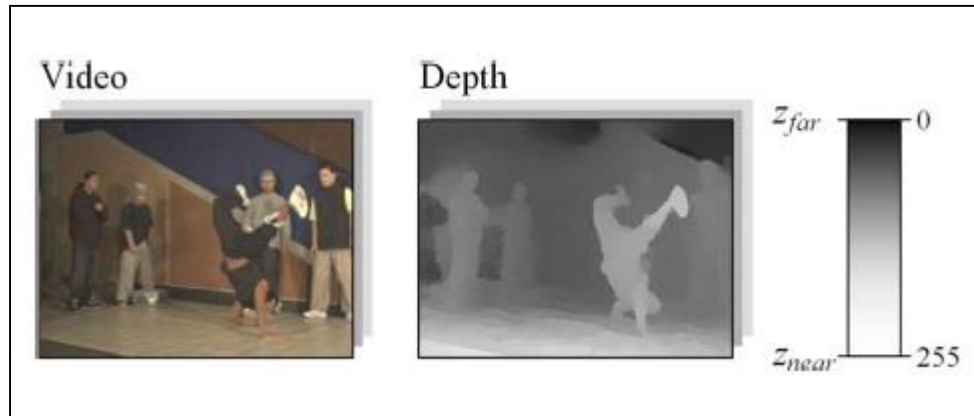


Figura 2 – Estrutura interna da representação do formato vídeo plus depth [2]

Outra forma de codificar vídeo stereo utilizando apenas duas vistas, é utilizar o formato *video plus depth* [2]. Neste formato a composição do vídeo 3D baseia-se num fluxo de vídeo auxiliar com a imagem de profundidade que, tal como no vídeo 2D, também pode ser codificado com H.264/AVC, e esta pode ser comprimidos de forma muito eficiente. Na figura 2 podemos ver como é representada a informação tridimensional do vídeo em formato *video plus depth*.

2.2 Métricas e Modelos de Avaliação de vídeo

As métricas de qualidade mais conhecidas são a relação sinal ruído PSNR (Peak signal-to-noise ratio) e o erro quadrático médio MSE (Mean Squared Error). O PSNR é muitas vezes utilizado para a avaliação do desempenho das normas de compressão de vídeo (como por exemplo H.264/AVC). As equações do PSNR e do MSE são apresentadas e descritas de seguida [4]:

$$MSE = \frac{1}{N} \sum_{n=1}^N (X_n - Y_n)^2 \quad (1)$$

$$PSNR = 10 \log_{10} \frac{L^2}{MSE} \quad (2)$$

Onde X_n e Y_n correspondem respetivamente aos valores de pixéis da imagem original e distorcida. N é o número total de pixéis da imagem. Na equação PSNR, L representa o número máximo de pixéis que é possível armazenar numa imagem. Apesar destes métodos estarem associados a uma computação simples o que as torna atrativas para muitas aplicações, não consideram a perceção do sistema visual humano, que é um fator importante a considerar na medição da qualidade do vídeo 3D [4].

Uma métrica de avaliação de vídeo que tem em conta essa preocupação é a apresentada em [5]. Nesta é extraída a informação estrutural que é relevante para a qualidade do vídeo, tendo em conta o sistema visual humano HVS (*Human visual system*):

$$SSIM(x, y) = \frac{(2u_x u_y + c_1)(2\sigma_{xy} + c_2)}{(u_x^2 + u_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (3)$$

Os argumentos x e y são respetivamente a imagem original e distorcida. Os parâmetros u_x e u_y referem-se respetivamente à média de x e y . Os parâmetros σ_x , σ_y e σ_{xy} representam respetivamente a variância de x , variância de y e a covariância entre o x e o y . As constantes c_1 e c_2 são utilizadas para evitar que $u_x^2 + u_y^2$ e $\sigma_x^2 + \sigma_y^2$ se aproximem de zero, respetivamente. O modelo SSIM (*Structural Similarity Index*) tem vantagens significativas em imagens com pouco ou nenhum movimento, pois para além de considerar a distância física entre os pixéis considera também as distorções estruturais. Contudo, em muitas sequências com maiores movimentos perde alguma precisão [27].

Estes tipos de métricas podem ser utilizados para desenvolver modelos de avaliação de qualidade de vídeo. Estes serão mencionados neste documento pelo termo VQA.

De acordo com a local da rede onde se pretende monitorizar o vídeo e dependência de informação proveniente do vídeo original, os modelos VQA podem ser classificados por: *Full-*

Reference (FR), *Reduced-Reference* (RR) e *No-Reference* (NR). Nos modelos FR a qualidade de vídeo é obtida pela comparação do vídeo distorcido com o vídeo de referência. Como é necessário o sinal original, este modelo não é adequado para monitorizar a qualidade de vídeo em locais remotos, onde o sinal de referência não está disponível. Métodos RR permitem determinar a qualidade de vídeo utilizando uma quantidade de informação reduzida do sinal de referência. No entanto é necessário transmitir pela rede um conjunto de informações do sinal de referência por um canal paralelo. Os métodos NR permitem medir a qualidade do vídeo codificado, em qualquer ponto remoto da rede sem que para isso necessitem de informações do vídeo original e sem necessidade de canais adicionais. Na figura 3 indicam-se os pontos da rede onde é possível aplicar cada um dos métodos referidos [3], [5], [7] e [15].

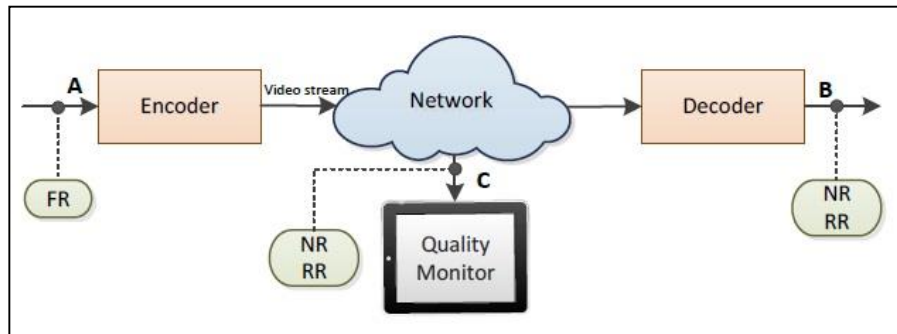


Figura 3 – Pontos de rede onde é possível instalar cada um dos tipos de modelos existente [15].

O tipo de modelo que interessa utilizar neste trabalho é NR, pois permitem avaliar a qualidade de vídeo sem recurso ao vídeo original tornando a monitorização do vídeo transparente para o consumidor final, automática e permanente. Nas próximas seções descrevem-se os modelos VQA teóricos desenvolvidos com base neste tipo de modelo.

Os modelos VQA sem referência (NR) podem ser classificados de acordo com o nível de informação de transporte e vídeo que consideram na avaliação da qualidade do vídeo:

Media-layer model: - utiliza informação ao nível dos pixels; necessita de ter disponíveis as imagens de vídeo decodificadas e como tal tem uma aplicabilidade mais reduzida já que está associado a uma complexidade computacional mais elevada.

Packet-layer model: - utiliza informação existente no cabeçalho dos pacotes de transporte do vídeo e parâmetros de rede (atraso, variação do atraso, etc.). Também pode utilizar informação ao nível do CODEC utilizado. Na seção 2.3 será apresentado um modelo de avaliação deste tipo.

Bitstream-layer model: - utiliza a mesma informação que o modelo *packet-layer* e explora também a informação de codificação incluída no vídeo comprimido. Pelo facto de utilizar uma maior quantidade de informação, a precisão da estimativa de qualidade pode ser maior. Um modelo que utiliza esta quantidade de informação é apresentado na seção 2.4.

Hybrid model: - Para além de toda a informação utilizada nos modelos *bitstream-layer*, pode utilizar também informação ao nível dos pixels, permitindo uma maior precisão na medição da qualidade mas mais uma vez à custa de uma maior complexidade computacional.

A figura 4 mostra os vários tipos de modelos VQA e a sua relação com o tipo de informação que utilizam.

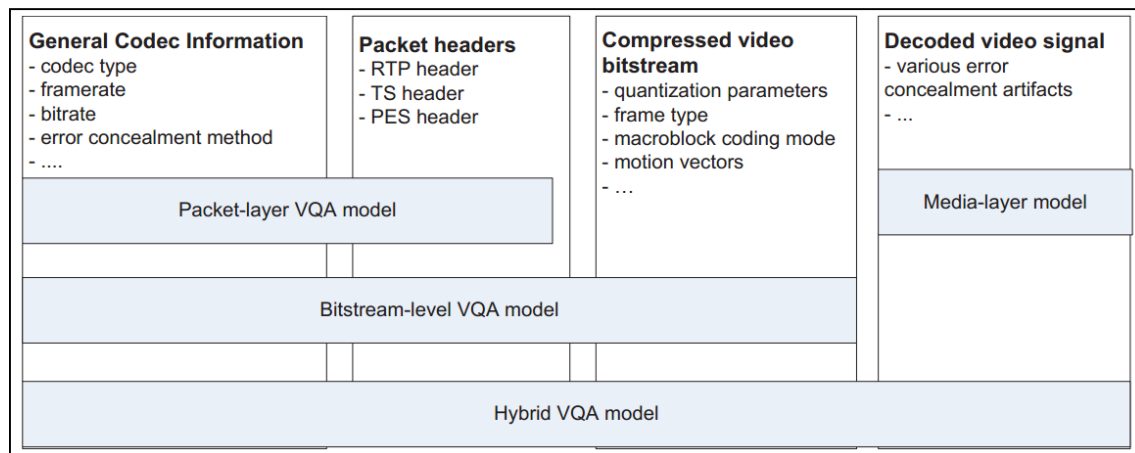


Figura 4 – Classificação dos vários tipos de modelos VQA sem referência (NR) de acordo com o tipo de informação de transporte e vídeo que lhes estão associadas [16].

Os modelos de maior interesse para este trabalho são os modelos bitstream e packet-layer pois estão associados a uma reduzida complexidade computacional face a outros modelos sem referência permitindo uma maior flexibilidade para serem introduzidos em diversos pontos da rede. Estes dois modelos são descritos com maior detalhe em [3]. Neste, embora se esteja a falar

de modelos de avaliação de vídeo 2D são de grande importância, pois a partir deles é possível extrapolar os modelos para vídeos 3D [3]. Esta referência tem uma compilação de um conjunto de metodologias de desenvolvimento de modelos de avaliação de vídeo e da sua aplicabilidade.

Existem vários modelos e ferramentas para avaliar a qualidade de serviço (QoS) ou experiência (QoE). Por exemplo em [9] é descrita uma ferramenta útil para planeamento de QoE/QoS que permite garantir a qualidade de serviço oferecida ao utilizador final. Este modelo utiliza como dados de entrada os parâmetros de rede da aplicação, vídeo e/ou informação dos equipamentos recetores. Este modelo especifica uma função que permite estimar a qualidade do vídeo considerando todos os parâmetros de entrada já descritos. O modelo necessita que se obtenham previamente um conjunto de coeficientes que variam conforme os *codecs* utilizados. Portanto antes de utilizar este modelo, será necessário obter os respetivos coeficientes para um determinado formato de vídeo. A saída deste modelo corresponde à qualidade estimada para aqueles dados de entrada e tipo de codecs associado. Na figura 5 podemos ver uma parte da arquitetura utilizada neste algoritmo para a avaliação da qualidade do vídeo, em que V_q corresponde á qualidade de vídeo estimada.

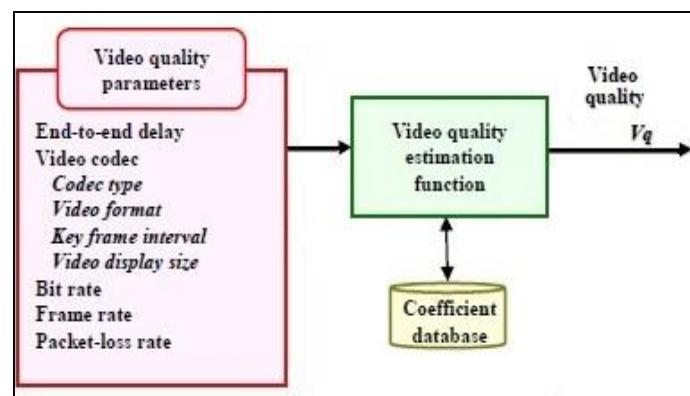


Figura 5 – Framework para o modelo de avaliação de qualidade de vídeo [9]

Outro exemplo é apresentado na figura 6 que mostra graficamente a arquitetura de um modelo de avaliação de qualidade de vídeo 2D, para um serviço IPTV transmitida sobre TS/RTP/UDP/IP e baseado num modelo *packet-layer*. Este modelo é apresentado em [10]. Sucintamente o modelo é

composto por 4 unidades: unidade de extração de parâmetros responsáveis por analisar os cabeçalhos dos pacotes e extração de parâmetros necessários para a estimação da qualidade do vídeo; a unidade de estimação de qualidade por distorção de codificação; a unidade de estimação de qualidade de vídeo afetada que foi afetado pela perda de pacotes, refletindo o padrão de perda de pacotes; uma base de dados com os coeficientes utilizados para a estimação de qualidade para o vídeo que utiliza um determinado *codec*. Cada formato de codificação implica um conjunto de coeficientes distinto. Uma aplicação que utiliza um modelo *bitstream* é descrita em [11].

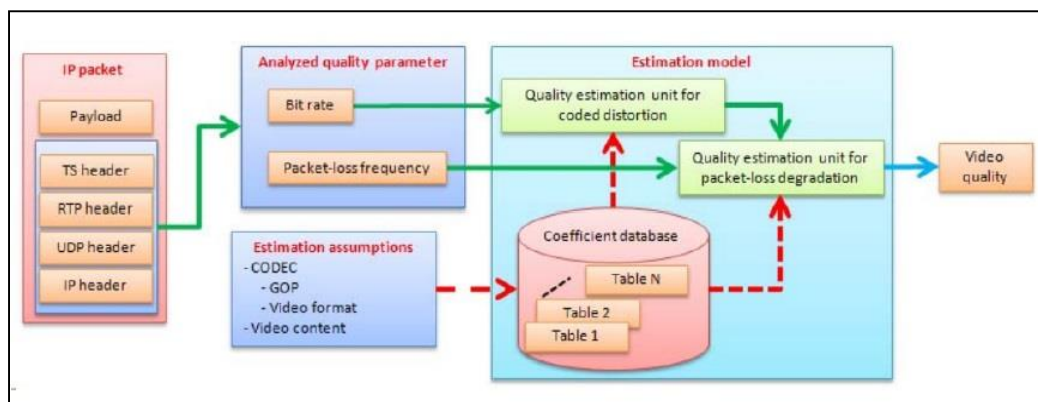


Figura 6 – Arquitetura de um modelo de avaliação para uma aplicação IPTV [10]

Também podem ser encontrados vários estudos na literatura sobre o impacto que os parâmetros de rede podem exercer sobre a qualidade do vídeo 2D. Por exemplo em [12] é apresentada uma aplicação que utiliza informação proveniente do cabeçalho RTP e onde se pode concluir que, a redução da QoS numa secção de uma imagem pode-se propagar devido aos mecanismos utilizados para fazer predição temporal de informação das imagens obtidas.

Nas próximas seções serão apresentados dois modelos VQA desenvolvidos para a avaliação da qualidade do vídeo 3D. Serão apresentados o enquadramento, processos de desenvolvimento e o desempenho alcançado de cada um dos modelos. Essa informação será importante para perceber melhor a implementação destes modelos em *software*, num protótipo de um monitor de avaliação de qualidade de vídeo que foi desenvolvido nesta dissertação e que será apresentado no capítulo 3.

2.3 Avaliação de vídeo MVC em redes broadcast

Nesta seção é descrito o modelo VQA packet-stream para vídeo 3D no formato MVC [15], [28] e [29]. A descrição deste modelo é importante devido ao facto de o protótipo desenvolvido neste trabalho ter sido construído com base num protótipo existente e onde foi implementado este modelo de avaliação. Como tal, esta compreensão é fundamental para se perceber a descrição do protótipo existente, a sua generalização e implementação de um novo cenário de monitorização que também usa este modelo VQA para a avaliação da qualidade do vídeo 3D.

No que se refere ao fornecimento de um serviço de vídeo 3D, as operadoras ou fornecedores de serviço podem utilizar vários tipos de redes tais como redes *broadcast* e redes IP. Em ambas as situações pode ser utilizado o MPEG-2 *Transport Stream* (TS) embora a sua utilização seja imperativa em redes *broadcast* [13], em redes IP pode ser uma opção que introduz alguma redundância [14].

O modelo de avaliação de vídeo 3D apresentado nesta seção pressupõe uma rede *broadcast*. Ou seja, uma rede em que a transmissão de vídeo corresponde à pilha protocolar TS/PES/NAL. As unidades de dados da camada de codificação de vídeo ou o termo inglês VCL (*Video Coding Layer*) é fragmentado em diferentes pacotes NAL (*Network Abstraction Layer*), que por sua vez é fragmentado e diferentes pacotes PES (*Packetized Elementary Stream*) que compõe o *payload* de um pacote TS.

Na tabela 2.1 são descritos os parâmetros dos cabeçalhos dos pacotes TS, PES e NAL utilizados pelo modelo para calcular a estimativa da qualidade do vídeo. É descrito ainda o significado de cada parâmetro.

Este modelo tem como objetivo permitir a estimação da degradação da qualidade de vídeo resultante de erros introduzidos nos pacotes TS e que deram origem a perdas numa imagem isolada. Essa medição é feita pela diferença entre o SSIM da imagem de referência (sem erros) e da imagem degradada. O método de encobrimento utilizado para o cálculo do SSIM da imagem com erros é a cópia de imagem. Este modelo utiliza uma estimativa para prever o tamanho de

cada imagem perdida de cada tipo de imagem B, P e I. Esta é estimada com base na média do tamanho das imagens corretamente recebidas, para cada tipo e vista. A estrutura GOP do vídeo é considerada num parâmetro estático já que é tipicamente definido previamente pelo fornecedor dos serviços.

Tabela 1 – Descrição dos parâmetros de cabeçalhos indicados para a pilha protocolar TS/PES/NAL.

Pacote de transporte de vídeo	Nome do parâmetro	Descrição do parâmetro
TS	Synchronisation Byte	Byte utilizado para delimitar o início de um pacote TS
	Payload Unit Start Indicator	Indica se a carga útil do pacote TS corresponde ao início de um dado do pacote PES
	PID	ID do pacote para identificar o pacote TS
	Continuity counter	Contador do número de pacotes que têm carga útil
PES	Packet start code prefix	Utilizado para determinar o início de um pacote PES
	PES packet length	Indica o tamanho dos dados existentes no payload do PES
NAL	NAL Ref Idc	Indica o conteúdo de um pacote NAL e é utilizado para reconstruir imagens a partir da predição de outras imagens
	NAL Unit Type	Indica o tipo de carga útil de um pacote NAL
	Non IDR Flag	Indica se a carga útil de um pacote NAL transporta um IDR-imagem
	View ID	Especifica se um pacote NAL pertence apenas a um cenário multi-vista

Na figura 7 é apresentada a arquitetura do modelo de estimação do SSIM que resulta da aplicação no modelo VQA teórico, os parâmetros que resultaram da análise e extração de informação dos cabeçalhos dos pacotes. T1 representa o parâmetro *Continuity Counter* (CC) do cabeçalho TS.

Este permite detetar se ocorreram perda de pacotes. P_1 é obtido a partir da análise aos cabeçalhos PES e fornece o tamanho da imagem, o nome do parâmetro correspondente é o *PES packet length*. A partir de A_1 , é possível identificar qual o tipo de imagem que está identificada no cabeçalho NAL (*NAL_unit_type*). U_1 e U_2 referem-se respetivamente ao tamanho e estrutura GOP do vídeo estimado.

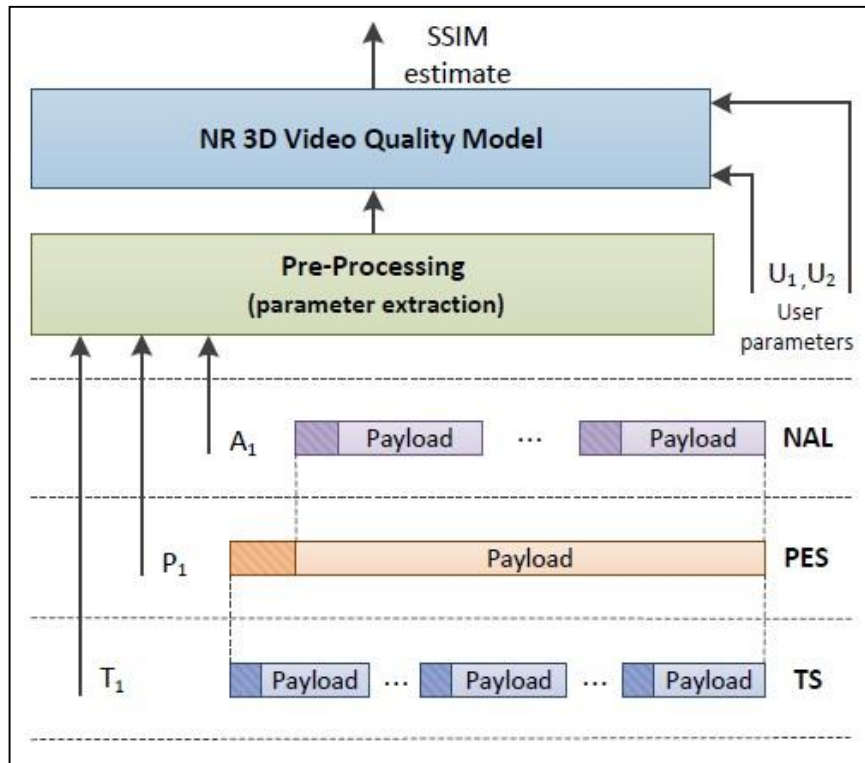


Figura 7 – estrutura do modelo packet layer utilizado para estimativa de qualidade objetiva (SSIM) para um sistema broadcast [15].

Outro aspeto que normalmente é tido em conta para o cálculo da degradação de uma imagem é a distorção de codificação. Esta refere-se à distorção introduzida no vídeo original antes de ser codificado. Neste modelo não é tido em conta este aspeto, e como tal considera-se o valor máximo do SSIM para uma imagem corretamente recebida.

Nas referências feitas para este modelo, é possível consultar os vídeos utilizados e a sua estrutura. Não são apresentados esses dados neste documento devido ao facto de não serem relevantes no

processo de desenvolvimento de um novo cenário de monitorização que utiliza este modelo VQA, como será apresentado mais à frente.

A equação genérica que define o modelo VQA obtido é apresentado na equação 4 onde $dSSIM$ referido como Δ_{SSIM} corresponde à diferença entre o $SSIM$ da imagem de referência e o $SSIM$ calculado para a imagem recebida, assumindo o método de encobrimento do erro cópia de imagem; n é o grau do polinómio; L_{fs} é o tamanho da imagem perdida em bytes; e p_n é os coeficientes do polinómio.

$$dSSIM_p = p_n L_{fs}^n + \dots + p_2 L_{fs}^2 + p_1 L_{fs} + p_0 \quad (4)$$

Este modelo foi estudado para três conjuntos de vídeos, codificados com os valores dos parâmetros QP (*quantization parameter*), PSNR e bitrate apresentados na tabela 2.

Tabela 2 – QP's, PSNR e bitrate dos conjuntos de vídeo experimental.

Parâmetros de codificação	I-QP	P-QP	B-QP	PSNR (dB)	Bitrate (Kb/s)
Conjunto 1	26	28	28	42	2847
Conjunto 2	28	30	30	41	2225
Conjunto 3	30	32	32	40	1727

Para cada um desses conjuntos, foram obtidos os valores para os coeficientes polinomiais p_n a p_0 tendo em conta a equação 5. Nesta, x representa o tipo de imagem, Δ_{SSIM} representa o $dSSIM$ apresentado na equação 4, EF_{SSIM} e D_{SSIM} representam respetivamente o $SSIM$ da imagem de referência e da imagem com recebida.

$$\Delta_{SSIM}_x = EF_{SSIM}_x - D_{SSIM}_x \quad (5)$$

Os coeficientes de terceiro grau obtidos para o modelo são descritos na tabela 3. Aplicando ao modelo os coeficientes de polinómios de diferentes graus (linear, quadrático e cúbico), o autor conclui que a terceira ordem consegue estimar com razoável precisão o tamanho das imagens perdidas, permitindo obter um RMSE em torno de 0.1 e uma correlação de *Pearson* em torno de 0.8.

Como já foi referido, este modelo é utilizado na primeira versão de um protótipo de um monitor de qualidade de vídeo 3D que pode ser utilizado para as condições que acabaram de ser apresentadas para este modelo VQA. Esse protótipo será utilizado como ponto de partida para a introdução de novos modelos e cenários em que possam ser aplicados. O modelo VQA que se descreve na próxima seção é um deles.

Tabela 3 – Coeficientes do polinómio de grau 3 para as imagens P e B.

Conjunto de vídeos	Frames P	Frames B
Conjunto 1	p0 = 0.05365	p0 = -1.90E-02
	p1 = 9.29E-06	p1 = 5.78E-05
	p2 = -1.19E-09	p2 = -3.77E-09
	p3 = 4.22E-14	p3 = 2.57E-14
Conjunto 2	p0 = 4.74E-03	p0 = 1.30E-02
	p1 = 1.78E-05	p1 = 2.57E-05
	p2 = -1.87E-10	p2 = 1.07E-08
	p3 = 2.72E-14	p3 = -1.50E-12
Conjunto 3	p0 = -0.03292	p0 = 2.01E-02
	p1 = -2.92E-05	p1 = 2.13E-05
	p2 = 3.86E-08	p2 = 2.23E-08
	p3 = -3.28E-12	p3 = -3.69E-12

2.4 Avaliação de vídeo V+D em redes IP propensas a erros

Nesta seção é apresentado outro modelo, criado com o objetivo de permitir monitorizar a qualidade de vídeo 3D num qualquer ponto arbitrário de uma rede propensa a erros, utilizando apenas informação dos cabeçalhos dos pacotes (modelos NR) [17]. Este será implementada nesta dissertação num protótipo do monitor de qualidade de vídeo 3D e os seus resultados serão objeto de estudo para investigar a relação entre parâmetros de qualidade de experiência e qualidade de serviço em redes IP.

O modelo pressupõe uma representação do vídeo 3D no formato vídeo-mais-profundidade ou do termo inglês *video-plus-dept* (V+D). Como referido no início deste capítulo, esta representação compreende a transmissão do vídeo 2D e uma componente de profundidade. Este pode ser considerada uma representação mais complexa que outras devido à necessidade de realizar operações complexas de síntese dos vídeos obtidos de diversas câmaras [16]. Cada componente é geralmente transmitida por dois streams H.264/AVC separados. É a partir da monitorização da informação do tráfego de profundidade que este modelo pretende estimar a qualidade objetiva e subjetiva do vídeo 3D.

Este modelo foi desenvolvido a partir de uma rede neuronal que a partir de um conjunto de parâmetros de entrada, cerca de sete, processados numa janela deslizante de 10s permite obter um valor para o SSIM. Estes parâmetros são extraídos e calculados a partir da informação existentes nos cabeçalhos da informação de profundidade. Para além de permitir obter valores para a qualidade objetiva, este modelo fornece também o valor da degradação da qualidade subjetiva (DMOS - *Differential Mean Opinion Score*) diretamente a partir dos valores obtidos para o SSIM.

A pilha protocolar onde este modelo tem aplicabilidade é IP/UDP/RTP/NAL/SLICE ou seja diz respeito á transmissão de vídeo em redes IP, como por exemplo um serviço de IPTV 3D. No entanto o conjunto de informação necessária para a estimação da qualidade objetiva e subjetiva é apenas necessário analisar os cabeçalhos dos pacotes RTP, até ao nível do SLICE.

É importante referir que a rede neuronal utilizada para a obtenção deste modelo foi devidamente treinada com recurso a milhares de fluxos de profundidade garantindo uma boa generalização do modelo e está devidamente validada, tendo obtido valores para o RMSE de 0.0058 e uma correlação de *Pearson* acima do 0.9211.

Para o treino e aprendizagem da ANN (*artificial neural network*) foram utilizadas sequências de vídeos de profundidade com cerca de 10s, codificados com o *software* H.264/AVC *Joint Model Reference Software* com configuração *High Profile* (HP). As imagens correspondentes a um GOP dependem apenas das imagens de referência existentes no mesmo GOP, closed-GOP.

Foram utilizadas 30 imagens por GOP com uma estrutura fixa IBBBP...P, oito *slices* por imagem com o mesmo número de macro blocos e entropia de codificação CABAC. Cada *slice* foi encapsulada numa NAL e transportada apenas uma NAL por pacote RTP. Ao contrário do que aconteceu com o modelo apresentado na seção 2.3, as características do vídeo são importantes não só para o desenvolvimento deste em *software* mas também para a avaliação e análise dos resultados obtidos nos testes e simulações (capítulo 4).

Na figura 8 é apresentado o diagrama funcional do modelo de estimação do SSIM e DMOS. Este é constituído por dois modelos: o primeiro que corresponde ao modelo criado com base na rede neuronal (ANN) e permite obter a estimativa para o SSIM ($SSIM_p$) com base num conjunto de parâmetros extraídos dos cabeçalhos dos pacotes *PLPs* (*packet-layer parameters*); o segundo modelo permite calcular uma estimativa para o cálculo do DMOS ($DMOS_p$). O processo de desenvolvimento utilizado para criar o modelo DMOS é detalhadamente descrito em [17].

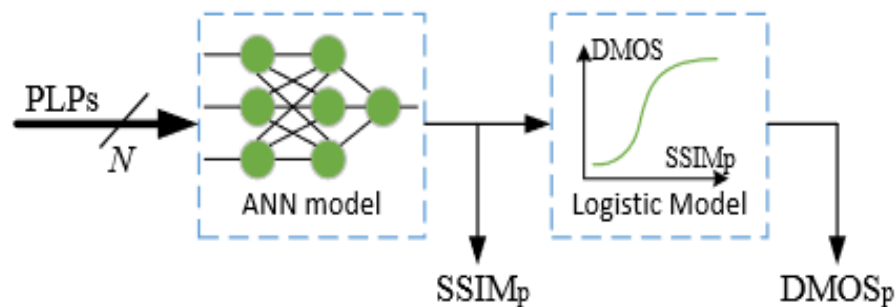


Figura 8 – Arquitetura do modelo utilizado para a estimação do SSIM e DMOS [17].

Os valores para o $SSIM_p$ foram deduzidos a partir da aprendizagem feita pela ANN quanto ao SSIM calculado pelas sequências de vídeos com erros face às imagens de referência. Milhares de sequências de vídeos foram utilizadas, como resultado de um padrão de erros gerado por um modelo Gilbert-Elliot utilizando o *Transmitter-Simulator software* [25].

O valor estimado do SSIM é calculado de acordo com as equações 6 e 7 onde N é o número de parâmetros de entrada PLP_i , H é uma constante que representa o número de nós internos

da rede neuronal e que são utilizados para o cálculo da qualidade objetiva, w e b são coeficientes obtidos ao longo do processo de aprendizagem do modelo.

$$SSIM_p(z) = \sum_{j=1}^H (w_j^{out} \cdot \tanh(z_j)) + b^{out} \quad (2.6)$$

$$z_j(PLP) = \sum_{i=1}^N (w_{ji}^{in} \cdot PLP_i) + b_j^{in} \quad (2.7)$$

Os PLPs são calculados a partir da análise dos cabeçalhos RTP, NAL e SLICE, sendo definidos e classificados da seguinte forma:

- *PLR (Packet Loss Rate)*: representa o rácio de *slices* de profundidade que foram perdidos, assumindo um slice por pacote RTP. É detetado pela perda de pacotes RTP numa janela temporal de 10s.
- *ILR (Information Loss Rate)*: representa a quantidade de informação perdida devido à perda de pacotes. A quantidade de bytes perdida de uma imagem, tal como o modelo apresentado na seção 2.1, pode ser estimado a partir dos tamanhos das imagens de cada tipo (I, P e B) que foram recebidas no passado.
- *FDR (Information Loss Rate)*: representa o rácio de imagens degradadas devido à perda de uma imagem ou a erros de propagação causados por imagem de referência. Para calcular este parâmetro é necessário ter conhecimento prévio da estrutura GOP de forma a prever qual a dependência entre as diversas *imagens* (I, P e B).

Foram utilizados PLPs tendo em conta cada tipo de *imagem*, já que o seu tamanho e número por GOP podem ter grandes variações. Por esta razão foram utilizados no modelo da rede neuronal 7 PLPs: PLR_I , PLR_P , PLR_B , ILR_I , ILR_P , ILR_B , FDR . Desta forma é possível obter uma maior estimativa para o cálculo do $SSIM_p$.

O desempenho deste modelo foi testado com 1, 3, 6 e 7 PLPs e com 1 até 15 *nós internos* (H). Obtiveram-se valores mais elevados de PLCC quando se utilizam 7 PLP, verificando-se pequenas diferenças entre 6 e 7 PLPs. São também para estes que se obtém um menor valor para o RMSE. Relativamente aos valores para os parâmetros H (*hidden nodes*) com base nos dados de desempenho obtidos, o autor propõe a utilização de H=3 com 7 PLP já que oferece uma

generalização suficiente tendo em conta que se utilizaram os vídeos típicos neste tipo de trabalhos e cujas características exploram os principais desafios que se colocam à degradação da qualidade devido a erros de transmissão e propagação.

Nas tabelas 4 e 5 são apresentados os valores ideais a serem utilizados para parametrizar o modelo de estimação de qualidade objetiva e que foram utilizados no protótipo desenvolvido nesta dissertação. Este modelo pressupõe a utilização de 3 nós internos e 7 PLPs.

Tabela 4 – Pesos de entrada para parametrização da equação 7.

Parâmetros i e $w_{j,i}^{in}$	PLP						
	PLR_I	PLR_P	PLR_B	ILR_I	ILR_P	ILR_B	FDR
i	1	2	3	4	5	6	7
$w_{1,i}^{in}$	-0.2385	-0.2256	0.0403	-0.0604	-0.3774	0.3762	-0.1133
$w_{2,i}^{in}$	0.0471	0.0089	-0.0141	-0.0395	-0.0186	0.0028	0.0005
$w_{3,i}^{in}$	-0.1611	-0.0352	0.0195	0.1265	0.0711	-0.0157	-0.0026

Tabela 5 – Pesos de saída e tendências de entrada/saída utilizados nas equações 6 e 7.

Índice i	Pesos de saída w_j^{out} e tendências entrada/saída b_j^{in}/b^{out}			
	j	w_j^{out}	b_j^{in}	b^{out}
	1	-0.0358	-3.2400	0.7256
	2	0.2298	0.7037	
	3	0.1182	1.2160	

A equação 2.8 apresenta o modelo logístico deduzido de acordo com a recomendação ITU-R BT.500-11 [18] e com base num conjunto de avaliações subjetivas feitos a partir dos vídeos com erros e de referência realizadas por um conjunto de voluntários.

$$DMOS_p = \frac{73.89}{1 + e^{173.6(SSIM_p - 0.9951)}} \quad (2.8)$$

Este modelo logístico será utilizado pelo protótipo do monitor desenvolvido nesta dissertação para se obterem os valores de qualidade de experiência correspondentes ao SSIM estimado pelo modelo VQA e que implementa em *software* as equações 6 e 7

2.5 Modelo de rede para avaliação de vídeo

Nesta seção será descrito o modelo de rede utilizado neste trabalho, para aplicar um padrão de erros aos vídeos 3D permitindo obter um conjunto de dados de entrada no protótipo do monitor de qualidade de experiência desenvolvido. Para criar um padrão de erro que reflita o comportamento das redes atuais e aceite como referência utilizou-se o modelo de rede ITU-T G.1050 [26]. Este capítulo faz um enquadramento do funcionamento e configurabilidade deste simulador com base na definição da própria norma, facilitando assim uma melhor compreensão do impacto dos erros introduzidos nos vídeos utilizados no estudo de modelos, assim como os cenários de testes utilizados para introdução de padrões de erros.

O modelo de rede ITU-T G.1050 [26] corresponde a um simulador de uma rede IP típica que possa ser utilizada para avaliar o desempenho de qualquer tráfego que no caso desta dissertação será o vídeo 3D que possa ser transmitida sobre a mesma. Este utiliza um formato de dados de entrada baseando-se em ficheiros PCAP. Este modelo pretende simular o comportamento atual das redes IP por meio da definição de uma arquitetura de rede em que os parâmetros de rede refletem os padrões atuais típicos.

Um conjunto de casos de teste vem anexado ao simulador de referência que está disponível e é de utilização livre. Estes refletem vários cenários de qualidade de serviço e podem ser utilizados para fazer avaliação dos vídeos que se pretendem estudar. Para tal é necessário adicionar a cada caso de teste o vídeo em questão. Também é possível a definição de novos cenários, quer pela alteração dos parâmetros de rede, quer pela alteração do tráfego interferente utilizado em cada cenário. Com os resultados da simulação é ainda possível imolar o comportamento dos vídeos obtidos após a simulação e que refletem o estado da rede para cada cenário.

A última versão desta norma baseia-se na simulação de eventos discretos ao contrário da anterior que se baseava num modelo matemático e onde era mais difícil personalizar e criar novos cenários. A utilização de eventos discretos resulta das considerações tidas pelos autores da norma, que consideram que a utilização de um padrão de erros baseado em probabilidades já não se aplica às redes IP [26]. Embora continuem a ser utilizadas probabilidades para descrever alguns parâmetros de rede, o padrão de erros introduzido como um todo resulta dos fenómenos que ocorrem em redes reais tais como as filas de espera ou descarte de pacotes num *router*. Pretendeu-se com estas alterações dotar o simulador de um comportamento mais realista. Este simulador foi implementado com a linguagem de programação C++.

Neste simulador, a rede IP é modelada como uma rede de elementos de base que pode ser vista na figura 9.

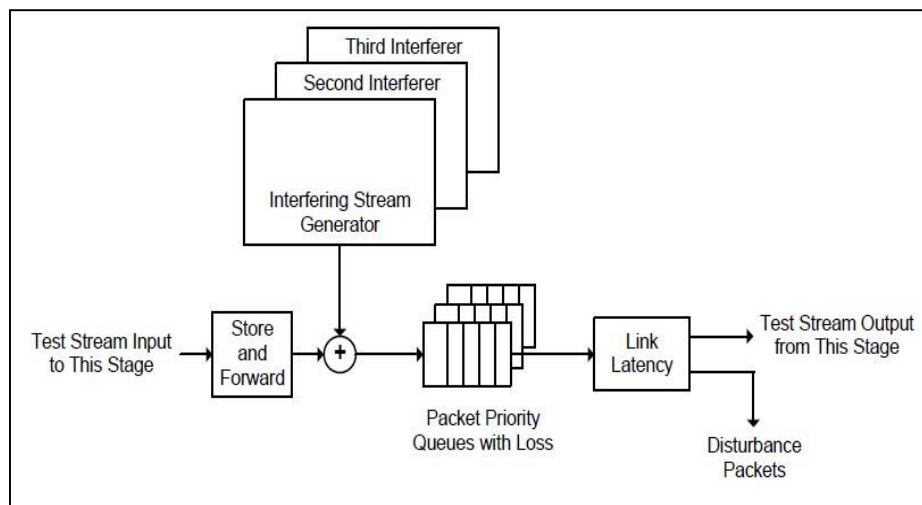


Figura 9 – Modelo base de um elemento de rede [26].

O processamento do simulador pode ser descrito numa direção como se descreve, no entanto este é um modelo bidirecional:

- O objeto gerador de pacotes introduz os pacotes para dentro do simulador. Características como tempos de chegada, tamanho dos pacotes e outro tráfego interferente são lidos do ficheiro PCAP;

- O switch recebe os pacotes e determina qual a porta de saída correspondente;
- Os cabos ligam as portas de um switch até a entrada do próximo;
- O processo repete-se para todos os pacotes, cabos e switch;
- Os dados de partida e chegada dos pacotes são armazenados para posterior análise, assim como a informação sobre perdas de pacotes.

Todos os elementos de rede do simulador têm como base o elemento apresentado na figura 9. Estes elementos podem ser: *switches* núcleo; *router* de periferia de rede; tecnologias de acesso DSLAM e GPON OLT; os equipamentos terminais *modem DSL (Digital Subscriber Line)* e GPON; *firewall*; LAN; e os fios entre os dispositivos. E apenas as tecnologias DSLAM e GPON OLT implementam várias filas de pacotes, uma por cada prioridade de QoS.

Os cenários de teste originais incluem uma arquitetura de rede core-to-LAN, permitindo estudar fluxos de informação na direção servidor-cliente. A figura 10 mostra esse modelo de rede. É um modelo deste tipo que é utilizado por defeito nos cenários de teste incluídos no simulador e nas simulações realizadas nesta dissertação. Um modelo LAN-to-LAN pode ser criado com a concatenação de dois modelos core-to-LAN.

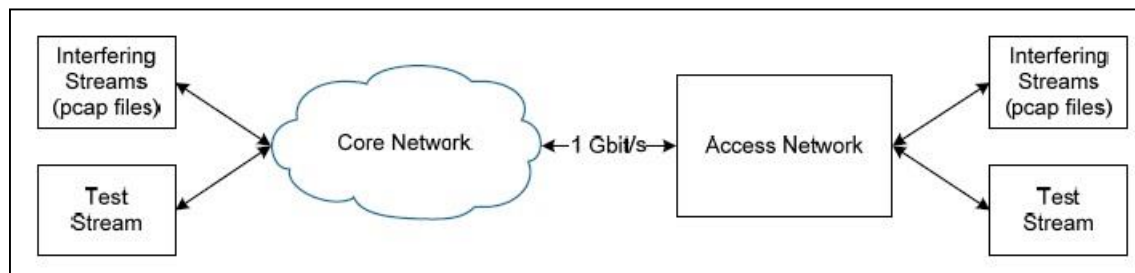


Figura 10 – Modelo de rede core-to-LAN. Modelo utilizado por defeito nos cenários de teste incluídos no simulador [26].

Todos os arquivos de tráfego utilizados neste simulador estão no formato PCAP tal como definido no ficheiro pcap.h da biblioteca libpcap [21]. O tráfego interferente disponibilizado foi capturado em dispositivos terminais de clientes. Estes arquivos foram anonimamente capturados por um conjunto de investigadores a partir de uma variedade de fontes reais. De forma a garantir a privacidade e direitos de conteúdo real estes ficheiros não têm *payload*. No entanto o tamanho

do pacote com o *payload* é considerado pelo simulador. Os fluxos de tráfego introduzidos no simulador também podem ser utilizados com carga útil.

Antes de utilizar um ficheiro PCAP numa simulação é necessário realizar um conjunto de procedimentos de forma a garantir que aquele tráfego descreve um comportamento aproximado ao que teria na realidade: é necessário dividir essa captura nos fluxos *upstream* e *downstream*, bem como realizar um estudo da largura de banda utilizada por esse fluxo.

Outros dados de entrada para além do tráfego em formato PCAP são os valores dos parâmetros dos elementos de rede já mencionados. A tabela C1 do anexo C é apresentada uma tabela com a gama de valores possíveis para cada elemento de rede onde se pressupõe uma rede core-to-LAN. As únicas tecnologias de acesso possíveis até ao momento são DSL (*Digital Subscriber Line*) ou GPON (*Gigabit-capable Passive Optical Networks*). Aos fluxos PCAP é associada uma prioridade para permitir fazer a gestão do tráfego que circula pela rede. O simulador possui 7 níveis de prioridade, contudo apenas 3 são utilizados: prioridade 1 corresponde à prioridade mais alta; 7 a prioridade mais baixa; e a prioridade intermédia 2, que é a tipicamente utilizada para gerir tráfegos IPTV.

Os cenários de teste previamente preparados e incluídos no simulador descrevem as características do modelo de rede IP de acordo com 3 perfis:

- **Perfil A (*Well-Managed Network*)**- rede em que não existem links lotados e em que todos os links do router de fronteira implementam QoS. Estes testes incluem serviços de voz e video geridos;
- **Perfil B (*Partially Managed Network*)**- rede que minimiza o número de links lotados e com um ou mais links com QoS nos routers fronteira. Estes serviços incluem uma mistura de voz e video gerido e não gerido;
- **Perfil C (*unmanaged Network*)**- rede que inclui links que podem lotar e um ou mais links com serviço de QoS nos router de fronteira. Estes serviços incluem transmissão de dados não geridos e outros serviços de video.

Os parâmetros específicos do núcleo de rede, tráfego interferente e parâmetros da rede de acesso formam um caso de teste e estão alinhados com o nível de perfil adotado. Os ficheiros que correspondem ao tráfego interferente correspondem a uma mistura realista de tráfegos típicos.

Nos casos de teste definidos, a formatação dos nomes seguem uma lógica bem definida. O primeiro carácter do nome define a tecnologia de acesso considerada. Quando é um D significa que a tecnologia de acesso é DSL, G refere-se à tecnologia GPON. As taxas de acesso e níveis de erros correspondem a valores típicos do ano 2010.

O segundo carácter corresponde ao nível de perfil considerado. é utilizado o w (*well-managed network*) para definir o perfil A, o p (*partial-managed network*) para definir o perfil B, u (*unmanaged network*) para definir o perfil C e c (*core-only*) para definir cenários realizados unicamente no núcleo de rede. O terceiro, e último, carácter é um dígito cujo valor é proporcional à dificuldade do caso de teste. As tabelas C2 e C3 do anexo C apresentam as tabelas com os elementos e parâmetros utilizados nos cenários de testes para o perfil B e C.

Os cenários de testes apresentados nas tabelas C2 e C3 são utilizados como base para a criação dos cenários de teste utilizados neste trabalho. Um dos fluxos foi substituído pelo fluxo de teste e outros parâmetros de rede foram modificados.

Como resultado de cada simulação são gerados os ficheiros PCAP recebidos no destino, e pela comparação entre os ficheiros PCAP de entrada e saída são obtidos os valores para o atraso e perda de pacotes. Estes valores são registados em ficheiros de extensão .out e no formato CSV (*comma-separated values*), para que possam ser facilmente tratados utilizando folhas de cálculo. Os ficheiros PCAP de saída que contém *payload* podem ser utilizados para que se reproduza o estado final de um fluxo de tráfego que está a ser testado. Os ficheiros CSV são depois utilizados para gerar gráficos que apresentam os atrasos, e perdas de pacotes. Histogramas com a variação do atraso e funções de distribuição cumulativas são também geradas de forma a permitir uma análise das características de rede.

Embora este modelo esteja continuamente a ser melhorado e de se aproximar mais das redes IP atuais, este ainda tem algumas limitações. Este simulador não permita especificar um conjunto de

casos de testes com vários tipos de condições de rede para redes sem fios. Por este motivo, nesta dissertação será realizada um conjunto de adaptações aos modelo, nomeadamente a configuração de alguns parâmetros do simulador, que permita simular este tipo de redes. Isto porque se pretende neste trabalho obter resultados que sejam válidos com a utilização de femto-células.

Capítulo 3

3 Protótipo implementado

Neste capítulo é apresentado o desenho e descrição do desenvolvimento em *software* de um protótipo que permite a avaliação da qualidade de vídeo 3D com base em modelos de qualidade de vídeo 3D. Inicialmente será descrito brevemente o ponto de partida deste desenvolvimento, o protótipo de um monitor de qualidade existente. Este corresponde ao protótipo desenvolvido no decorrer dos trabalhos realizados para o desenvolvimento do modelo teórico apresentado na seção 2.3, e apenas funcional para redes DVB com vídeo 3D no formato H.264/MVC.

De seguida, será descrito o processo de generalização desse monitor, para que se possa adicionar facilmente outros modelos de avaliação de acordo com a configuração do utilizador. Serão então descritas a implementação de dois novos modelos de avaliação de vídeo: o primeiro que é uma evolução do modelo apresentado na seção 2.3 mas num cenário em que o vídeo em formato TS é transmitido sobre uma rede IP; e o segundo que trata do modelo teórico apresentado na seção 2.4, que foi implementado de raiz para esta dissertação. Por fim, será descrita a implementação de um módulo que permite a monitorização em tempo real para qualquer modelo que faça a transmissão do vídeo sobre redes IP.

Ao longo da exposição vai-se fazendo um enquadramento dos conteúdos apresentados anteriormente e os principais desafios que ocorreram no processo de implementação.

3.1 Arquitetura da solução

Na figura 11 é apresentada a arquitetura da solução desenvolvida. Seguindo o fluxo das setas verifica-se que vários dados de entrada alimentam o monitor. Estes podem ser: vídeos, que são

ficheiros com uma sequência de imagens; ficheiros em formato PCAP e uma interface de rede que será gerida pelo módulo de monitorização em tempo real. É apresentado o ficheiro TS porque até ao momento apenas são analisados ficheiros deste tipo, mas um outro qualquer ficheiro pode ser utilizado. Qualquer uma das fontes de dados pode fornecer pacotes de vídeo a qualquer cenário de monitorização.

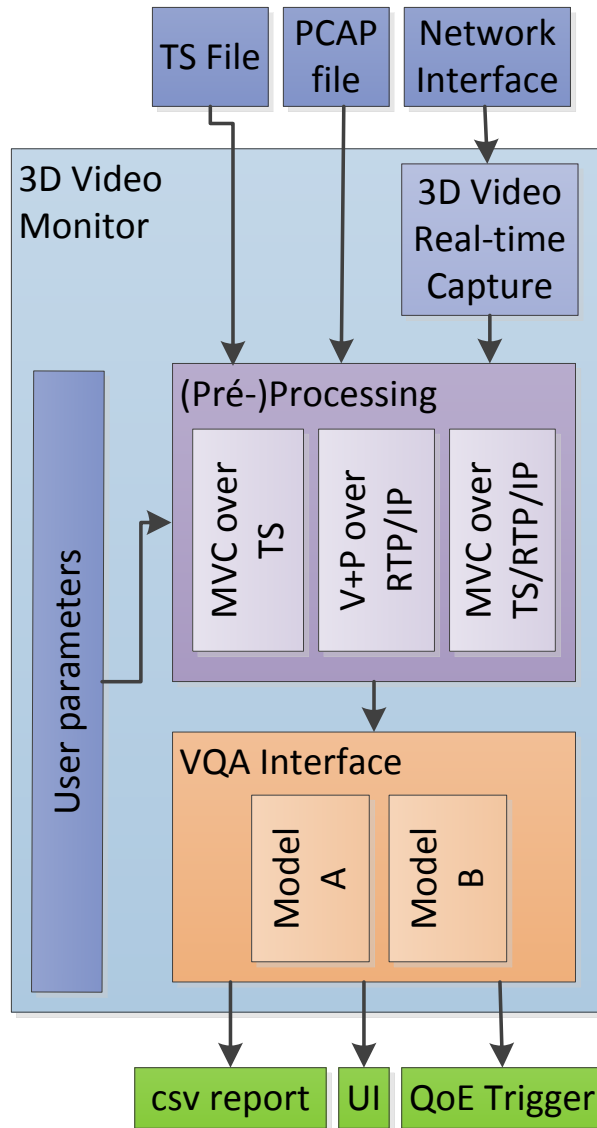


Figura 11 – Estrutura lógica do monitor de vídeo 3D.

Estes cenários de monitorização são configurado com recurso a uma interface gráfica onde são parametrizadas as características dos vídeos que serão avaliados, para que em cada um dos casos se possam aplicar corretamente os modelos de avaliação definidos.

Em cada um dos cenários de monitorização os pacotes começam por ser analisados e um conjunto de parâmetros é extraído dos cabeçalhos dos pacotes. Estes parâmetros vão ser aplicados aos modelos de avaliação de vídeo que terão como resultado uma estimativa da qualidade de vídeo. Depois de serem obtidos valores para a qualidade de vídeo, estes valores são transmitidos para o processo principal que trata de atualizar a interface gráfica para que o utilizador possa ficar com uma perceção da qualidade que está a ser medida. Esses valores são ainda registados num relatório, juntamente com os parâmetros extraídos dos cabeçalhos e que deram origem a esses resultados de qualidade. Pode ainda ser necessário notificar um *broker* de que um determinado limite de qualidade foi detetada para que o monitor possa interagir com outros componentes de gestão de rede.

3.2 Descrição do protótipo existente

Nesta seção é apresentado o protótipo utilizado como ponto de partida para o desenvolvimento do novo protótipo. Pretende-se dar um enquadramento da estrutura do *software* para que se tenha uma melhor compreensão de todo o processo de generalização e a adição de novos modelos e módulos que foram necessários desenhar e implementar para esta tese.

Descrição do Software

O protótipo que foi tido como ponto de partida é constituído por dois processos. No primeiro são analisados os cabeçalhos dos pacotes de vídeo e é calculada uma estimativa para a sua qualidade. O outro é o processo principal que gere todos os outros processos e inclusive a interface utilizador GUI (*Guide User Interface*) onde são configurados os parâmetros de avaliação de

vídeo e onde são apresentados os gráficos com os resultados da qualidade objetiva obtidos. Estas funcionalidades foram separadas entre os dois processos para que a execução das funcionalidades não se bloqueiem mutuamente. Na figura 12 é apresentado um diagrama com o fluxo de informação entre os diferentes processos.

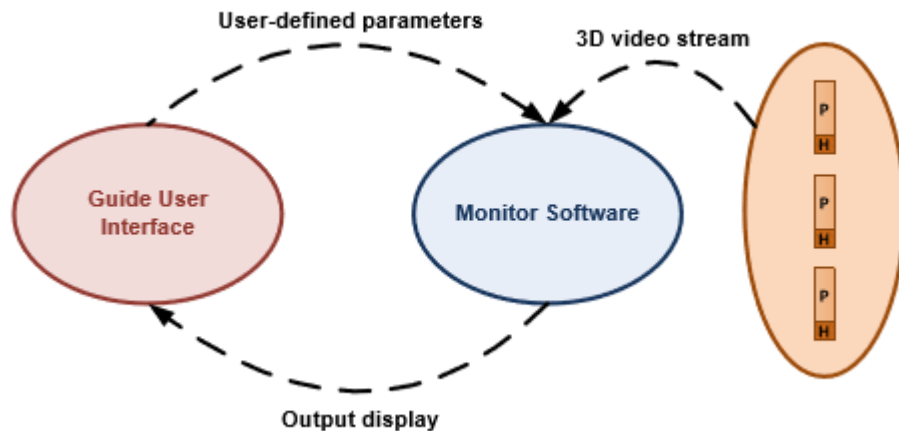


Figura 12 – Diagrama com fluxo de informação criado entre os dois processos utilizados no protótipo do monitor [15].

O protótipo foi implementado na linguagem C++ utilizando o ambiente de desenvolvimento *Qt Creator*. Este ambiente de desenvolvimento ou IDE (*Integrated Development Environment*) foi escolhido devido à sua facilidade em desenvolver interfaces com o utilizador, por utilizar uma linguagem baseada em objetos e por ser uma linguagem de programação relativamente eficiente. A linguagem orientada a objetos facilita o mapeamento dos conceitos de monitorização de qualidade e de rede para *software*. Nesta primeira versão do *software*, foram desenvolvidas as classes: *Monitor*, *Access_unit*, *TS*, *PES*, *NAL* e *Frame*.

Processo de Monitorização

A classe *Monitor* é a principal função do *software* pois é responsável pela monitorização da qualidade do vídeo, ou seja, extração de parâmetros dos pacotes TS e estimativa da qualidade de vídeo objetiva e subjetiva. Um conjunto de unidades NAL é referida como *access unit*. A

descodificação de cada *access unit* corresponde a uma imagem descodificada. Esta é constituída por duas imagens, uma para cada vista. Para cada um dos pacotes que é necessário extrair informação do cabeçalho, foi criada uma classe: TS, PES e NAL. A classe *Frame* foi criada para materializar cada uma das vistas independentes.

O objeto *Monitor* tem um método principal designado *process*. Este é responsável por executar todo o processo de monitorização e tratamento dos resultados: análise e extração de parâmetros, cálculo da estimativa da qualidade objetiva, registos dos valores de qualidade e parâmetros recolhidos num relatório no formato CSV e transmissão de informação para o processo com a interface utilizador para atualização gráficos com a qualidade.

O objeto *access unit* é utilizado no método *process* do *Monitor* para realizar a extração e análise de parâmetros dos pacotes que são necessários para a avaliação do vídeo como a deteção de perda de pacotes TS e obtenção de outros parâmetros extraídos das vistas base e auxiliar.

O método *readAU* é a função principal da classe *access unit*, nesta é analisada toda a informação referente às duas componentes de cada vista. Esta informação vai sendo armazenada no objeto *Monitor* para que este possa mais tarde, medir a qualidade do vídeo após o processamento de uma determinada janela temporal. Nesta é realizada a análise do vídeo utilizando a classe TS responsável por analisar cada um dos componentes PES e NALU recorrendo às classes e métodos correspondentes. É na classe TS que são detetadas a perda de pacotes TS e é a partir da classe NALU que é possível identificar o tipo de cada uma das imagens I, P e B.

Interface com Utilizador

A interface com o utilizador tem o objetivo o vídeo e os parâmetros que caracterizam o vídeo e necessários para o cálculo da estimativa de qualidade e apresentação de informação gráfica onde é apresentada a qualidade global e temporal do que está a ser medido. Na figura 13 é apresentada a interface com o utilizador. Nesta são identificados os seguintes componentes:



Figura 13 – Interface gráfica do protótipo com indicação dos vários componentes [15].

Input parameters- permite especificar os parâmetros de codificação do vídeo e que normalmente são fixados por um fornecedor de serviços: *Error Concealment* é utilizado para especificar a técnica de cancelamento do erro e que influencia a qualidade do vídeo; *GOP size* é necessário para especificar o tamanho de um grupo de imagens que têm uma determinada interdependência; *Reference Frames* utilizada para identificar as *imagens* de referência existentes no GOP;

Temporal Window utilizada para especificar o tamanho da janela temporal utilizada para a avaliação da qualidade do vídeo; *Frame rate* é utilizado para especificar o rácio de imagens apresentadas por segundo.

Instantaneous Video Quality Status- nesta componente é apresentada a qualidade subjetiva do vídeo 2D e 3D instantânea.

TS Layer Specific Output- nesta componente é apresentado o número de pacotes TS perdidos que foram detetados no vídeo (*Lost TS packets*); e o número de erros detetados no vídeo (*Errors*).

Video Quality History- nesta componente são apresentados 4 gráficos: *3D Video Quality* e *2D Video Quality*. Estes apresentam respetivamente o histórico da qualidade subjetiva do vídeo 3D e 2D; *Error Bursts* que apresenta o histórico do tamanho médio dos erros em rajada; e *Packet Loss Ratio* que apresenta o histórico do rácio de pacotes perdidos ao longo do tempo.

Tools Bar- nesta componente é possível executar operações sobre os streams de vídeo que se pretende monitorizar, tais como abrir e correr. Tem ainda o botão Save para guardar o relatório com a qualidade do vídeo medida ao longo de cada janela temporal.

TS File Path- aqui é apresentado o caminho do vídeo selecionado e que se pretende monitorizar.

É ainda importante referir, que ao longo do processo de monitorização o processo de monitorização do vídeo vai registando num relatório em formato CSV a informação extraída e qualidade medida em cada janela deslizante: o tipo, tamanho e número de *imagens* do tipo I, P e B, qualidade objetiva (SSIM) e subjetiva (DMOS) para cada janela temporal.

3.3 Generalização do Software

Será feita uma apresentação da nova arquitetura, sem entrar em detalhes de implementação mas que permita apresentar o trabalho desenvolvido e justificar as opções tomadas.

O protótipo tinha uma estrutura que permite a análise e a estimativa da qualidade do vídeo 3D apenas para um modelo que avalia a qualidade do vídeo TS no formato MVC. Para aumentar a sua aplicabilidade e permitir a adição de novos modelos de avaliação para outros tipos de vídeo que suportem outras formas de representação de vídeo 3D referida, por exemplo V+P, e tornou-se necessário generalizar a sua estrutura funcional. Contudo no protótipo existente teria que se garantir o funcionamento do monitor para o modelo de avaliação de vídeo já existente.

Nesta primeira fase é necessário redesenhar o protótipo existente e garantir o funcionamento do modelo de avaliação de vídeo já existente.

O conceito associado ao objeto *Monitor* foi alterado. Esta continha a função principal *process* que realizava a análise do vídeo e fornecia o valor da estimativa da qualidade objetiva e subjetiva calculada para uma janela deslizante para um único tipo de vídeo num único cenário de monitorização. Este passa agora a ser visto como uma entidade genérica, que permite a seleção e configuração de um cenário de rede com o qual se pretende utilizar para avaliar a qualidade do vídeo num determinado formato de codificação H.264. Esta classe permite apontar para um determinado cenário de rede e torna-se o único ponto de acesso entre o processo da interface utilizador e o processo de monitorização.

As funcionalidades da antiga classe *Monitor* que eram utilizadas para monitorizar a qualidade do vídeo TS codificado com o formato MVC e que é transmitido numa rede *broadcast* dão origem a um novo objeto com a designação *MVC_BroadcastSystems*. O nome da classe faz referência ao formato de codificação de vídeo 3D (MVC) e a um cenário de rede que corresponde a um conjunto de protocolos que são utilizados para na transmissão do vídeo.

A nova classe *Monitor* mantém a API de interação com a interface utilizador. Um conjunto de outras funcionalidades foram adicionadas para permitir a interação com o cenário de monitorização específico, neste caso o *MVC_BroadcastSystem*.

Foi então criada uma nova classe abstrata *NetworkScenario* para normalizar a interação entre a nova classe *Monitor* e um cenário de rede existente ou que se pretenda adicionar. Isto quer dizer que a nova classe *Monitor* passa a manipular um cenário de rede utilizando uma API especificada

nesta nova classe. Isto permite criar uma independência entre o processo de gestão e configuração dos cenários de rede e os cenários de monitorização propriamente ditos, que fazem a monitorização da qualidade do vídeo. O *Monitor* pode assim manipular um *NetworkScenario* onde é considerado um formato de codificação e um cenário de monitorização. Um *NetworkScenario* concreto é distinguido dentro do *Monitor* através de uma macro que identifica univocamente aquele tipo de *NetworkScenario* entre o processo principal e o *Monitor*. Na figura 14 é apresentado um diagrama de classes que descreve a relação entre a classe *Monitor* e um *NetworkScenario*. É ainda apresentado o cenário de monitorização de rede concreto *MVC_BroadcastSystem*. O conceito *NetworkScenario* ou cenário de monitorização de rede é descrito com maior detalhe na seção 3.4.

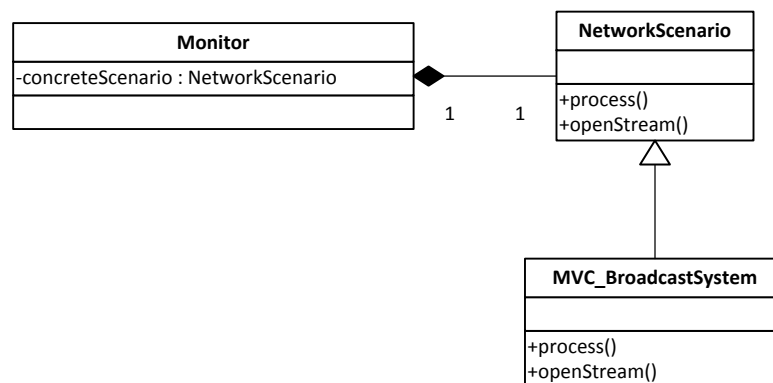


Figura 14 – Relação entre a classe *Monitor* e o cenário de monitorização concreto *MVC_BroadcastSystem*.

Os métodos mais importantes que é necessário implementar num *NetworkScenario* concreto são:

- *process()*: este método é onde se realiza a análise e extração de parâmetros dos cabeçalhos dos pacotes que são depois utilizados pelo modelo VQA teórico que se está a utilizar. Fornece o valor da qualidade objetiva e subjetiva do vídeo 2D e 3D para cada janela deslizante;
- *openStream()*: este método é utilizado para abrir um descritivo para o ficheiro de entrada que contém os vários pacotes contidos no vídeo.

Depois de normalizar a interação entre a interface utilizador com um cenário de monitorização através do objeto *Monitor*, que utiliza uma API definida na classe abstrata *NetworkScenario*, o modelo descrito na seção 2.3, que já estava implementado no protótipo, está pronto para ser utilizado.

O conceito VQA já foi introduzido no capítulo 2, também foi introduzido no protótipo. Este corresponde a um modelo de avaliação teórico criado para estimar a qualidade de vídeo num determinado cenário com base num conjunto de parâmetros extraídos pelo processo de monitorização. Pode surgir a necessidade de substituir o modelo teórico utilizado num determinado cenário de monitorização, por exemplo, para passar a refletir o impacto de um novo padrão de tráfego de rede. Também pode ser implementado um novo modelo e utiliza-lo num cenário de monitorização já implementado sendo que para tal este tem de considerar os mesmos parâmetros que são extraídos dos cabeçalhos dos pacotes. Para que tal seja possível a classe VQA também foi generalizada.

Com a introdução desta alteração, um *NetworkScenario* concreto passa a estar associado a um VQA que pode ser alterado ao longo do tempo ou para diferentes aplicações permitindo a reutilização de todo o código de análise e extração de informação. O VQA de um cenário de monitorização passa a ser especificado no momento da criação de um *NetworkScenario* no *Monitor*. Para implementar o modelo de avaliação teórico existente (modelo apresentado na seção 2.3) foi criada a classe *VQA_MVC_BroadcastSystems* que implementa a interface VQA. Para utilizar um modelo de avaliação de qualidade de vídeo em *MVC_BroadcastSystem* basta configura-lo na inicialização do *Monitor*. Na figura 15 é apresentada a relação entre a interface VQA e concreta, e a relação entre esta e o cenário de monitorização *MVC_BroadcastSystem*.

Os métodos mais importantes que um objeto do tipo VQA têm de implementar são:

- *calcVQ2D* – calcula para um conjunto de parâmetros de entrada o valor estimado para a qualidade objetiva do vídeo 2D;
- *calcVQ3D* – calcula para um conjunto de parâmetros de entrada o valor estimado para a qualidade objetiva do vídeo 3D.

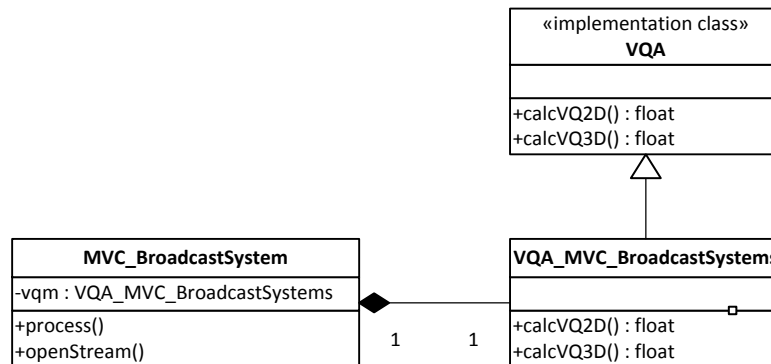


Figura 15 – Relação entre o cenário *MVC_BroadcastSystem* e o respectivo VQA concreto.

Com esta nova arquitetura, para criar um novo cenário é necessário criar uma classe que implemente a classe abstrata *NetworkScenario*, e adiciona-la como uma opção do *Monitor*, para que possa ser utilizada através da interface utilizador. Da mesma maneira um novo modelo de avaliação de qualidade de vídeo pode ser criado através da implementação da interface *VAQ* e pode ser utilizado num *NetworkScenario* concreto que esteja implementado, sendo que para isso basta configura-lo na classe *Monitor*.

Estas generalizações simples são de extrema importância pois permitem a implementação de um cenário de monitorização relativamente complexo sem grandes problemas de integração. Nas seções que se seguem serão apresentados outros conceitos que permitem uma maior flexibilidade na manipulação de pacotes de transporte associados aos protocolos de rede utilizados e também relativamente à proveniência dos vídeos monitorizados.

3.4 Cenários de Monitorização

Como foi referido no início do capítulo 2, o vídeo 3D pode ser codificado de várias formas e para cada uma delas é representado e armazenado de forma diferente. Surge então a dificuldade em separar a análise dos pacotes dos cabeçalhos para cada um dos formatos de codificação. Por outro lado os cabeçalhos dos pacotes, que terão de ser analisados para calcular uma estimativa de

qualidade, também mudam de acordo com o tipo de rede e aplicação utilizadas na transmissão do vídeo.

Um cenário de monitorização é um conceito que está associado à necessidade de análise de um conjunto de cabeçalhos dos pacotes para uma determinada rede e/ou formato de codificação. Do ponto de vista de integração no desenho do *software*, um cenário de rede tem apenas de implementar a classe *NetworkScenario* para ser utilizado. Um *NetworkScenario* engloba toda a análise aos cabeçalhos dos pacotes, extração de parâmetros necessários para o cálculo da estimativa e partilha desses valores com o monitor.

Podemos então dizer que o conceito cenário de monitorização está associado à interligação de três conceitos: um modelo NR que utiliza um determinado conjunto de parâmetros para calcular uma estimativa para a qualidade do vídeo; uma pilha de protocolos que são utilizados para transportar o vídeo 3D e cujos cabeçalhos serão analisados; um formato de representação do vídeo 3D, já que para diferentes formatos será necessário adotar abordagens diferentes na fase de extração de parâmetros.

Nas próximas duas subsecções são apresentados os dois novos cenários de monitorização desenvolvidos. À medida que se descreve a sua implementação, serão também descritas outras funcionalidades que surgiram da necessidade de resolver os desafios que foram surgindo com a implementação da mesma. O desenvolvimento de novas funcionalidades segue o pressuposto inicial de seguir uma base genérica para possa ser utilizado por outros cenários e aumentar a aplicabilidade do protótipo.

3.4.1 Vídeo Multi-View-Coding em redes IP

O primeiro cenário criado e introduzido no protótipo utiliza o modelo teórico NR apresentado na secção 2.1 mas desta vez considerando inovadoramente a transmissão do vídeo utilizando TS sobre IP. Ou seja, este estima a qualidade do vídeo codificado no formato MPEG2-TS utilizado no cenário de monitorização *MVC_BroadcastSystems* transmitido sobre uma rede IP. A

transmissão do vídeo TS é transmitida sobre a pilha de protocolos RTP/UDP/IP tal como é apresentado em [14]. Um dos pressupostos da sua implementação é a transmissão de um conjunto de pacotes TS ordenados sequencialmente no *payload* de um pacote RTP.

Quando transmitido numa rede IP, o vídeo em formato TS não corresponde a uma sequência de pacotes TS, já que este tem de ser encapsulado em todos os pacotes associados à pilha protocolar utilizados para a sua transmissão sobre IP. Portanto estará encapsulado num pacote IP que transporta um pacote UDP, que por sua vez transporta um pacote RTP que contém cerca de 7 pacotes TS. Surge a necessidade de lidar com um ficheiro que contém o vídeo contido em todos os pacotes referidos. Para isso é introduzido na estrutura funcional do protótipo um objeto que materializa o conceito dos ficheiros em formato PCAP.

A introdução deste conceito potencia o aumento da aplicabilidade do monitor já que tem sido cada vez mais populares as redes IP para fornecimento de serviços de TV (IPTV) ou outros serviços de vídeo. Outra vantagem indireta ao utilizar este formato de representação de pacotes de rede é a possibilidade de utilizar os ficheiros resultantes da saída do simulador de rede G.1050 descrito na seção 2.5. Foi criado o objeto PCAP para lidar com os parâmetros e extração de pacotes encapsulados de acordo de acordo com a sua estrutura interna.

Na figura 16 é apresentada a estrutura interna de um ficheiro PCAP [19]. Este contém um cabeçalho global constituído por um conjunto de parâmetros que indicam como devem ser lidos e interpretados os pacotes subsequentes. A seguir ao cabeçalho global vem um cabeçalho com informação dos instantes de captura e tamanho do pacote de dados que lhe está associado. Um pacote de dados contém todos os pacotes associados a uma pilha protocolar, por exemplo desde a camada lógica até à *slice* de vídeo.

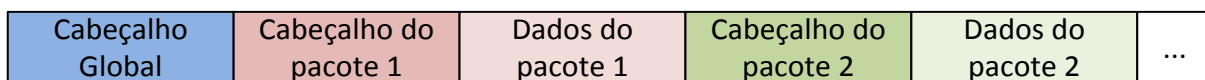


Figura 16 – Estrutura interna de um ficheiro PCAP.

Foi criada a classe *MVC_OverRTP* que permita estimar a qualidade do vídeo TS em redes IP com base no vídeo TS contido num ficheiro em formato PCAP. Como o modelo VQA pressupõe o

mesmo modelo NR teórico utilizado em *MVC_BroadcastSystems*, este não utiliza parâmetros provenientes de cabeçalhos de pacotes existentes neste cenário de monitorização, por exemplo como o IP e o RTP. Este modelo utiliza apenas os pacotes TS e como tal, o algoritmo do método *process()* é igual ao *MVC_BroadcastSystems*, já que a diferença entre ambas está unicamente na proveniência dos pacotes. Este facto fundamenta a introdução do conceito estrutura do vídeo.

Este conceito foi materializado na classe parametrizada *VideoStructure<T>*. Esta nova classe passa a ser utilizada para gerir os ficheiros que são utilizados para extrair o vídeo que se pretende monitorizar. O parâmetro *T* representado no nome da classe, corresponde a uma unidade de dados que representa o protocolo mais acima de onde é necessário extrair os parâmetros dos cabeçalhos, por exemplo, no caso do modelo NR que se está a considerar seria a classe *TS*. Este permite gerir ficheiros vídeo ou ficheiros em formato PCAP sendo que para isso basta configurar no momento da inicialização do objeto *VideoStructure* concreto. Esta configuração é feita no método *init()* do *NetworkScenario* concreto.

Para além dos métodos de configuração de proveniência dos dados (vídeo ou PCAP) do objeto *VideoStructure*, o método mais importante deste é o *getVideoUnit()*. Este método devolve a cada invocação, exatamente um pacote parametrizado por *T* (por exemplo *TS*) e que será utilizado no cenário de monitorização para extração de parâmetros e pacotes encapsulados (por exemplo, no caso do TS os pacotes *PES* e *NAL*). Esta funcionalidade potencia a utilização e aplicabilidade do protótipo no estudo e introdução de novos cenários de monitorização e modelos VQA, pois permite reutilizar código reduzindo o tempo de implementação.

Adaptando a classe *MVC_BroadcastSystems* a este novo modelo de acesso ao vídeo TS, é possível reutilizar o seu algoritmo de monitorização. O processo de monitorização da nova classe *MVC_OverRTP* é então igual ao utilizado pelo *MVC_BroadcastSystems*. A criação do novo cenário de monitorização resume-se à configuração do *VideoStructure* e manipulação do objeto *MVC_BroadcastSystems* no seu método *process()*. Na figura 17 é apresentado o diagrama que explica a relação entre ambas as classes.

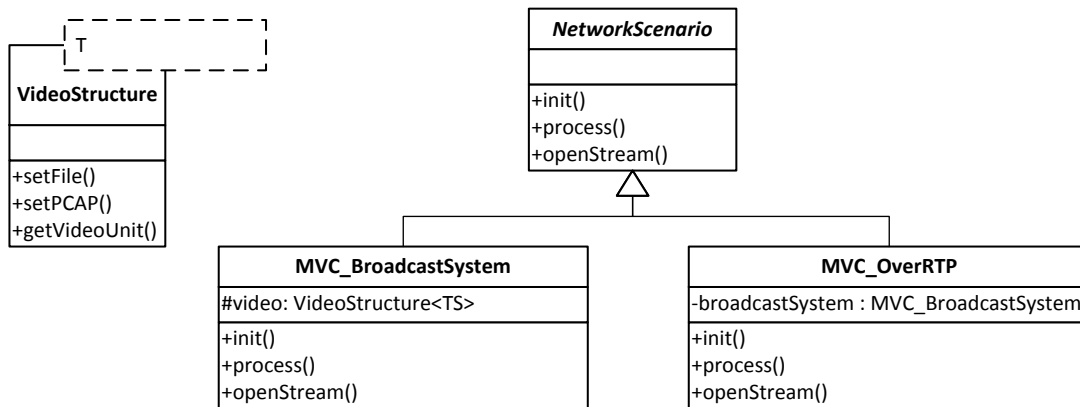


Figura 17 – Relação entre a classe abstrata *NetworkScenario* e as classes *MVC_BroadcastSystem* e *MVC_OverRTP*.

Na figura 18 é apresentado o fluxo de execução utilizado pelos objetos *MVC_BroadcastSystem* e *MVC_OverRTP* para configurar o tipo de ficheiro de onde serão extraídos os pacotes de vídeo TS utilizando o objeto *VideoStructure<TS>*. Ambos são realizados no método *init()*. No caso do objeto *MVC_OverRTP* essa configuração é feita via um objeto *MVC_BroadcastSystem* já que um objeto deste tipo é utilizado para efetuar a monitorização da qualidade do vídeo, este é representado na figura 18 com o nome atribuído ao fluxo do método *init()* do *MVC_OverRTP*.

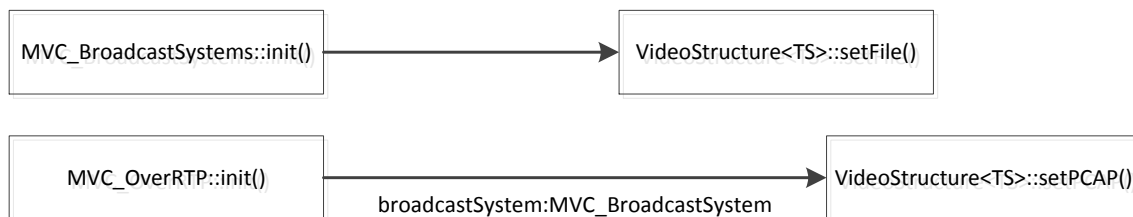


Figura 18 – Fluxo de execução para configurar a proveniência dos vídeos em ambos os cenários.

3.4.2 Video-Plus-Depth em redes IP

A maior complexidade de implementação centra-se na implementação do cenário de monitorização de vídeo no formato V+P transmitido sobre redes IP. Este introduz novos desafios já que é necessária a extração de informação de cabeçalhos RTP e também ao nível da Slice contida nos pacotes NAL.

Como o RTP é o protocolo de camada superior de onde é necessário extrair informação para o cálculo da estimativa de qualidade, a estrutura do vídeo será representada pelo objeto *VideoStructure<RTP>*. Este é utilizado no cenário de monitorização com a designação *VPlusD_OverRTP*. Pela leitura do nome da classe é possível verificar o formato de codificação do e o protocolo de transmissão do vídeo 3D.

O seu método *init()* da classe *VPlusD_OverRTP*, especificado pela classe abstrata que este implementa, está configurado para a manipulação de ficheiros do tipo PCAP, ou seja, a proveniência do vídeo transmitido nos pacotes RTP será um ficheiro neste formato. Este será também utilizado para fazer a monitorização em tempo real como será apresentado na próxima seção.

O desenvolvimento deste cenário de monitorização teve como base um ficheiro de entrada de dados que corresponde ao vídeo codificado no formato H.264/AVC com o *software* de referência *JM Reference Software*, em que o parâmetro *OutputFile* é configurado para indicar que o vídeo codificado vem encapsulado num pacote RTP. O vídeo nesse formato foi depois encapsulado num ficheiro PCAP com auxílio de um programa criado especificamente para o efeito e que se baseia em código do programa *rtpdump* do mesmo *software* de referência.

A análise e extração de parâmetros deste modelo é bem mais desafiante, face aos cenários desenvolvidos anteriormente. Para começar utiliza um maior número de parâmetros de entrada que é necessário analisar e extrair dos cabeçalhos dos pacotes, numa janela temporal de 10 segundos. Por outro lado, é necessário ter em consideração as perdas de propagação. Tendo em conta que as várias imagens codificadas são transmitidas numa ordem diferente daquela que é a

ordem pela qual são lidas por um leitor de vídeo, sendo necessário ordená-las e preservar toda a informação.

O vídeo H.264 é encapsulado num pacote RTP de acordo com o RFC 6184 - *RTP Payload Format for H.264 Video* [20]. Como já referido na seção 2.2, um pacote RTP transporta apenas uma NAL. Este tipo de encapsulamento do vídeo H.264 é classificada como encapsulamento de Modo 0. Isto significa que ao perder-se um pacote RTP está a perder-se apenas uma componente NAL do vídeo. Como tal esta pode ser detetada pela descontinuidade do número de sequência de um pacote RTP. Cada componente de uma imagem codificada pode ser do tipo I, P e B e sua identificação é feita através da análise de cada *Slice* que uma NAL contém.

Para lidar com a complexidade da extração de informação dos cabeçalhos RTP/NAL/Slice foi criada o objeto *GopStructure*. Este corresponde a uma estrutura de dados que representa a estrutura GOP do vídeo, tanto o tamanho como a sua estrutura. A especificação da estrutura e tamanho GOP é feita através da interface gráfica e é configurada inicialmente pelo utilizador.

No início do cenário de monitorização, esta estrutura é configurada e criada e á medida que o cenário de monitorização vai extraindo pacotes, estes são passados para esta estrutura de dados que os organiza de acordo com o valor do *timestamp* do cabeçalho RTP. O fluxo de informação do cenário de monitorização com o *GopStructure* é feito com base no *timestamp*. Depois de cada GOP inserido nesta estrutura de dados, o cenário de monitorização com base no *timestamp* consegue extrair o número e tamanho total de pacotes NAL de cada tipo que foi recebido e perdido. Uma imagem tem sempre o mesmo *timestamp* e é constituída por um determinado número fixo de NALs, no caso do modelo de avaliação que se está a implementar este é constituída por 8 NALs.

Quando é obtido o primeiro pacote RTP, é calculado o *timestamp* do primeiro pacote RTP do GOP a que este pertence. Este é obtida a partir da informação extraída ao nível do *slice*. É com base nesta informação que se detetam os pacotes de um GOP perdidos anteriormente. A classe *GopStructure* pode ser configurada para qualquer vídeo que se baseie na transmissão sobre RTP.

Após receber todos os pacotes pertencentes a uma janela temporal, e antes de obter os pacotes do próximo, é iniciada a extração de informação dos cabeçalhos com base em pedidos à estrutura de dados. Depois de obter toda a informação necessária para o cálculo dos PLPs, são calculados os valores para a qualidade objetiva e subjetiva.

Na figura 19 é apresentado o diagrama de classes com a relação entre o cenário de monitorização criado e os seus principais métodos e atributos.

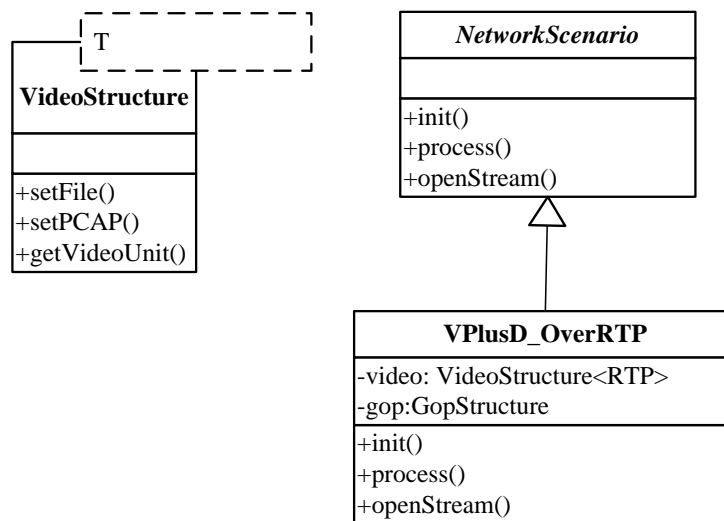


Figura 19 – Principais métodos e atributos do cenário de monitorização *VPlusD_OverRTP*.

Na fase de testes realizada após a integração do módulo que permite fazer monitorização em tempo real e que será apresentado na próxima seção, foi identificado um problema relacionado com o tamanho do *payload* que estava a ser considerado nos pacotes RTP utilizados na criação do modelo teórico. Verificou-se que existiam algumas NALs de com um *payload* duas a três vezes maior ao limite suportado por um pacote protocolo RTP (cerca de 1500 bytes). Como estes pacotes RTP foram gerados ao codificar o vídeo com o *software* de referência *JM Reference Software*, este pormenor passou despercebido.

Considerar um encapsulamento do vídeo deste tipo nos pacotes RTP permite estudar a qualidade do vídeo num ambiente de laboratório mas inviabiliza a sua utilização num cenário real. Para solucionar esta questão considerou-se a possibilidade de utilizar fragmentação das NALs de vídeo no pacote RTP. Neste caso as NAL que excedam o limite máximo do *payload* de um pacote RTP serão fragmentados e transmitidos noutros pacotes RTP.

Para respeitar as condições e características dos vídeos utilizados na construção do modelo teórico, assumiu-se o pressuposto de que um fragmento NAL corresponde à perda da NAL completa. Este pressuposto pode causar algumas diferenças nos resultados já que se está a assumir uma perda de informação maior face aquela a que é perdida na realidade.

O formato de encapsulamento do vídeo H.264/AVC que suporta fragmentação de pacotes NAL é designado por encapsulamento de Modo 1. Os fragmentos necessários para transportar uma NAL podem ter um tamanho variável e são designados como *Fragmentation Unit* (FU). Para acomodar este novo conceito foi criado o objeto *FU* que permite aceder à informação do seu cabeçalho.

Foi necessário adaptar a estrutura de dados *GopStructure* para passar a suportar este novo conceito. Este cenário de monitorização passa a suportar vídeos com dois dos três modos de encapsulamento de vídeo H.264/AVC que é possível fazer nos pacotes RTP.

Como já foi referido, para cada janela deslizante é obtido um valor para a qualidade objetiva e subjetiva. Esta é feita pela aplicação dos PLPs calculados no cenário de monitorização nas equações 6 a 8. Estes modelos teóricos foram materializados num objeto do tipo VQA com a designação *VQA_VPLUSD_OverRTP*.

Na figura 20 é apresentado o diagrama de classe com a relação existente entre o cenário de monitorização *VPlusD_OverRTP* e o modelo VQA que este utiliza para avaliação da qualidade de vídeo 3D, *VQA_VPlusD_OverRTP*. São ainda apresentados os principais métodos e atributos utilizados um diagrama de classes com a relação entre o cenário de monitorização e este modelo VQA.

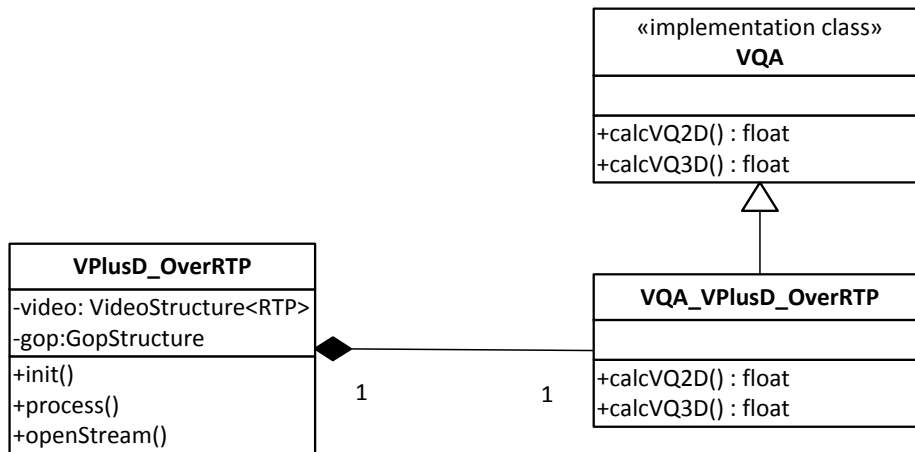


Figura 20 – Relação entre o cenário VPlusD_OverRTP e o respetivo modelo

Na tabela 4 e 5 são apresentados os coeficientes utilizados na implementação do modelo descrito na seção 2.4. Estes foram calculados para vários valores de H ou nós internos e o modelo proposto utiliza sete PLPs. Na implementação deste cenário de monitorização são utilizados também 7 PLPs mas é possível configurar menos. Dada esta diversidade e tendo como objetivo que este cenário de monitorização se possa adaptar facilmente a outras realidades, estes parâmetros são lidos a partir de um ficheiro de configuração denominado *param.conf*. A partir deste ficheiro é possível especificar os valores H e número de PLPs que se pretende utilizar. Estes parâmetros são detetados automaticamente na inicialização do VQA. Na figura 21 é apresentada a estrutura do ficheiro de configuração de leitura dos parâmetros do modelo VQA apresentado na seção 2.4. Os parâmetros N e H são lidos no início do ficheiro. Os restantes repetem-se para cada nó interno.

No final da execução do processo de monitorização, e à semelhança do que acontece com o modelo implementado na seção anterior, são armazenados num ficheiro em formato CSV os valores da qualidade objetiva e subjetiva e também o número e tamanho em bytes de cada tipo de imagem codificada que é recebida e perdida.

Nº PLPs		N							
Nº Nós internos		H							
Para cada nó interno H	bout	boutH,1	bout H,3	bout H,6	bout H,7				
	wout	wout H,1	wout H,3	woutH,6	wout H,7				
	bin	bin H,1	bin H,3	bin H,6	bin H,7				
	1 PLP	PLRH,1	0	0	0	0	0	0	0
	3 PLPs	0	PLRI H,3	PLRP H,3	PLRB H,3	0	0	0	0
	6 PLPs	0	PLRI H,6	PLRP H,6	PLRB H,6	ILRI H,6	ILRP H,6	ILRB H,6	0
	7 PLPs	0	PLRI H,7	PLRP H,7	PLRB H,7	ILRI H,7	ILRP H,7	ILB H,7	FDR H

Figura 21 – Estrutura do ficheiro de configuração para leitura de parâmetros do modelo VQA apresentado na seção 2.4.

O relatório é gerado com recurso à classe *Report*. Esta teve de ser adaptada para permitir construir um relatório adaptado aos dados gerados para o cálculo da estimativa de qualidade. A tabela 6 apresentada a estrutura do relatório gerado neste cenário de monitorização. Para cada janela deslizante novas entradas são adicionadas sequencialmente.

Tabela 6 – Estrutura e exemplo de relatório gerado pelo Módulo *Report*.

Nº da Janela deslizante	Slices previstos		Número de Slices Perdidos		Tamanho perdido (bytes)		Qualidade por janela deslizante			Qualidade Global	
	Tipo	Total	Total	Perdidos devido a erros de propagação	Total	Perdida	2D	3D	3D	3D	3D
							DMOS	DMOS	SSIM	DMOS	SSIM
1	I	80	79	10	1575	1400	22,12	73,89	0,7773	73,89	0,7773
1	P	640	494	80	47217	36590	22,12	73,89	0,7773	73,89	0,7773
1	B	1680	1114	210	72104	48614	22,12	73,89	0,7773	73,89	0,7773

Este relatório é bastante útil pois permite estudar a qualidade obtida. Para permitir a geração sequencial de vários relatórios, sem que haja sobreposição de dados, o objeto *Report* foi adaptado para atribuir a cada relatório o nome do vídeo monitorizado.

Os valores relativos à perda de pacotes e qualidade subjetiva instantânea e global são depois enviadas para o processo de interface utilizador, que atualiza os gráficos. São apresentados ao

utilizador valores de qualidade subjetiva. Devido ao facto de ambos modelos VQA implementados expressarem a qualidade subjetiva do vídeo de formas distintas MOS (*Mean Opinion Score*) e DMOS é necessário adaptar os gráficos de acordo com o novo modelo. A interface deteta o modelo que está a ser utilizado e vai seleccionando a representação correspondente, MOS ou DMOS.

Ao nível da interface gráfica foi ainda alterada a disposição de alguns elementos para que estes sejam apresentados sem cortes. Na figura 22 é apresentada a interface utilizador após estas alterações e após monitorizar com o modelo apresentado nesta seção. Na figura 22 é possível verificar que a qualidade subjetiva é representada em termos de DMOS. É possível verificar na figura o cenário de monitorização selecionado: *VPlusD_OverRTP*; e o nome do ficheiro PCAP utilizado.

Devido a questões relacionadas com a gestão do tempo não foi possível normalizar os módulos associados ao *Report* e à Interface Utilizador, seguindo um modelo de generalização utilizado para a criação de novos cenários de monitorização e modelos VQA.

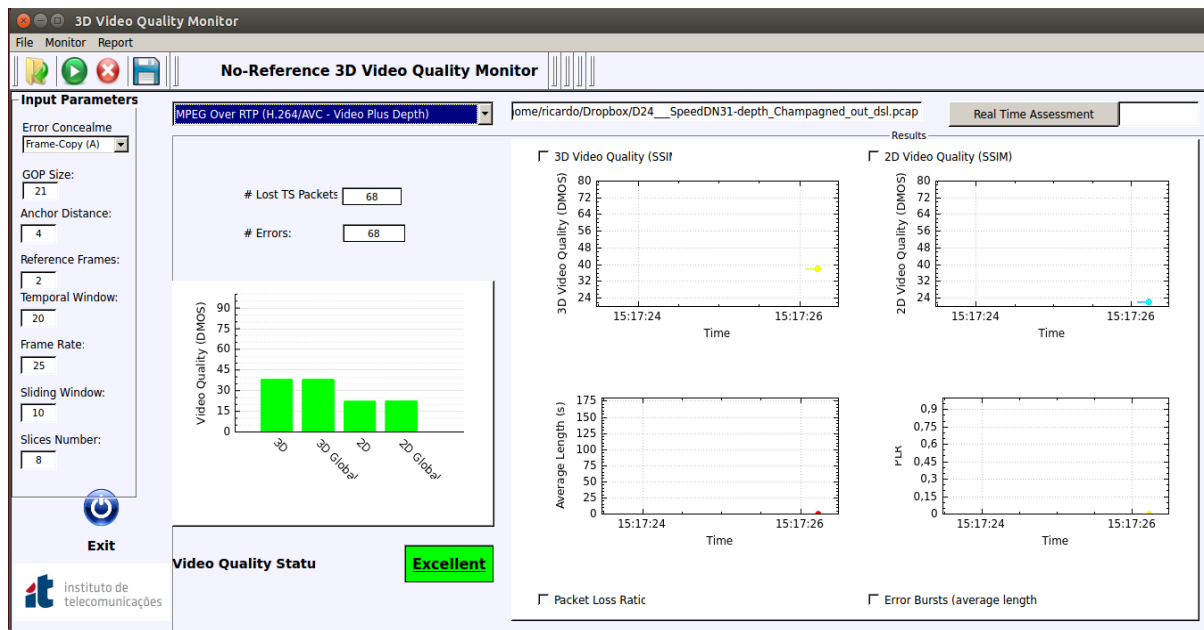


Figura 22 – Interface gráfica do Monitor de qualidade de experiência.

3.5 Monitorização em tempo real e outras funcionalidades

O desenvolvimento apresentado até esta seção permite estudar a qualidade de vídeo de uma forma estática e isolada, ou seja num ambiente de laboratório. Esta versão do protótipo pode ser utilizada para vários fins, nomeadamente o estudo um determinado aspeto da qualidade de vídeo que se pretenda aprofundar, conhecer os limites e aplicabilidade dos modelos implementados e até estudar como estes podem ser melhorados.

No entanto a possibilidade de se conseguir monitorizar o vídeo em tempo real aumentaria consideravelmente a aplicabilidade do protótipo. Este poderia servir de inspiração para futuros fornecedores de serviço que pretendem assegurar os níveis de SLA acordados com os clientes através do dimensionamento dos recursos das suas redes de acordo com os parâmetros de qualidade que pretendem oferecer. O protótipo do monitor poderia ainda ser utilizado para despoletar alterações das rotas do tráfego de vídeo de acordo com a qualidade de cada canal. Do ponto de vista académico, este pode ser utilizado para desenvolver métodos de gestão de tráfego ou outros estudos em que seja necessária fazer uma leitura da qualidade do vídeo num determinado ponto da rede.

Para responder a este desafio desenhou-se e acrescentou-se ao protótipo um módulo que permite a configuração de uma interface de rede na qual se pretende filtrar os pacotes com o vídeo que será inspecionado pelos cenários de monitorização em tempo real. Para esse efeito foi criada a classe *NetworkSniffer* com base na biblioteca *libpcap* [21].

A classe *NetworkSniffer* oferece um conjunto de funcionalidades que permite configurar uma interface de rede, compilar e aplicar filtros para selecionar apenas o tráfego de vídeo pretendido. Devido à necessidade de se estar permanentemente à escuta da interface de rede e para não bloquear os processos já existentes, foi criado um novo processo.

Para que sejam mantidos os pressupostos utilizados no desenvolvimento do protótipo, é necessário ter em atenção que a introdução deste módulo representa a introdução de uma nova

fonte de dados. Ou seja, o formato de entrada de dados do vídeo passa a ser uma interface de rede.

Como mencionado anteriormente, para garantir a reutilização dos cenários de monitorização independentemente da proveniência dos dados, foi utilizado o objeto *VideoStructure<T>*. para servir este novo propósito, este objeto foi adaptado permitir a gestão dos pacotes provenientes de uma interface de rede. A sua configuração é feita à semelhança do que acontece com os formatos de vídeo e PCAP, que são configurados no momento da criação do cenário de monitorização.

O novo processo criado para filtrar os pacotes de vídeo da interface de rede de forma a não interferir com o processo de monitorização e a interface utilizador tem o nome *realTimeThread*. A função principal deste processo é ficar à escuta na interface de rede, armazenando os pacotes que vão sendo filtrados até que estes sejam solicitados por outro processo. Para tal utiliza o objeto *NetworkSniffer*. Quando são solicitados pacotes a este processo, estes são eliminados localmente passando apenas a existir no processo que os solicitou. Foi utilizado um mecanismo idêntico aos semáforos para assegurar que o acesso à lista com os pacotes capturados não é acedida simultaneamente pelos dois processos.

A interação do utilizador com o protótipo é feita a partir da interface utilizador. É esta que lança e gere o processo de monitorização *mThread*, que vai consumir os pacotes capturados pela *realTimeThread*. Como o processo de captura de pacotes fica à escuta de pacotes passivamente e como o processo de monitorização não consegue prosseguir com o processamento na ausência de pacotes, a comunicação direta entre ambas iria dificultar a comunicação entre a interface utilizador e os outros processos. Por essa razão optou-se que a comunicação entre a *mThread* e a *realTimeThread* se faça via o processo principal.

A *framework* do Qt Creator permite apenas que processo principal consiga comunicar diretamente com as sub-tarefas lançada por si. A comunicação entre as diferentes sub-tarefas teria de ser realizada com base nos recursos do sistema operativo, o que poderia comprometer a portabilidade do código do protótipo, quando executado noutros sistemas operativos. Por outro

lado, a interface utilizador está estreitamente ligada com o processo principal sendo que este deve poder agir sobre todos os processos de acordo com as instruções do utilizador.

Na figura 23 é apresentado o fluxo de execução que ocorre quando um cenário de monitorização tenta obter os pacotes filtrados na interface de rede. O fluxo de execução que ocorre quando um cenário de monitorização requisita pacotes provenientes da interface de rede: Na etapa (1) o processo de monitorização invoca o método principal do Monitor que por sua vez invoca o método principal do cenário de monitorização (2). No passo (3) o cenário de monitorização lança uma exceção para o processo de monitorização para que este lhe solicite pacotes. Este processo solicita os pacotes ao processo principal (4) que de seguida os vai requisitar ao processo de captura de pacotes em tempo real (5). Depois de obter alguns pacotes vai adicioná-los ao cenário de monitorização que os solicitou (6).

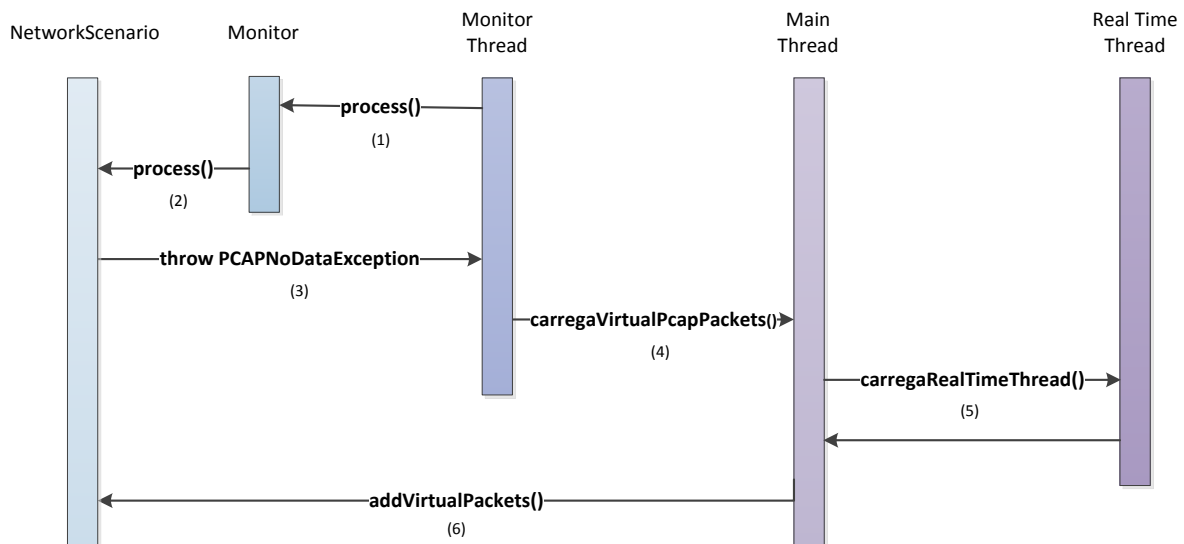


Figura 23 – Diagrama com o fluxo de execução que ocorre quando se solicitam pacotes provenientes da interface de rede.

A comunicação entre processos é realizada com recurso a objetos partilhados que são sincronizados com um mecanismo simples que é idêntico aos semáforos e com recurso a exceções. Quando o processo de monitorização não tem mais pacotes a ser processados, lança

uma exceção. Ao receber a exceção, a interface utilizador vai requisitar mais pacotes ao processo *realTimeThread*. Os pacotes são armazenados temporariamente na interface utilizador, caso existam, e ambos prosseguem com a sua execução normal. De seguida estes são adicionados ao cenário de monitorização que os solicitou para que este continue com a sua execução normal. Se continuar a não ter pacotes disponíveis, uma nova exceção é lançada e todos estes passos são repetidos.

Quando se desenvolve um cenário de monitorização que se pretende utilizar em tempo real é necessário ter em consideração que estes devem ter mecanismos de acesso a todos os pacotes processados antes da ocorrência da exceção responsável por solicitar mais pacotes à interface utilizador, caso contrário vai ocorrer perda de informação. Qualquer um dos cenários de monitorização descritos nas seções 3.4.1 e 3.4.2 podem ser utilizados com este tipo de fonte de dados. No entanto apenas o cenário de monitorização *VPlusD_OverRTP* está preparado para utilizar esta fonte de dados sem que ocorram perdas decorrentes da sua utilização. Estas perdas podem ocorrer porque são obtidos resultados de qualidade com base numa janela deslizante que normalmente pode ser constituída por um ou mais GOP. Se a meio de um GOP o cenário de monitorização ficar sem pacotes para processar a monitorização tem de ser interrompida, para requisitar mais pacotes ao outro processo. Será então necessário adicionar ao cenário de monitorização um mecanismo de recuperação de pacotes já processados naquele GOP de forma a considerar toda a informação recebida pelo processo de captura em tempo real.

No decorrer dos testes de integração do módulo de monitorização em tempo real surgiram alguns desafios. Os testes e previsibilidade do comportamento do protótipo tornou-se mais complexa já que se tornava difícil identificar onde ocorrem determinados erros. Para facilitar esta tarefa foi criado o objeto *MonitorLog*. Este objeto é um *singleton*, ou seja, apenas existe uma instância deste objeto e é composto por um conjunto de funcionalidades que permitem escrever num ficheiro de log. Esse objeto pode ser acedido por qualquer processo, o que permitiu identificar toda a sequência de execução dos mesmos assim como a interação entre eles.

O objeto *MonitorLog* regista num ficheiro de texto todos os logs enviados pelos processos. Como tal a gestão do acesso ao ficheiro log é da responsabilidade do sistema operativo resolvendo qualquer problema de acesso ao ficheiro partilhado.

Outra vantagem da introdução deste módulo no protótipo é permitir detetar erros ocorridos num cenário operacional. Para tal é necessário definir um limite máximo de bytes para o qual a informação do ficheiro não é eliminada, para que seja possível fazer uma boa gestão da utilização da memória do disco.

Para testar este módulo de captura de pacotes em tempo real foi necessário transmitir um vídeo a partir de uma interface de rede e que está configurada como fonte de informação no processo *realTimeThread*. Foi utilizado o FFmpeg [22] para fazer a transmissão do vídeo.

Até este ponto várias funcionalidades e extensões foram adicionadas aos módulos do protótipo do monitor, permitindo aumentar a sua aplicabilidade. Contudo, a sua utilização implica sempre interação com o utilizador, pois é necessário selecionar o cenário de monitorização, configurar parâmetros como o GOP, janela deslizante, se é em tempo real, etc..

Um procedimento importante após a integração de um cenário de monitorização que utiliza um novo modelo teórico é verificar o correto funcionamento do mesmo, para garantir que se comporta como o esperado. Para tal, é necessário realizar um conjunto de testes onde se aplicam um determinado padrão de erro a um conjunto de vídeos e se extraem os valores da qualidade medida pelo protótipo. Estes resultados podem depois ser estudados e comparados com os valores de qualidade medidos com recurso aos vídeos originais.

Para que seja possível extrair sequencialmente a informação de qualidade de um conjunto de vídeos foi adicionada uma nova funcionalidade que permite utilizar o protótipo a partir da linha de comando. Sem necessidade da intervenção do utilizador, é possível automatizar a utilização do monitor.

Por último, foi ainda criado o objeto *QoE_Communicator*. Este pretende ser um servidor que fica à escuta no porto 60000 de mensagens num formato bem conhecido. Este permite a comunicação

com um *broker* que age de acordo com a qualidade que é medida pelo monitor. Na troca de mensagem é especificado um limiar de qualidade para o qual o monitor informa o *broker*. Infelizmente não foi possível concluir o desenvolvimento deste módulo e testá-lo experimentalmente.

Capítulo 4

4 Testes e resultados

Neste capítulo será descrito o processo desenvolvido para o estudo da relação entre várias condições de rede face à qualidade do vídeo medido com recurso ao protótipo criado. Pretende-se também inferir se o modelo teórico de avaliação de qualidade tem confirmação com um conjunto de cenários de teste que refletem condições de rede mais realistas e com tráfego interferente real. Como já foi referido, os testes desenvolvidos assentam sobre o modelo de avaliação teórico apresentado na seção 2.4 e que a descrição da sua implementação no protótipo é feita na secção 3.4.2. Este processo é descrito em três partes distintas:

- Descrição do processo de obtenção dos parâmetros de qualidade medidos pelo protótipo criado: definição dos cenários de teste que refletem o estado da rede para o qual se pretende medir a qualidade do vídeo, execução de simulações utilizando a ferramenta apresentada na seção 2.5 e por fim a obtenção do output gerado pelo protótipo e os vídeo obtidos no recetor.
- Descrição do processo de obtenção do SSIM real para os vídeos com erros tendo em conta todas as considerações e pressupostos adotados no decorrer da implementação do protótipo.
- Apresentação dos resultados obtidos e que relacionam os parâmetros de qualidade de experiência com algumas as condições de rede que se fizeram variar nos cenários de teste.

4.1 Definição dos cenários de teste

Na seção 2.5 foi apresentado um simulador de rede que permite efetuar a avaliação de vídeo na medida em que é possível introduzir um padrão de erros num vídeo. É possível obter o impacto

gerado pela transmissão do mesmo, desde o núcleo da rede a um terminal. Pretende-se assim determinar uma relação entre os parâmetros de rede utilizados num conjunto de simulações de rede com a qualidade dos vídeo introduzidos nessas simulações. Pretende-se ainda inferir se os resultados teóricos têm confirmação com os resultados práticos obtidos com recurso a cenários de teste mais realistas face aos realizados na definição do modelo teórico, que utiliza um modelo probabilístico que está a começar a afastar-se dos comportamentos das redes atuais [26]. Isso é conseguido através da utilizar-se o simulador G.1050 nas simulações de rede para simular diferentes condições de rede.

Utilizando o *ffmpeg* codificou-se a componente de profundidade do vídeo *champanhe.yuv* com os parâmetros de codificação especificados na seção 2.2. Foram realizados testes com este modelo de avaliação devido ao facto de este ter sido originalmente criado para vídeo transmitido sobre redes IP e pelo facto de este modelo ainda não ter sido testado com um pressuposto operacional.

A componente de profundidade do vídeo foi codificado e transmitida para uma interface de rede local, utilizando o *ffmpeg*. Esse vídeo foi depois capturado utilizando o *Wireshark* e todos os seus pacotes são armazenados num ficheiro em formato PCAP com a designação *depth_champagned*. Foi utilizado para o encapsulamento das NAL em RTP com o modo de encapsulamento 1, para considerar um cenário mais próximo do operacional apesar de poder chegar a resultados mais afastados dos obtidos num ambiente laboratorial. Como referido anteriormente algumas das imagens contidas vídeos experimentais contêm NALs com um tamanho superior ao tamanho máximo para um pacote RTP.

Por outro lado sem a fragmentação de pacotes NAL (utilizando o modo 0) não é possível transmiti-lo para uma interface de rede e como tal não é possível obter uma captura real para esse vídeo. Na fase de desenvolvimento do módulo de monitorização em tempo real (seção 4.4) foram realizados um conjunto de testes com o objetivo de tentar reduzir o tamanho das NAL que excedem o tamanho máximo através da diminuição gradual da qualidade dos vídeos. Verificou-se que para os limites e características de codificação do vídeo e transmissão, não foi possível utiliza uma NAL por pacote RTP.

Na tabela C2 e C3 do anexo C são apresentados vários cenários de teste definidos no simulador G.1050 e de onde se derivaram os cenários de teste utilizados nos testes desta dissertação. Para referenciar as condições de rede globais e tráfego interferente utilizados nos cenários de teste, adotou-se a nomenclatura Dp3 e Du7 que será explicada de seguida. Estes cenários consideram um modelo de rede core-to-LAN pretendendo simular o fornecimento de um serviço de vídeo instalado no núcleo da rede e que é fornecido aos consumidores finais, por exemplo o IPTV.

Os parâmetros definidos em cada um dos casos de teste apresentados nas tabelas C1 e C2 suportam duas tecnologias de acesso, DSL e GPON. Para os cenários de teste definidos neste trabalho resultam da adaptação dos parâmetros de rede da tecnologia DSL, e por isso é utilizada a letra ‘D’ da nomenclatura considerada (por exemplo Dp3). Na tabela C2 são apresentados os cenários definidos para tráfego parcialmente gerido (partially managed). Para identificar este tipo de gestão do tráfego os cenários de teste deste tipo vêm identificados com ‘p’, Dp3. O mesmo acontece para os cenários Du7, onde u vem da designação *unmanaged* (não gerido). O número inteiro utilizado na nomenclatura dos cenários referem-se ao nível de severidade refletidos nas condições de rede e onde 1 corresponde ao cenário menos severo e 7 corresponde ao cenário mais severo.

Foram realizados casos de teste com base nestes cenários para se verificarem as diferenças entre dois casos extremos de um domínio que reflete uma rede IP propensa a erros: de um lado temos um serviço que reflete algumas preocupações com o tráfego de vídeo e em condições de severidade médias; e por outro temos uma rede onde não é feita qualquer diferenciação do tráfego e caracterizado por condições mais adversas. Foram feitos previamente várias simulações com condições de tráfego mais favoráveis, como os cenários de rede geridos, mas os erros introduzidos não eram suficientemente significativos para serem considerados nestes testes.

Como o modelo de avaliação de qualidade de vídeo baseia-se apenas na componente de profundidade do vídeo 3D. Tendo em conta que o vídeo 2D e a informação de profundidade são transmitidos por canais separados podendo muitas vezes seguir rotas distintas [17], nos casos de teste serão apenas simulados os vídeos com a componente de profundidade do vídeo 3D.

Com base nos cenários de teste Dp3 e Du7 foram criados um conjunto de casos de teste mas tendo em conta as seguintes diferenças: no cenário Dp3 o fluxo ‘Iptv sd stream 1 - VBR (qty)’ foi substituído pelo fluxo depth_champagned.pcap mantendo-se a mesma configuração do tráfego; e no cenário Du7 foi adicionado o fluxo depth_champagned.pcap com as mesmas características de tráfego utilizadas no cenário Dp3. Apesar da componente de profundidade do vídeo 3D estar codificado em HD optou-se pela substituição por um fluxo IPTV SD porque a informação de profundidade tem um tamanho relativamente menor que a componente 2D e também pelo facto de a codificação H.264 permitir taxas de transmissão relativamente baixas.

Na tabela 7 é feita uma caracterização do tráfego para cada um dos cenários. O parâmetro S (*start*) corresponde ao momento em que a transmissão é iniciada relativamente ao início da simulação. O parâmetro P corresponde à prioridade atribuída ao tráfego de acordo com o referido na seção 2.5. O parâmetro BW corresponde à largura de banda atribuída na interface de rede para o fluxo. E R significa o número de vezes que esse tráfego é repetido. R com o valor -1 significa que o tráfego é repetido durante toda a simulação [26].

Tabela 7 – Tabela com a caracterização do tráfego utilizado nos cenários de rede utilizados nas simulações.

Casos de teste	Dp3				Du7			
	BW	S	R	P	BW	S	R	P
Caracterização dos fluxos	BW	S	R	P	BW	S	R	P
Depth_Champagned	0.361	2	1	2	0.361	2	1	7
IPTV SD Stream 1-CBR (qty)	0.260	1	-1	2	-	-	-	-
Peer-to-peer Rate Down/ Up	0.5/1.35	0	-1	7	0.5/1.35	0	-1	7
HTTP Rate Down / Up	1/0.5	0	-1	7	1/0.5	0	-1	7
OTT2 Rate Down / Up	1/1	0	-1	7	-/1	0	-1	7
VoIP/FoIP Rate Down / Up	1/1	1	-1	1	1/1	0	-1	7

Para concluir a definição dos casos de teste é ainda necessário definir os parâmetros de rede. De entre os parâmetros de rede que é possível variar estão a velocidade (SPEED), a taxa de erros (BER) e o tipo da tecnologia de acesso. O atraso (Delay) e o número de switch do núcleo de rede também podem ser configurados. Após um conjunto de testes preliminares chegou-se a conclusão que o número de switch do núcleo não alteravam significativamente os resultados da qualidade de e como tal este parâmetro não foi considerado neste estudo.

Como já foi referido, relativamente á tecnologia de acesso, o simulador disponibiliza as tecnologias DSL e GPON. Os limites típicos para estas tecnologias são apresentados na tabela C1 do anexo C. Decidiu-se adotar para estes testes uma tecnologia de acesso com características idênticas à das redes *wireless*, para testar a possibilidade de monitorizar-se vídeo em femto-células nas casas dos utilizadores [30]. Para aproximar as características da LAN a uma rede *wireless* foram utilizados um conjunto de valores mais adequados para a velocidade e taxa de erros da tecnologia de acesso e para os atrasos da rede, tipicamente apresentados para estas redes. Os valores de atraso do núcleo da rede e taxa de erros e velocidade da tecnologia de acesso é apresentada na tabela 8. São apresentados os limites mínimos e máximos para cada parâmetro, o deslocamento entre dois valores consecutivos e o número total de valores resultantes.

Tabela 8 – Parâmetros de qualidade de rede que se fizeram varias nos cenários de teste.

Variable Parameters	Valores mínimo, máximo e distância entre dois pontos			
	Min	Máx	Δ	Nº de Valores
Delay (ms)	2	30	2	15
BER ($\times 10^{-2}$)	1×10^{-5}	1.278	4.2577×10^{-1}	31

Velocidade (Mbytes/s)					
SPEED	2	11	36	48	54

Definiu-se para cada caracterização de tráfego Dp3 e Du7 um conjunto de casos de teste onde para cada valor de velocidade se fez variar os valores de BER mantendo um valor fixo do atraso de rede. Todos os outros parâmetros utilizados na definição dos casos de teste são iguais aos apresentados nas figuras C2 e C3 do anexo C. Foram definidos no total cerca de 4650 casos de teste para estes cenários.

4.2 SSIM estimado e medido

Depois de definidos os casos de testes, executaram-se as simulações e no final obtivemos os ficheiros PCAP do vídeo `depht_champaged` resultante de cada simulação. No final das simulações, os vídeos com erros foram colocados numa estrutura de pastas que permitiu identificar: no primeiro nível o perfil de tráfego Du3 ou Dp7; no segundo nível o atraso da rede com o nome D_n em que n é um inteiro atribuído sequencialmente para cada valor do atraso indicada, com início no valor mínimo e até ao valor máximo; e no terceiro nível onde existem as pastas SPEED e BER. Com base nessa estrutura de diretórios atribuiu-se um nome padronizado para facilitar a automatização da sua manipulação. Esses nomes seguem a seguinte estrutura: $D[u/p][3/7]_{SPEEDx-Dn_BERm}$ onde x é o valor da velocidade utilizada e m é um número inteiro atribuído sequencialmente para cada taxa de erros, a começar da mais baixa.

O protótipo foi executado em modo linha de comando, passando-lhe cada um dos resultados da simulação e os relatórios CSV com a informação da qualidade dos vídeo obtidos (SSIM e DMOS), o número total de *slices* recebidos e perdido, o número de *slices* perdidos devido a erros de propagação e o tamanho total de bytes recebidos e perdidos para cada um dos tipos de *slices* I, P e B. Esses relatórios têm o mesmo nome que o vídeo de entrada, para manter a estrutura indicada. Os parâmetros fixos utilizados na configuração do protótipo são os indicados na seção 2.4. As simulações com o simulador G.1050 e a obtenção da qualidade de cada um dos vídeos resultantes da simulação foram realizados sequencialmente de forma automatizada.

A partir dos valores dos parâmetros utilizados nas simulações de rede e da qualidade dos vídeos medida com o protótipo é possível obter uma relação entre a qualidade dos vídeos e condições de rede consideradas. Para obter um termo de comparação para os resultados obtidos pelo protótipo, é necessário medir a qualidade objetiva, SSIM, dos vídeos resultantes da simulação.

Para medir o SSIM real é necessário utilizar os vídeos de referência e os vídeos com erros no formato *raw* (.yuv). Para obter os vídeos com erros em formato *raw* foi necessário em primeiro lugar, extrair o vídeo H.264 dos ficheiros PCAP e de seguida descodifica-los. Para obter os vídeos H.264 contidos nos ficheiros PCAP utilizou-se a ferramenta Videosnarf 0.46 [23]. Esta

ferramenta permite detetar o vídeo H.264 numa sessão RTP de modo *offline*, e de simples utilização.

Devido ao elevado número de vídeos simulados e tendo em conta o espaço ocupado em disco depois de descodificados, para a medição do SSIM real realizou-se os seguintes passos:

- Extração do vídeo H.264 existente no ficheiro PCAP resultante da simulação G.1050;
- Descodificação do vídeo H.264 utilizando o *software* de referência JM 15. Depois de realizar um conjunto de testes com versões mais recentes concluiu-se que este permitia descodificar um maior número de imagens, permitindo assim utilizar uma maior quantidade de informação para o cálculo do SSIM e minimizando a perda de qualidade que resulta da descodificação do vídeo H.264 e assim conseguir um melhor termo de comparação com os resultados medidos no monitor;
- Cálculo do SSIM do vídeo descodificado com recurso ao vídeo de referência. Foi utilizada a função `ssim_index` [24];
- Geração de um relatório com o SSIM medido e o DMOS calculado utilizando a equação 2.8.

4.3 Apresentação dos resultados

Com base nos valores de qualidade objetiva e subjetiva obtidos realizou-se um estudo da relação destes com os parâmetros de rede variáveis que foram utilizados nas simulações de rede especificados na tabela 8. Desta forma é possível verificar como evolui a qualidade de vídeo ao longo de cada um destes parâmetros.

Para se fazer essa exposição, optou-se pela representação tridimensional da qualidade em função de dois desses parâmetros. Utilizou-se o Matlab para a gerar os gráficos e interpolar os valores entre os valores de qualidade obtidos e parâmetros de QoS utilizados. Para cada um dos valores do atraso foi criado um gráfico onde é apresentada a evolução do SSIM e DMOS ao longo a taxa de erros (BER) e a velocidade (SPEED). Esta forma de representação gráfica permite perceber

facilmente a variação da qualidade sem recurso a um número demasiado elevado de gráficos. De notar que para a qualidade subjetiva DMOS os valores variam no intervalo 22 e 73 correspondendo respetivamente a uma qualidade de experiência excelente e degradada.

Para cada um dos tipos de cenários de tráfego considerados nas simulações de rede, Dp3 e Du7, obtiveram-se gráficos para os valores de qualidade estimados e medidos diretamente dos vídeos resultantes das simulações de rede. Desta forma é possível comparar dois cenários com perfis de tráfego distintos e ao mesmo tempo comparar os valores estimados e efetivamente medidos.

Nas figuras 24 e 25 são apresentados os valores obtidos para um atraso fixo de 0.18 s. Do lado esquerdo são apresentados os valores para o SSIM e DMOS estimados pelo protótipo. Do lado direito são apresentados os valores obtidos para a qualidade medida com base nos vídeos obtidos nas simulações de rede. É apresentado um cenário com um valor de atraso intermédio para se ter uma ideia global da semelhança entre os resultados de qualidade estimados e medidos e perceber também o impacto do aumento do tráfego na qualidade de vídeo.

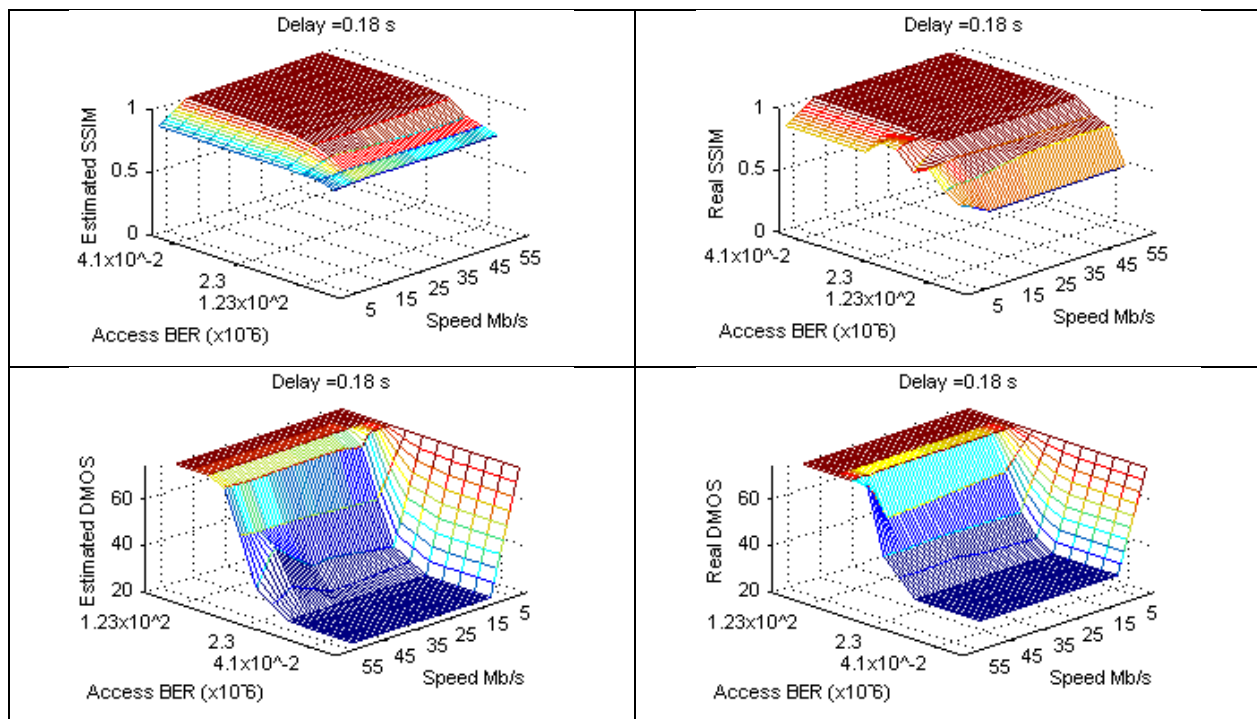


Figura 24 – SSIM e DMOS estimado e medido para um atraso fixo de 0.18s para o cenário Dp3.

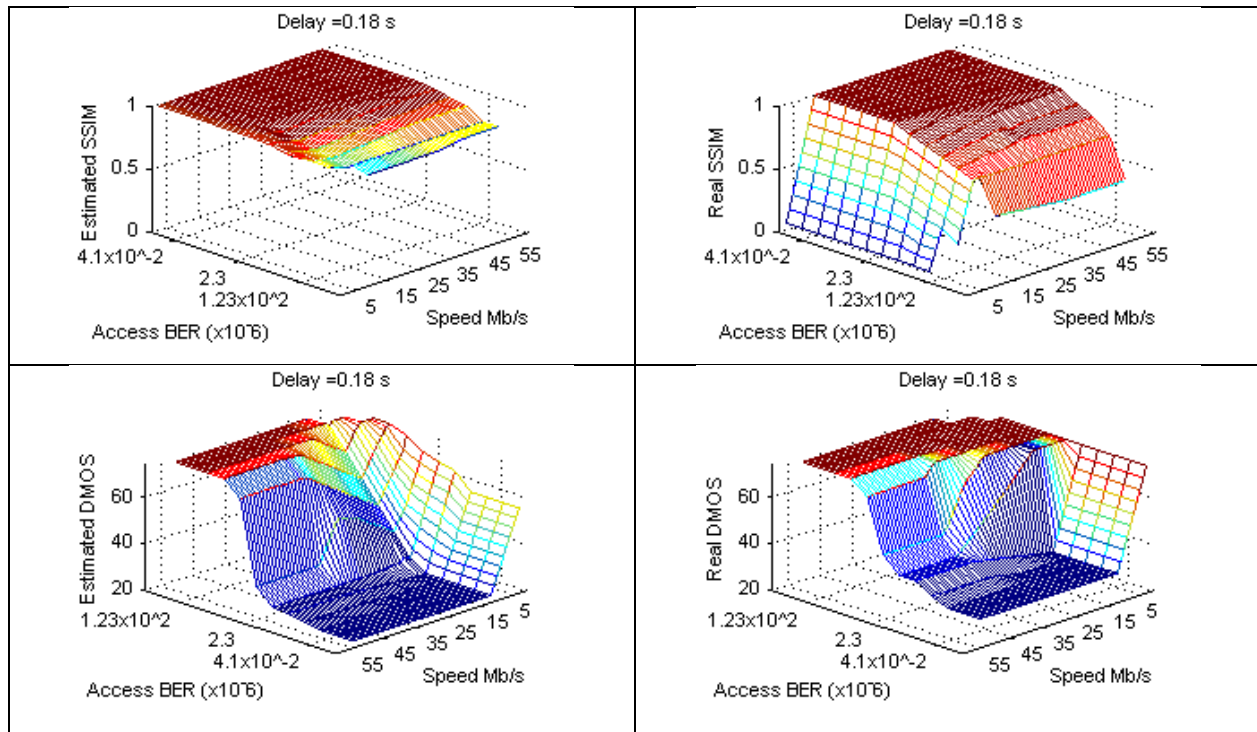


Figura 25 – SSIM e DMOS estimado e medido para um atraso fixo de 0.18s para o cenário Du7.

Comparando-se os dois cenários de rede, é possível verificar que a qualidade de vídeo no cenário Du7 decai mais rapidamente com o aumento da taxa de erros. Este facto resulta de este ser caracterizado por um maior volume de tráfego e de não ser realizada qualquer gestão de qualidade de vídeo 3D. Verifica-se ainda que para os cenários apresentados, é necessário garantir uma velocidade superior a 10 Mbytes/s para se conseguir obter uma qualidade razoável para uma taxa de erros mais baixa.

Ao comparar-se os valores estimados e medidos é possível verificar que estes são grosseiramente idênticos. As suas diferenças notam-se essencialmente nos limites mais adversos das condições de rede para a transmissão do vídeo como uma taxa de erros elevada e uma velocidade baixa. É possível verificar que os valores estimados têm um comportamento mais linear que os valores de qualidade medidos, já que contém menos alterações do padrão de qualidade ao longo dos parâmetros de rede.

Foram obtidos outros gráficos mas desta vez fazendo-se variar o atraso e a taxa de erros fixando-se a velocidade. Nas figuras 26 e 27 são apresentados esses gráficos obtidos para uma velocidade de 36 Mbytes/s. À semelhança das figuras 24 e 25, do lado esquerdo são apresentados os valores para o SSIM e DMOS estimado pelo protótipo. Do lado direito são apresentados os valores obtidos para a qualidade medida com base nos vídeos obtidos nas simulações de rede. Relativamente à variação do atraso não é possível concluir quanto ao impacto que este tem sobre a qualidade já que para a mesma taxa de erros não se identificam diferenças significativas na qualidade.

Este comportamento é previsível já que sem interatividade de tráfego ao longo da rede, o atraso afeta o tráfego de igual forma não se refletindo na qualidade experienciada. Já ao longo da taxa de erros nota-se uma acentuada degradação da qualidade. Por observação dos gráficos é possível oferecer uma qualidade de vídeo perto do excelente se for garantida uma taxa de erros menor que 2.3×10^{-6} .

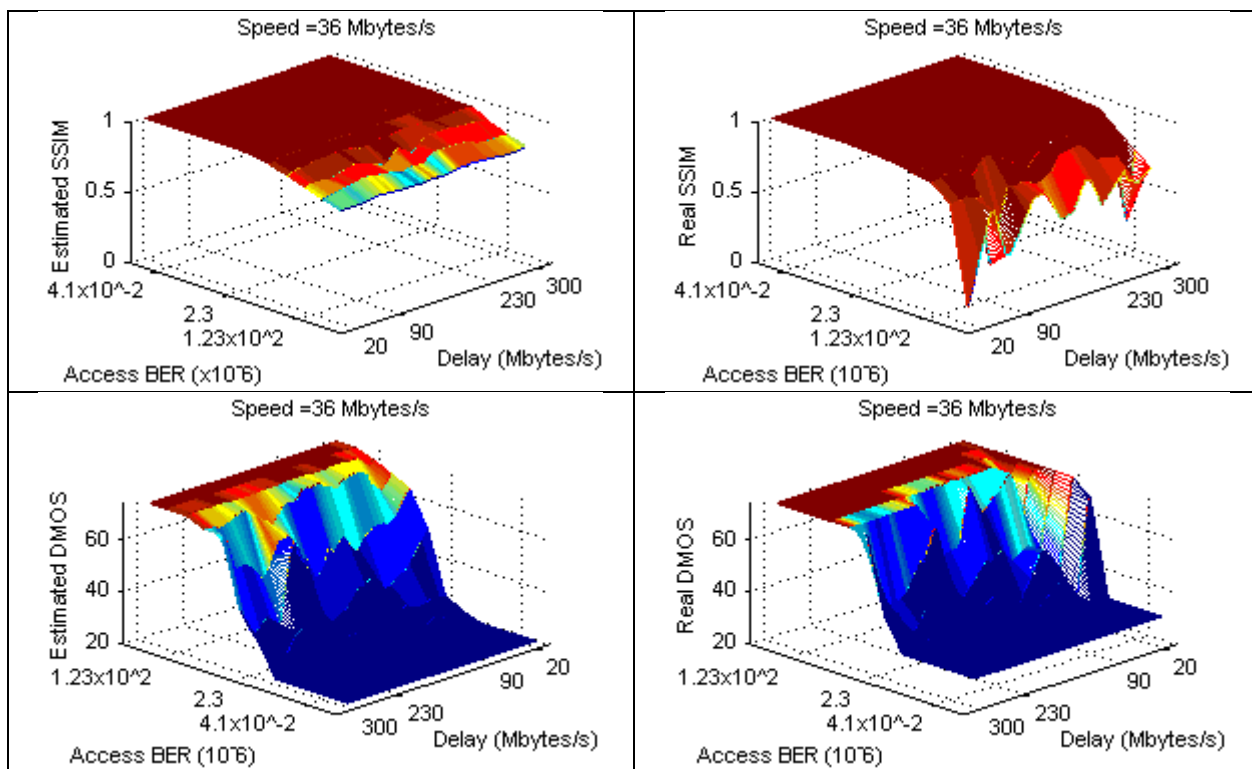


Figura 26 – SSIM e DMOS estimado e medido para uma velocidade fixa de 36 Mbytes/s para o cenário Dp3.

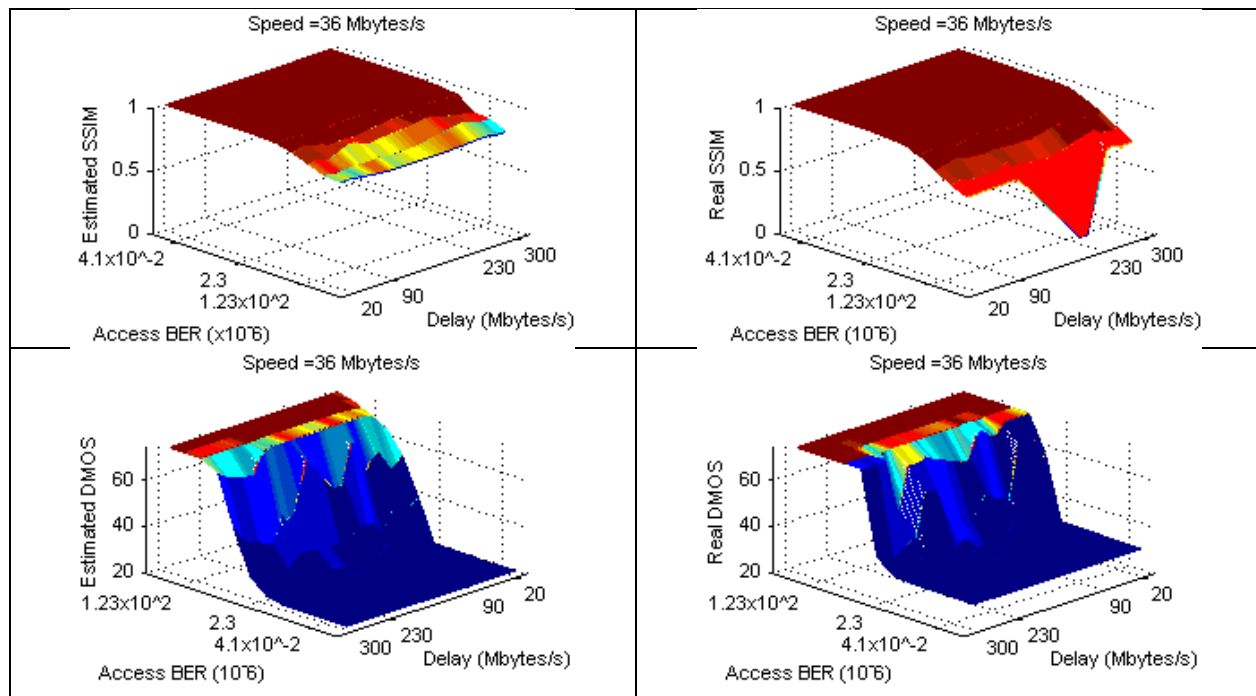


Figura 27 – SSIM e DMOS estimado e medido para uma velocidade fixa de 36 Mbytes/s para o cenário Du7.

Quando se comparam os resultados estimados e medidos é possível verificar que a qualidade estimada começa a decrescer para uma taxa de erros menor, relativamente aos mesmos valores medidos. Contudo estes vão decrescendo mais lentamente até atingirem a qualidade mais baixa. Para a qualidade medida os valores começam a decrescer mais tarde mas a queda é mais abrupta.

No anexo A são apresentados todos os gráficos com os valores de SSIM e DMOS estimados e medidos para cada um dos valores do atraso fixo para cada um dos cenários Dp3 e Du7. No anexo B são apresentados 60 gráficos idênticos mas para cada uma das velocidades de consideradas.

Ao comparar os resultados estimados pelo protótipo e os dados medidos é possível verificar que os valores da qualidade obtidos pelo protótipo são idênticos aos obtidos para os valores medidos. No entanto essa classificação é subjetiva e carece de uma análise mais detalhada para que se possa tirar tal conclusão. Na figura 28 é apresentada a relação entre ambos os resultados. Nesse gráfico são considerados todos os valores obtidos nos casos de teste com valores de SSIM superiores a 0.7. Na relação entre os valores de SSIM estimados e medidos neste intervalo foi possível obter uma correlação de *Pearson* de 0.9352, de onde se pode concluir que existe uma

forte correlação entre ambas. Para o DMOS essa correlação é igualmente forte tendo-se obtido uma correlação de 0.9108. A relação entre os valores de DMOS estimados e medidos são apresentados na figura 29

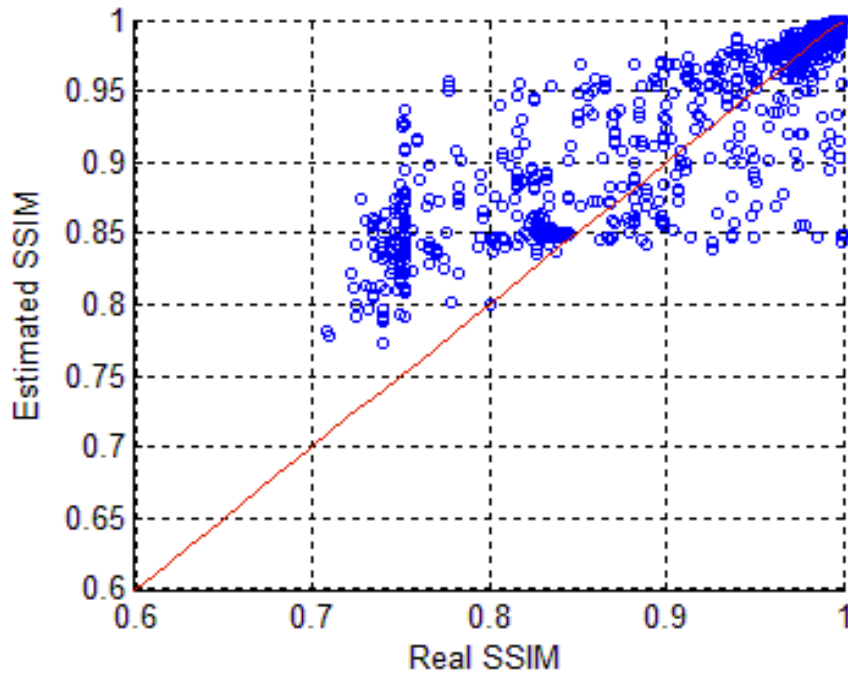


Figura 28 – Relação entre SSIM medido e estimado.

Foram considerados valores de SSIM superiores a 0.7 devido ao facto de valores mais baixos não serem relevantes do ponto de vista da qualidade de experiência já que se verifica que para um valor de SSIM de 0.9 se obtenha uma qualidade de experiência à obtida menor ou igual a 0.7. Esse fenómeno pode ser observado na figura 30 onde é apresentada a relação entre a qualidade objetiva e subjetiva. É possível verificar que para valores de SSIM superiores a 0.98 a qualidade de experiência vai degradando para um valor constante. Esta curva é bastante idêntica à apresentada em [15] o que sugere que os resultados obtidos com o protótipo refletem corretamente os dados definidos pelo modelo teórico

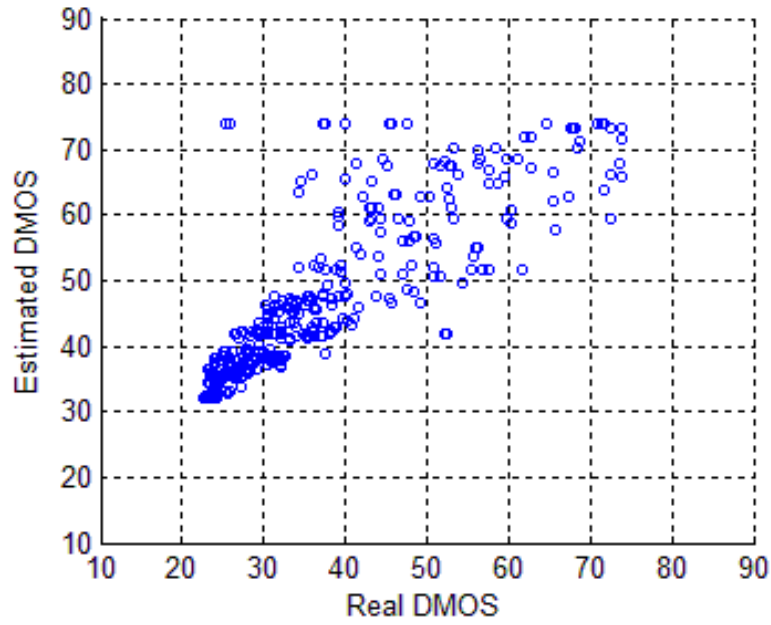


Figura 29 – Relação entre DMOS medido e estimado.

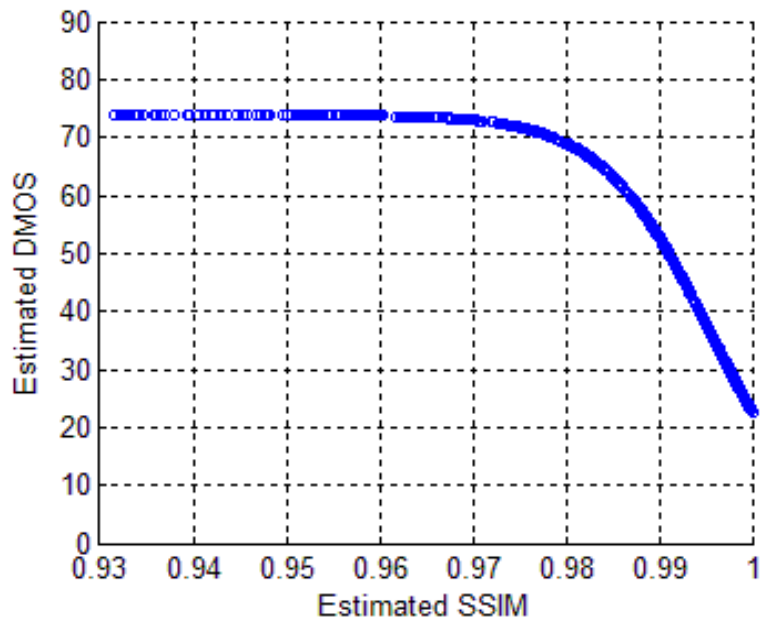


Figura 30 – Relação entre SSIM e DMOS estimado.

Capítulo 5

5 Conclusão

Os desenvolvimentos realizados com base no protótipo de um monitor de qualidade de vídeo 3D já existente permitem uma maior flexibilidade de integração de novos cenários. O monitor em si deixou de ser visto como um estimador da qualidade de vídeo para um cenário de monitorização específico, passando a permitir uma fácil integração de novos cenários de monitorização e modelos VQA.

As alterações realizadas ao nível da entrada de pacotes de vídeo do protótipo vem a potenciar a sua utilização já que permite a monitorização de vídeo transmitido sobre redes IP de forma *offline*, através da utilização de ficheiros em formato PCAP ou em tempo real através da monitorização de uma interface de rede. Este facto aliado à facilidade em integrar novos cenários e modelos VQA, tornam este protótipo uma ferramenta útil para aqueles que se dedicam ao estudo dos modelos de qualidade. Poderá ser útil também para estudar o impacto do desempenho de uma infraestrutura na qualidade de vídeo utilizando um modelo conhecido por exemplo através da monitorização da qualidade de vídeo medida em tempo real e indicada a um agente gestor da rede.

Relativamente ao modelo de avaliação de qualidade implementado de raiz foi possível verificar que os resultados obtidos pelo protótipo têm uma forte correlação com os resultados obtidos com o vídeo de referência. Estes ficam bastante próximos dos resultados apresentados para o modelo de avaliação teórico o que valida a correta implementação do mesmo. O método de obtenção de resultados baseou-se num simulador de rede de eventos discretos que reflete as características das redes atuais e é diferente do modelo probabilístico utilizado na implementação do modelo teórico.

No entanto por considerar que todos os fragmentos de uma NAL são perdidos quando pelo menos um deles se perde, é possível afirmar que este modelo teórico é mais agressivo face à qualidade real, já que se está a ignorar informação de imagens codificadas que na realidade serão utilizadas para gerar as imagens que o utilizador vai experienciar.

No futuro este protótipo pode ser utilizado no desenvolver diversas aplicações, não só pela introdução de novos modelos de avaliação de vídeo que podem ser estudados, como em aplicações de rede que necessitem de valores de qualidade de vídeo. Alguns tópicos que podem ser desenvolvidos no futuro são:

- - Adaptar o cenário de monitorização do protótipo *MVC_OverRTP* para que este possa ser utilizado para monitorização em tempo real sem que haja lugar a perdas resultantes da recolha de pacotes que são provenientes da interface gráfica.
- - Estudar a validade/domínio do modelo utilizado no cenário de monitorização *MVC_OverRTP* em redes IP, por exemplo utilizando uma abordagem idêntica à descrita neste trabalho. Este modelo foi desenvolvido para avaliação em redes DVB e derivado do encapsulamento nos pacotes RTP este pode apresentar resultados diferentes.
- Calibrar o modelo de avaliação de qualidade utilizado no cenário *VPlusD_OverRTP* de forma a considerar a informação de NALs fragmentadas que é ignorada no caso de perda de pelo menos um fragmento.
- - Uniformização dos módulos Report e Interface utilizador de acordo com as características do modelo que se está a utilizar a cada momento no protótipo. Apesar de toda a uniformização feita e das ferramentas de que dispõe o protótipo, a uniformização deste módulo permite que as novas integrações sejam mais fáceis e mantém a arquitetura do protótipo gerível a longo prazo.
- - Terminar o módulo de comunicação de qualidade de experiência que permite a notificação de um *broker* que pode atuar sobre o monitor e agir de acordo com os valores de monitorização medidos num ponto da rede. O desenvolvimento deste módulo irá permitir aumentar a aplicabilidade deste protótipo permitindo desenvolver um conjunto de outros estudos relacionados coma gestão de tráfego de vídeo ao longo de redes IP.

- – Novos modelos de avaliação podem ser desenvolvidos e posteriormente adicionados ao protótipo.

Referências

- [1] “3DVQM: 3D Video Quality Monitor: System specification”. 30 Dezembro 2011.
- [2] Shierl, Thomas; Narasimhan, Sam –“*Transport and Storage Systems for 3-D Video Using MPEG-2 Systems, RTP, and ISO File Format*”. April 2011.
- [3] Feitor, Bruno; Assunção, Pedro – “3DVQM: 3D Video Quality Monitor – Métodos sem referência – nível de pacotes de dados”. Leiria, 2012. Draft.
- [4] C. Sun, X. Liu, X. Xu, and W. Yang, “An efficient quality assessment metric for 3Dvideo” in Computer and Information Technology (CIT), 2012 IEEE 12th International Conference on, 2012, pp. 209-213.
- [5] ITU-T Recommendation J.144, “Objective perceptual video quality measurement techniques for digital cable television in the presence of a full reference”, Mar. 2004.
- [6] A. Takahashi, D. Hands, and V. Barriac, ”Standardization activities in the ITU for a QoE assessment of IPTV”, Communications Magazine, IEEE, vol. 46, no. 2, pp.78-84, 2008.
- [7] S. Olsson, M. Stroppiana, and J. Baina, “Objective methods for assessment of video quality: state of the art”, Broadcasting, IEEE Transactions on, vol. 43, no. 4, pp.487-495, dec 1997.
- [8] ITU-T Recommendation P.1201. 2012, ”Models and tools for quality assessment of streamed media - Parametric non-intrusive assessment of audiovisual media streaming quality”. Geneva, Suíça.
- [9] ITU-T Recommendation G.1070. 2012, “Multimedia Quality of Service and performance - Opinion model for video-telephony applications”. Geneva, Suíça.
- [10] K. Yamagishi, K. Ushiki, T. Hayashi, and A. Takahashi, “Parametric packet-layer model for monitoring video quality of IPTV services”, in NTT Technical Review, Apr. 2009, pp. 110-114.
- [11] S. Argyropoulos, et al, “No-reference bit stream model for video quality assessment of H.264/AVC video based on packet loss visibility”, in Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on, May 2011, pp. 1169-1172.

- [12] P. Perez, J. Macias, J. J. Ruiz, and N. Garcia, “*Efect of packet loss in video quality of experience*”, Bell Labs Technical Journal, vol. 16, no. 1, pp. 91-104, 2011.[Online]. Available: <http://dx.doi.org/10.1002/bltj.20488>
- [13] K.-H. Thung and P. Raveendran, “*A survey of image quality measures,*” in Technical Postgraduates (TECHPOS), 2009 International Conference for, 2009, pp. 1–4.
- [14] MacAulay, Alex, Felts, Boris, Fisher, Yuval, “*IP Streaming of MPEG-4: Native RTP vs MPEG-2 Transport Stream*”, in Envivio WHITEPAPER, October 2005
- [15] Feitor, Bruno –“*3DVQM: 3D Video Quality Monitor*”. Leiria : [s.n], Novembro 2013. Dissertação de Mestrado.
- [16] C.Fehn,“*A 3D-TV Approach Using Depth-Image-Based Rendering (DIBR),*” in Visualization, Imaging, and Image Processing, 396, 84–89 (2003).
- [17] Joao R. S. Soares, Luís Alberto da Silva Cruz, Pedro A. Amado Assunção and Rui Marinheiro “*No-reference lightweight estimation of 3D video objective quality*” in Image Processing (ICIP), 2014 IEEE International Conference on; Paris, Oct 2014, pp. 763-767
- [18] ITU-R, “*Recommendation ITU-R BT.500-11: Methodology for the subjective assessment of the quality of television pictures,*” (2002).
- [19] Hani's blog, “*A look at the pcap file format.*” [Online.] . Available: <http://www.kroosec.com/2012/10/a-look-at-pcap-file-format.html?view=flipcard>,
- [20] Wang ,Y.-K, Even, R, Technologies, Huawei, Kristensen, T, Tandberg, Jesup, R, Communications, WorldGate, RFC6184, May 2011 < <http://tools.ietf.org/html/rfc6184>>
- [21] Tcpdump/Libpcap Community, “*TCPDUMP & LIBPCAP.*” [Online]. Available: <http://www.tcpdump.org/>
- [22] FFmpeg project, “*FFmpeg.*” [Online]. Available: <https://www.ffmpeg.org/>
- [23] Siperá VIPER Lab, Arjun Sambamoorthy, Anil Mahale, Jason Ostrom, “*VideoSnarf Security Too.I*” [Online]. Available: <http://ucsniff.sourceforge.net/videosnarf.html>
- [24] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “*Image quality assessment: From error visibility to structural similarity*” IEEE Transactions on Image Processing, vol. 13, no. 4, pp.600-612, Apr. 2004.
- [25] M. Naccari, “*H.264/AVC bitstream transmission simulator,*” (2008).

- [26] ITU-T Recommendation G.1050. 2011, “*Multimedia Quality of Service and performance-Network model for evaluating multimediatransmission performance over Internet Protocol*”. Geneva, Suíça
- [27] Z. W. Ligang, Z. Wang, L. Lu, and A. C. Bovik, “*Video quality assessment using structural distortion measurement,*” in in Proc. IEEE Int. Conf. Image Proc, 2002, pp. 65–68
- [28] Feitor, Bruno et al. – “*No-Reference Quality Models for Single Frame Loss in 3D Video*”. Castelo Branco, Portugal. Conftele 2013. 2013
- [29] Feitor, Bruno et al. – “*Objective quality prediction model for lost frames in 3D vídeo over TS*”. Budapest, Hungary. ICC 2013. 2013.
- [30] H. Alves, R. N. Marinheiro, and J. Moura, “*Flow-Mobility for PMIPv6,*”. Évora, Conftele 2015, p. 4.

Anexo A

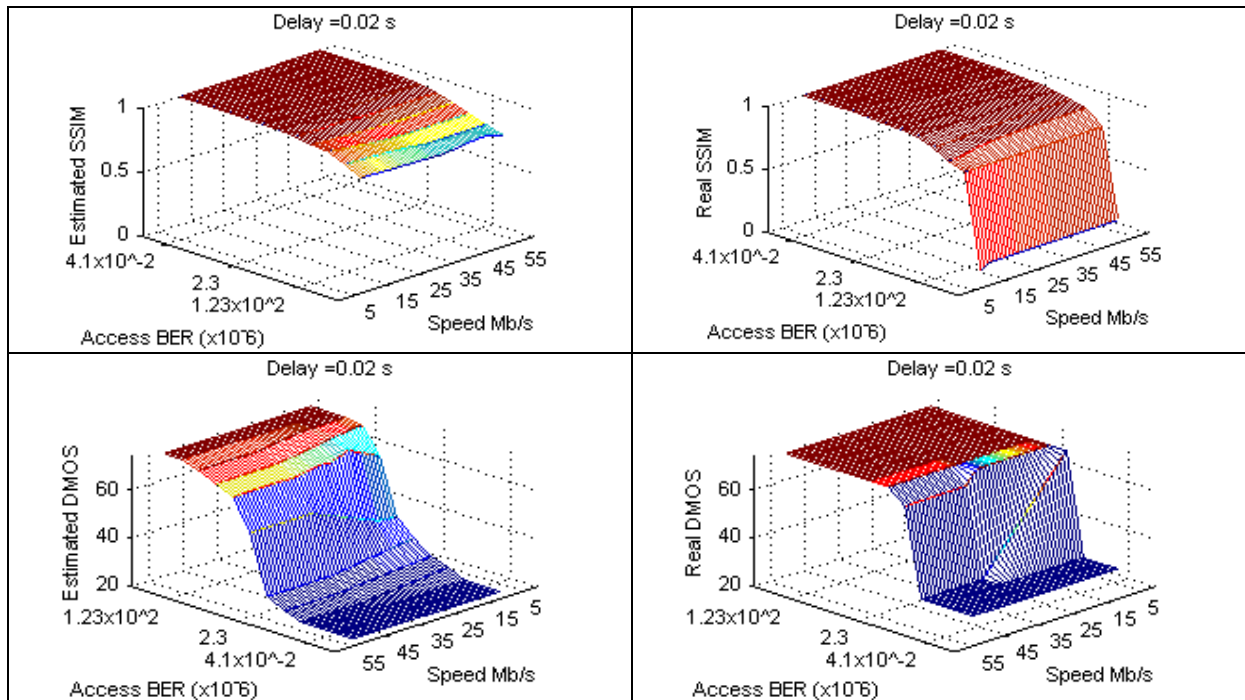


Figura A1 - SSIM e DMOS estimado e medido para um atraso fixo de 0.02s para o cenário Dp3.

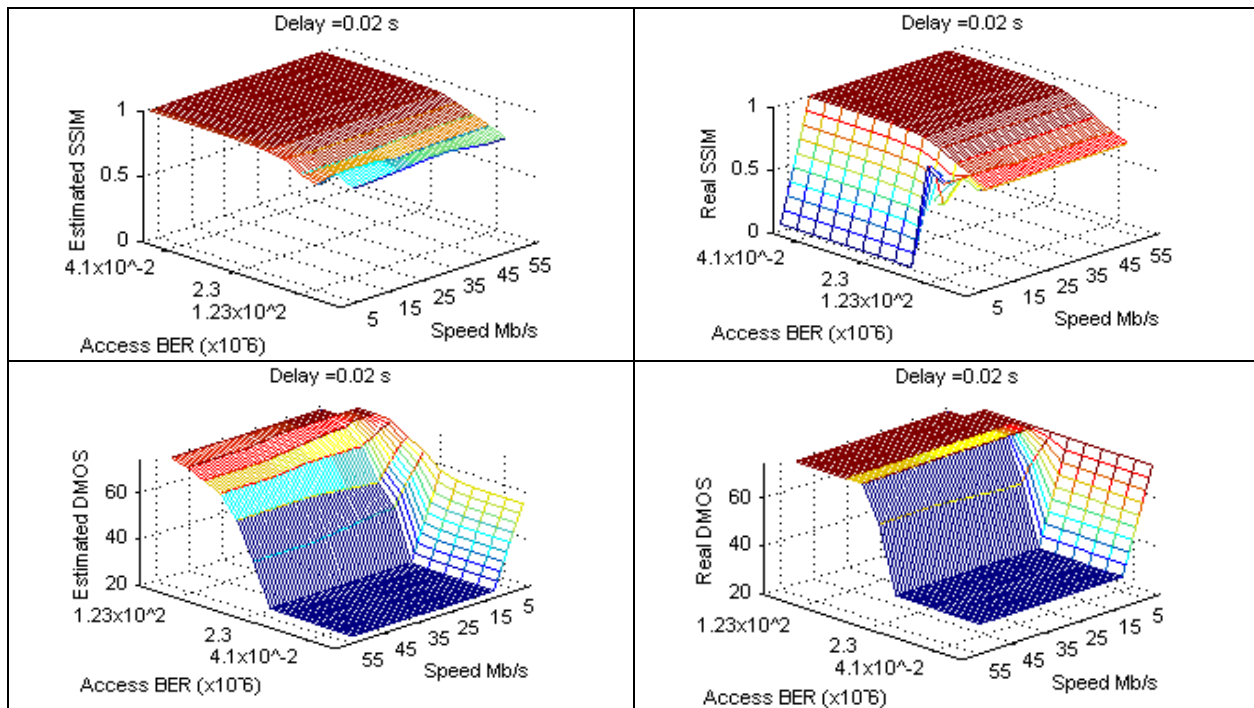


Figura A2 - SSIM e DMOS estimado e medido para um atraso fixo de 0.02s para o cenário Du7.

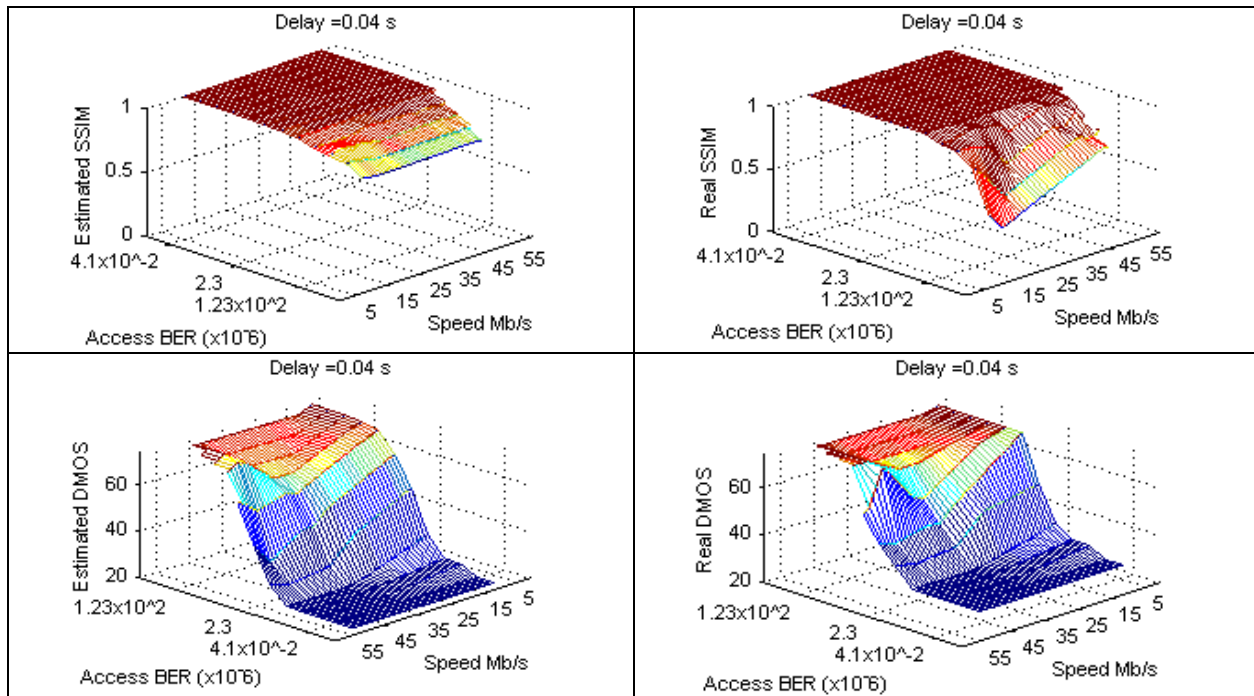


Figura A3 - SSIM e DMOS estimado e medido para um atraso fixo de 0.04s para o cenário Dp3.

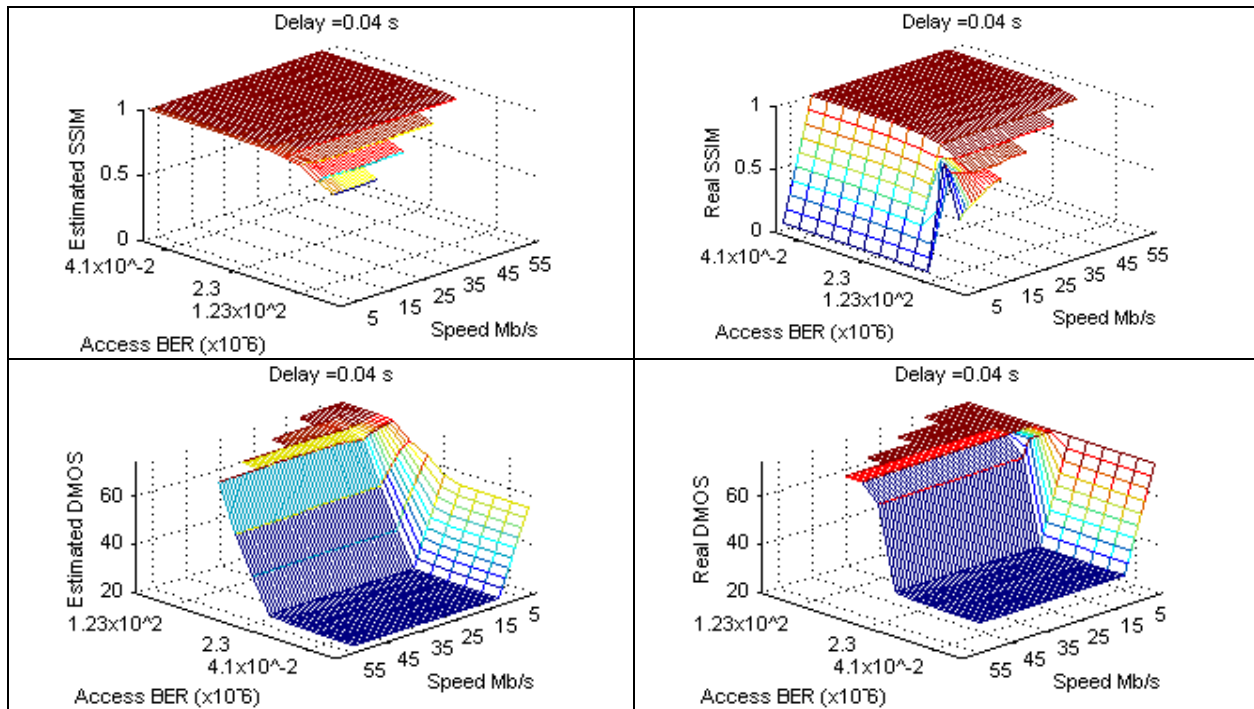


Figura A4 - SSIM e DMOS estimado e medido para um atraso fixo de 0.04s para o cenário Du7.

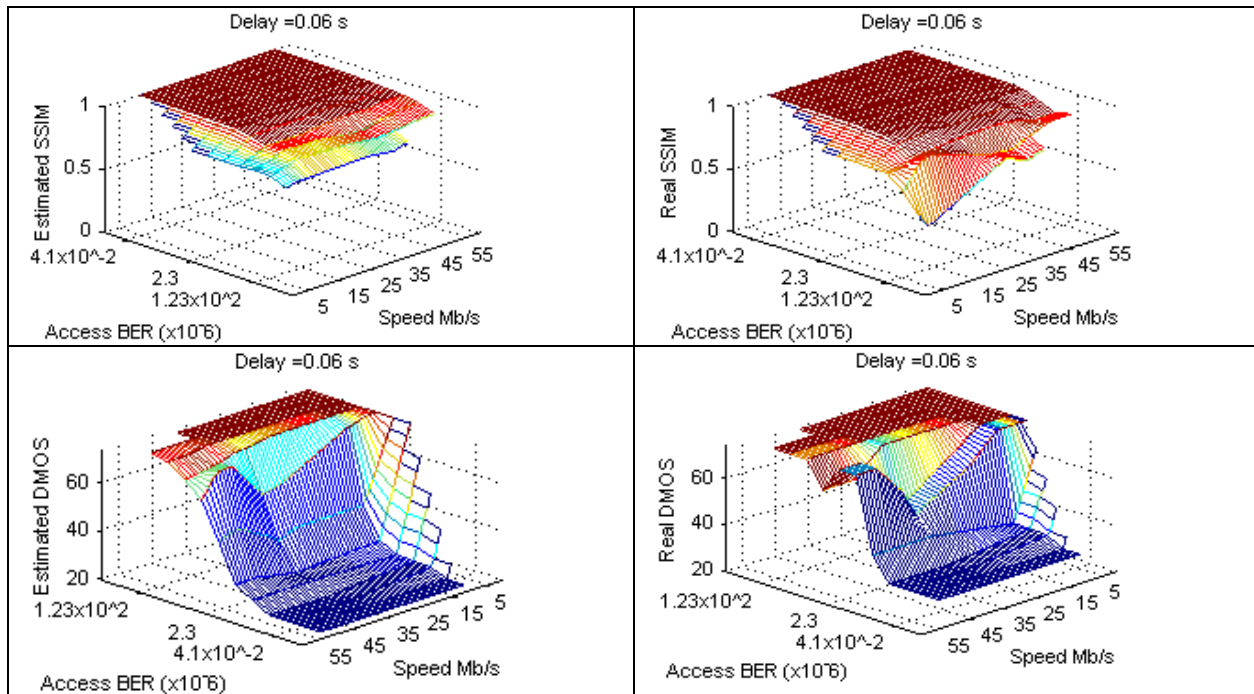


Figura A5 - SSIM e DMOS estimado e medido para um atraso fixo de 0.06s para o cenário Dp3.

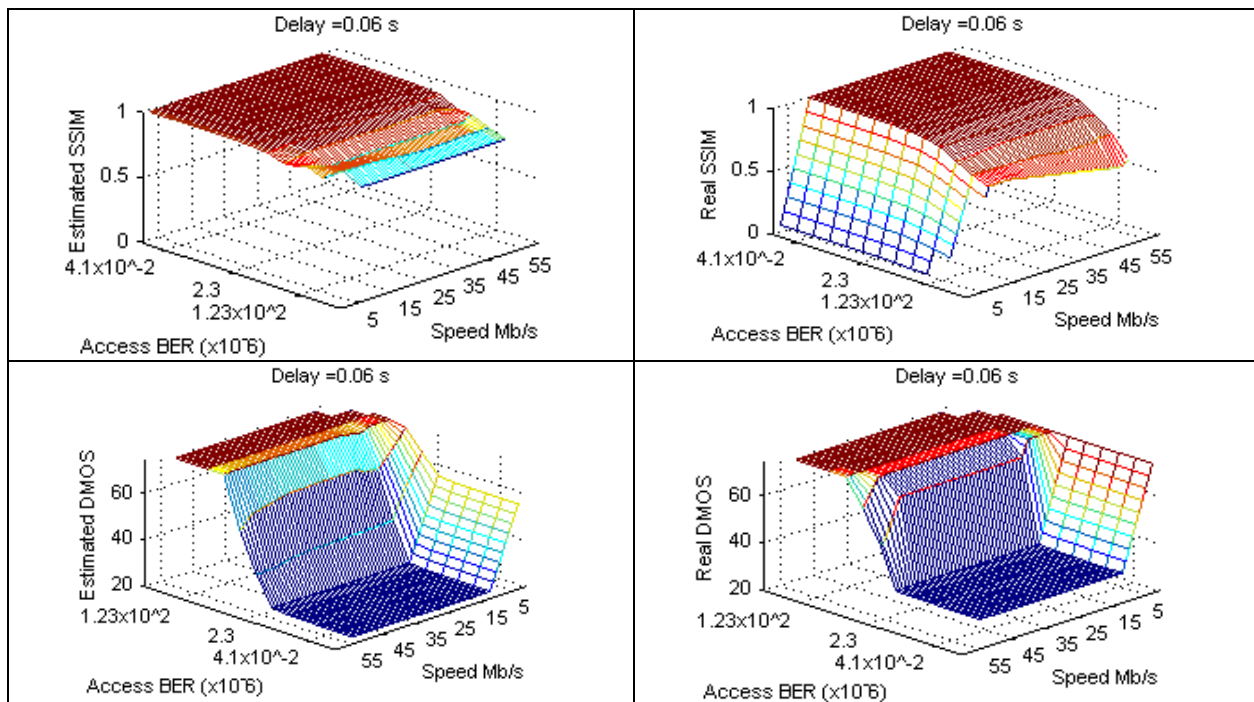


Figura A6 - SSIM e DMOS estimado e medido para um atraso fixo de 0.06s para o cenário Du7.

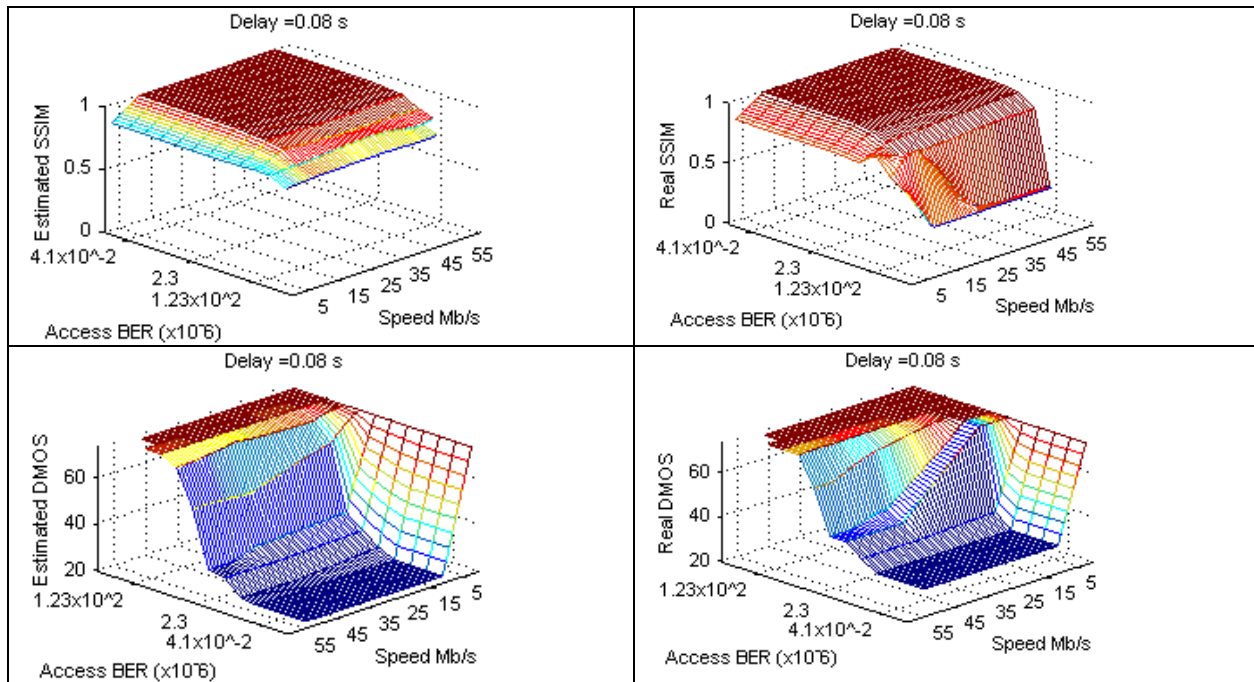


Figura A7 - SSIM e DMOS estimado e medido para um atraso fixo de 0.08s para o cenário Dp3.

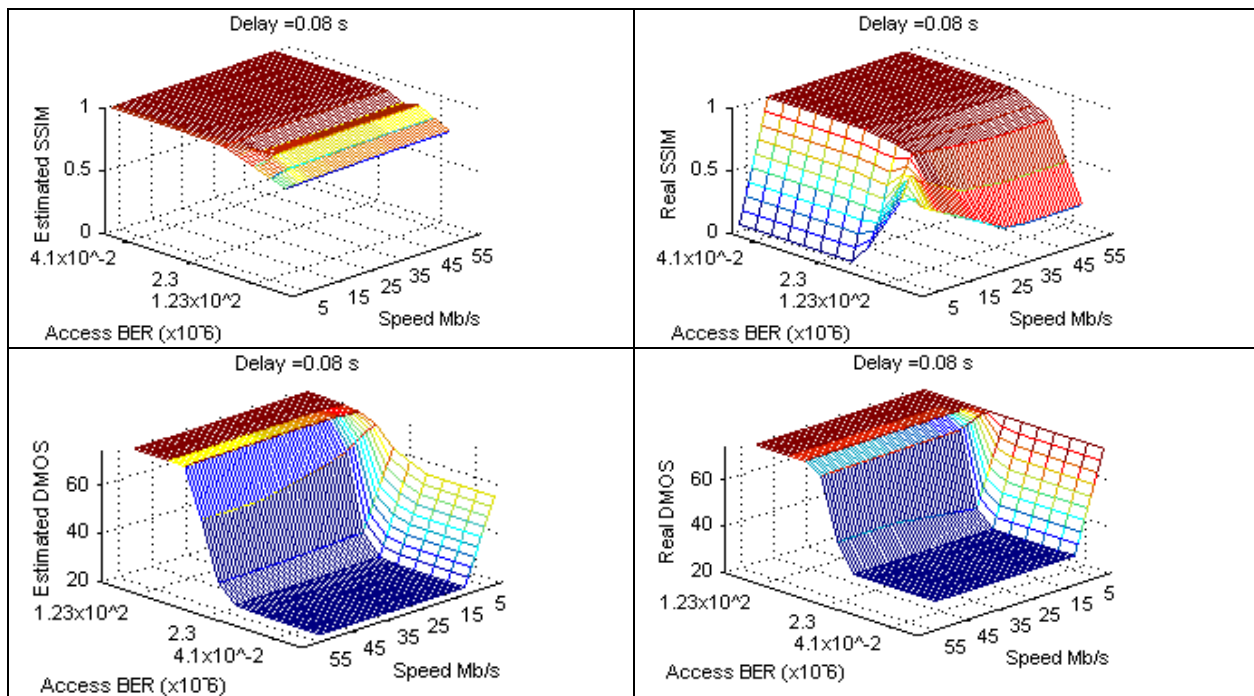


Figura A8 - SSIM e DMOS estimado e medido para um atraso fixo de 0.08s para o cenário Du7.

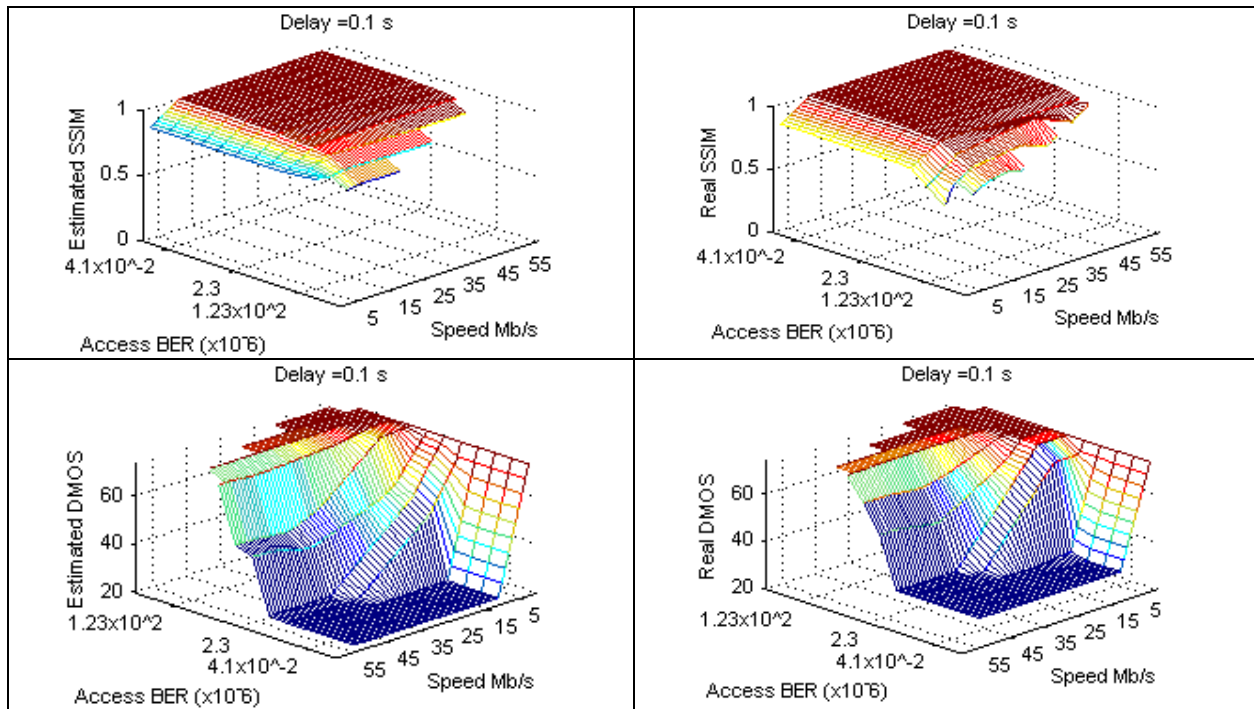


Figura A9 - SSIM e DMOS estimado e medido para um atraso fixo de 0.1s para o cenário Dp3.

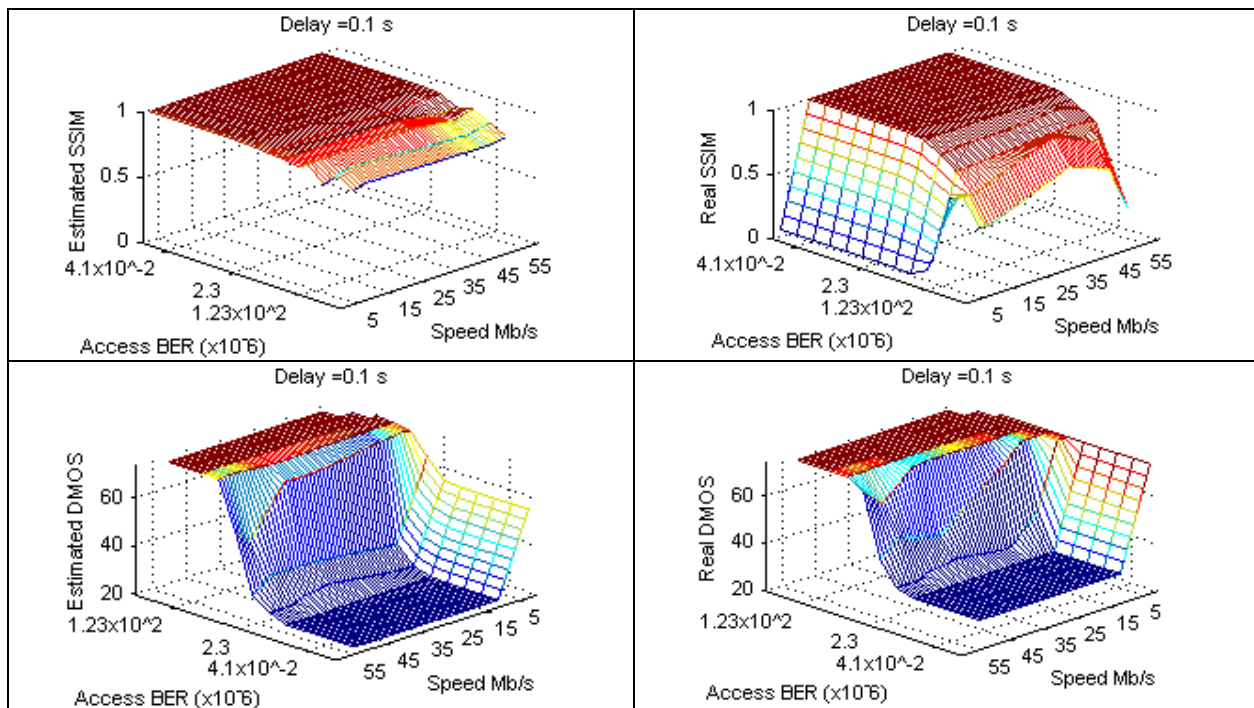


Figura A10 - SSIM e DMOS estimado e medido para um atraso fixo de 0.1s para o cenário Du7.

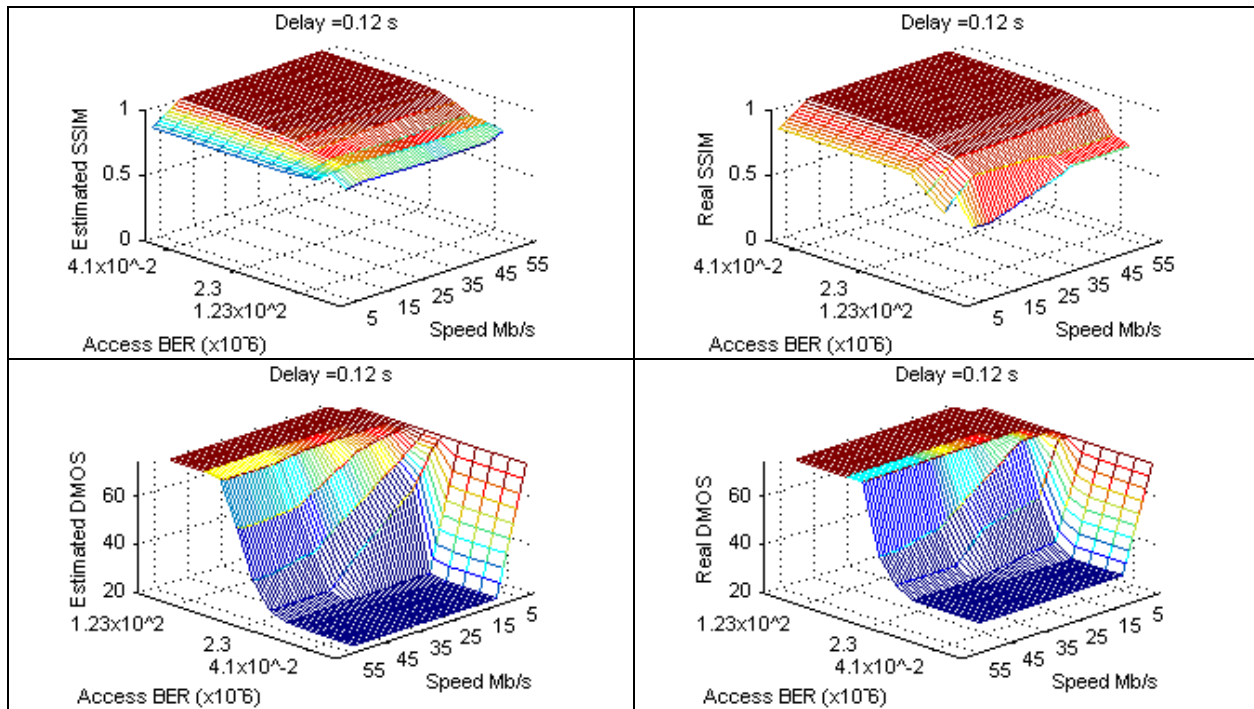


Figura A11 - SSIM e DMOS estimado e medido para um atraso fixo de 0.12s para o cenário Dp3.

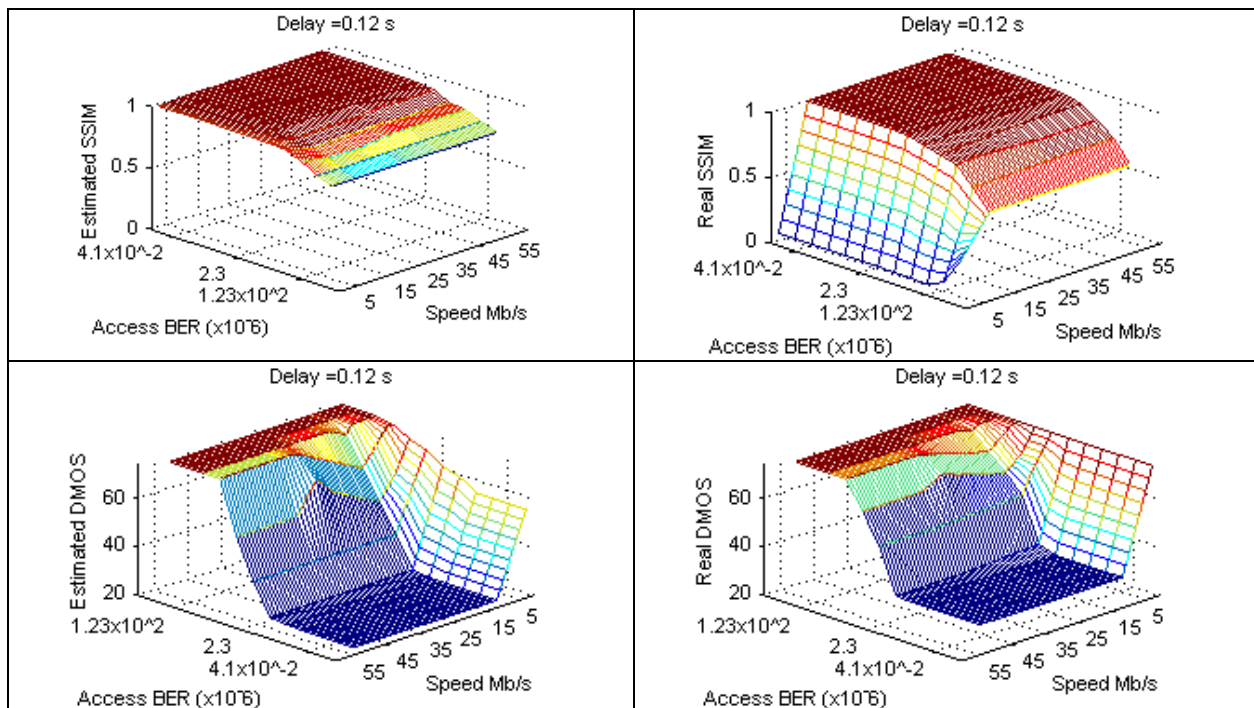


Figura A12 - SSIM e DMOS estimado e medido para um atraso fixo de 0.12s para o cenário Du7.

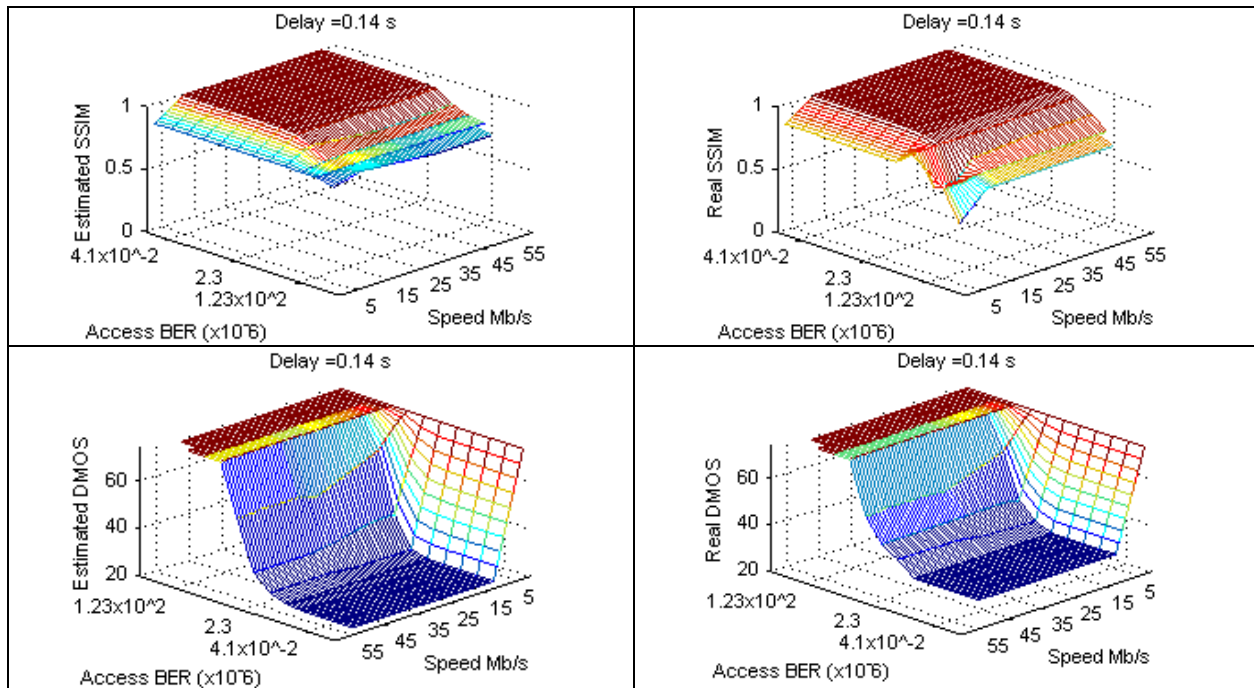


Figura A13 - SSIM e DMOS estimado e medido para um atraso fixo de 0.14s para o cenário Dp3.

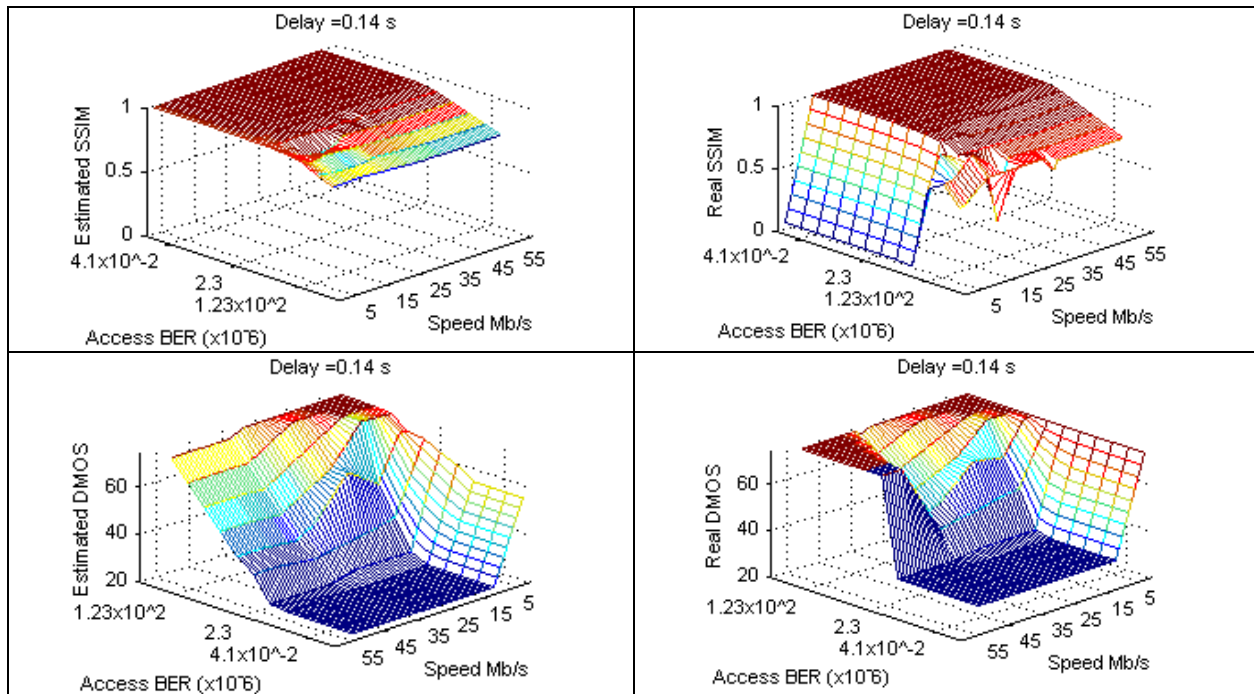


Figura A14 - SSIM e DMOS estimado e medido para um atraso fixo de 0.14s para o cenário Du7.

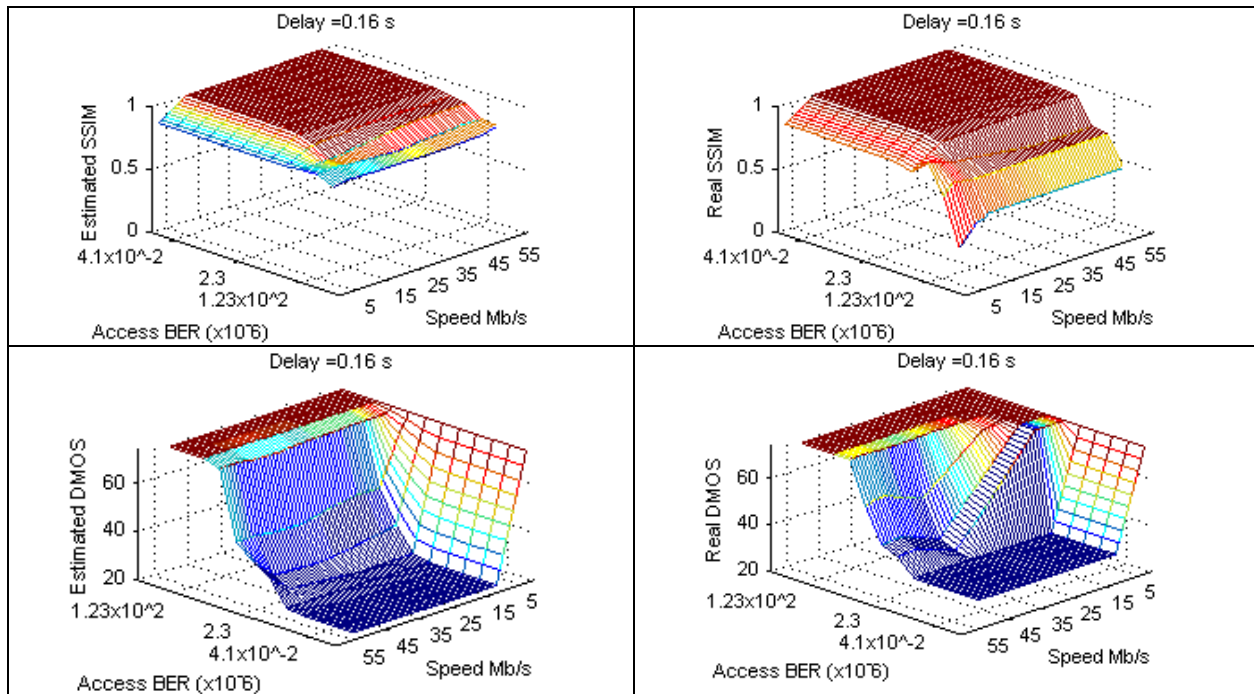


Figura A15 - SSIM e DMOS estimado e medido para um atraso fixo de 0.16s para o cenário Dp3.

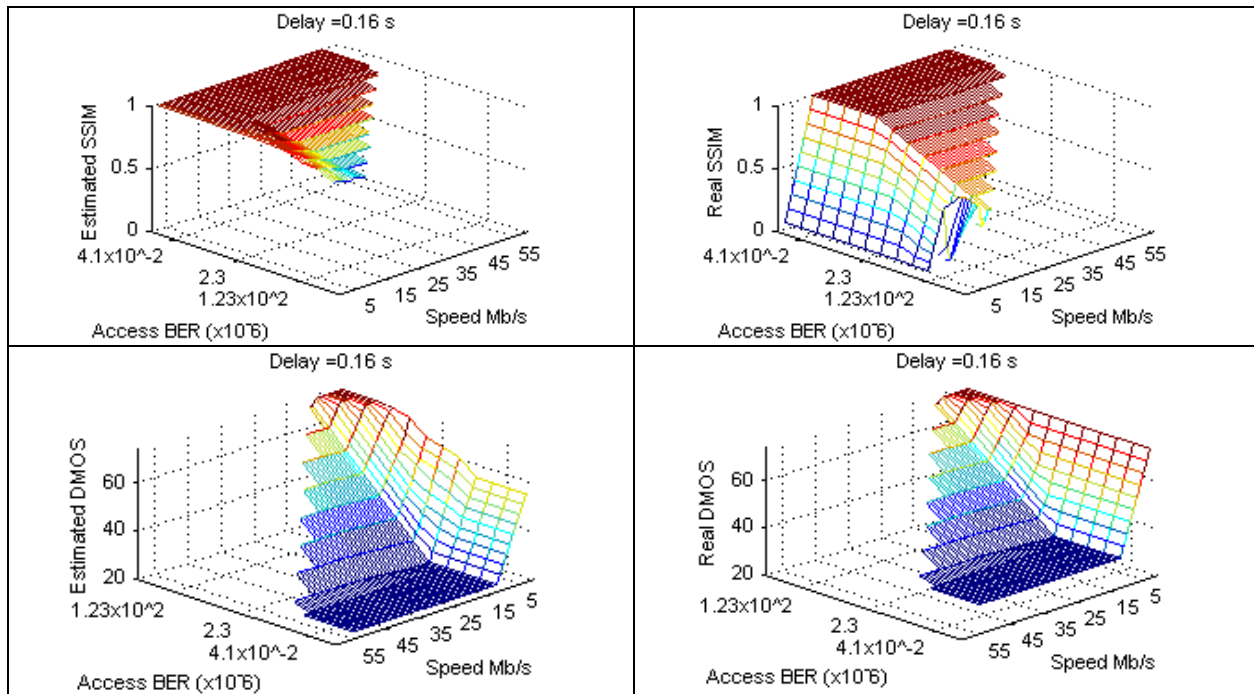


Figura A16 - SSIM e DMOS estimado e medido para um atraso fixo de 0.16s para o cenário Du7.

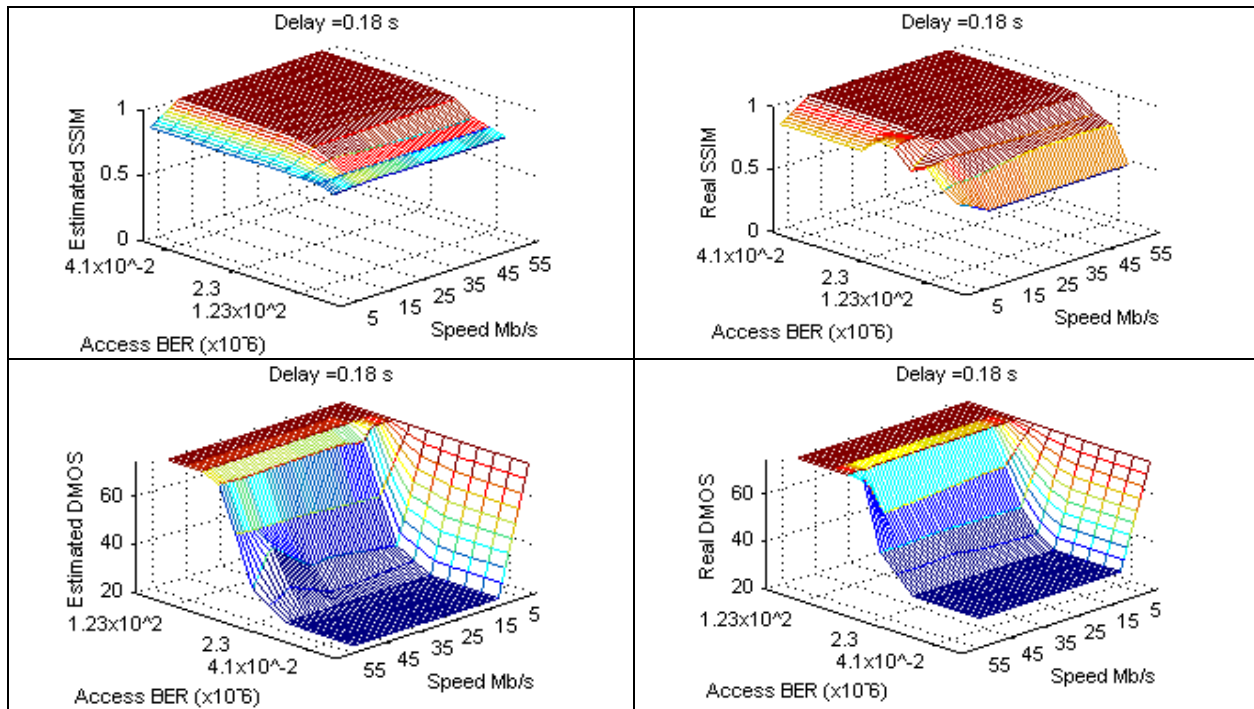


Figura A17 - SSIM e DMOS estimado e medido para um atraso fixo de 0.18s para o cenário Dp3.

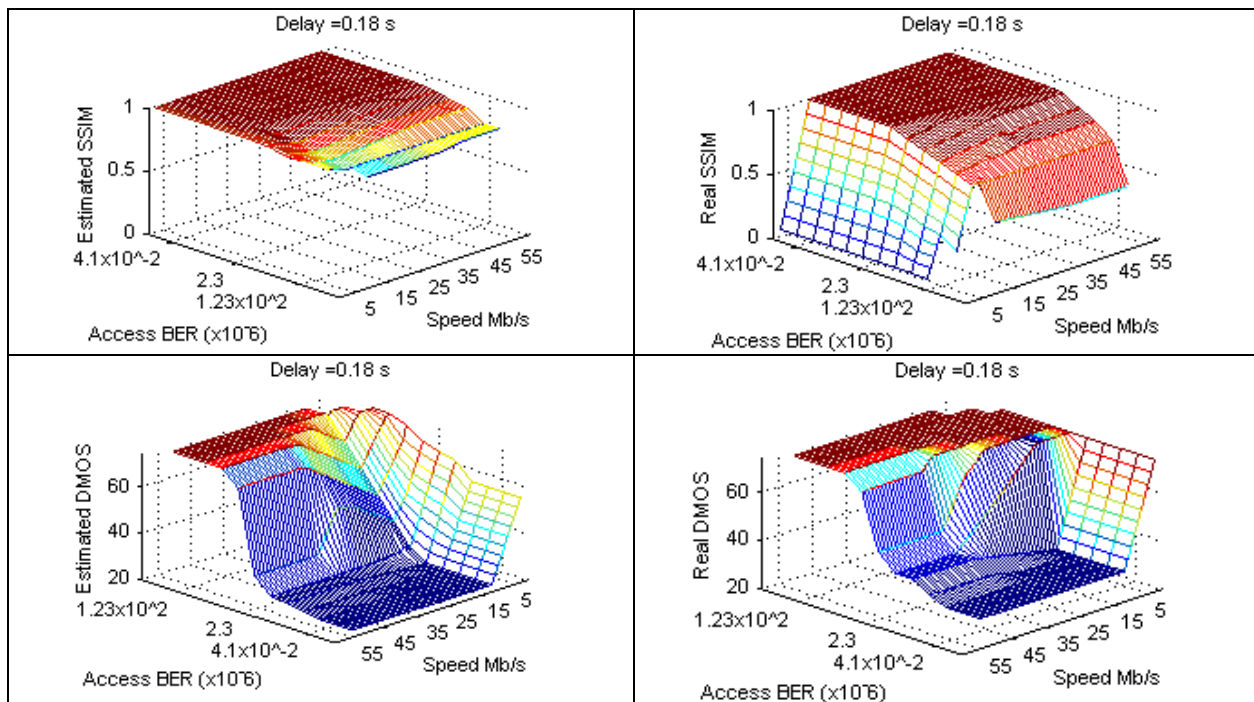


Figura A18 - SSIM e DMOS estimado e medido para um atraso fixo de 0.18s para o cenário Du7.

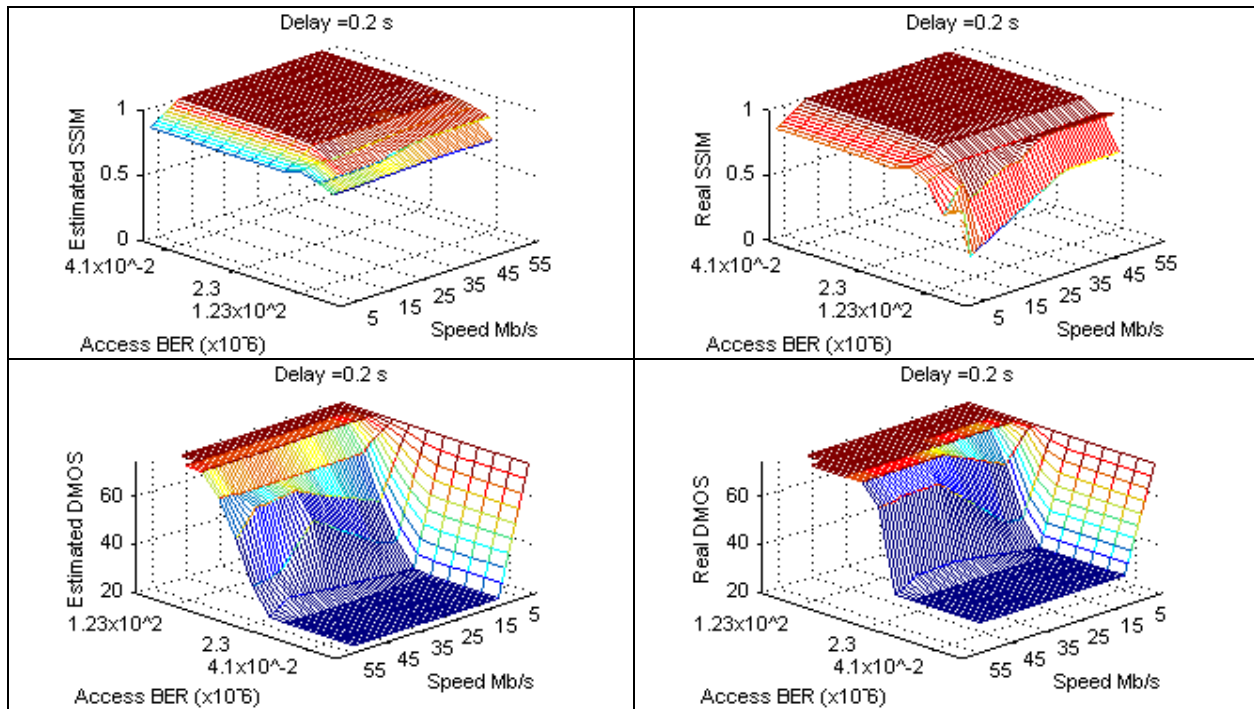


Figura A19 - SSIM e DMOS estimado e medido para um atraso fixo de 0.2s para o cenário Dp3.

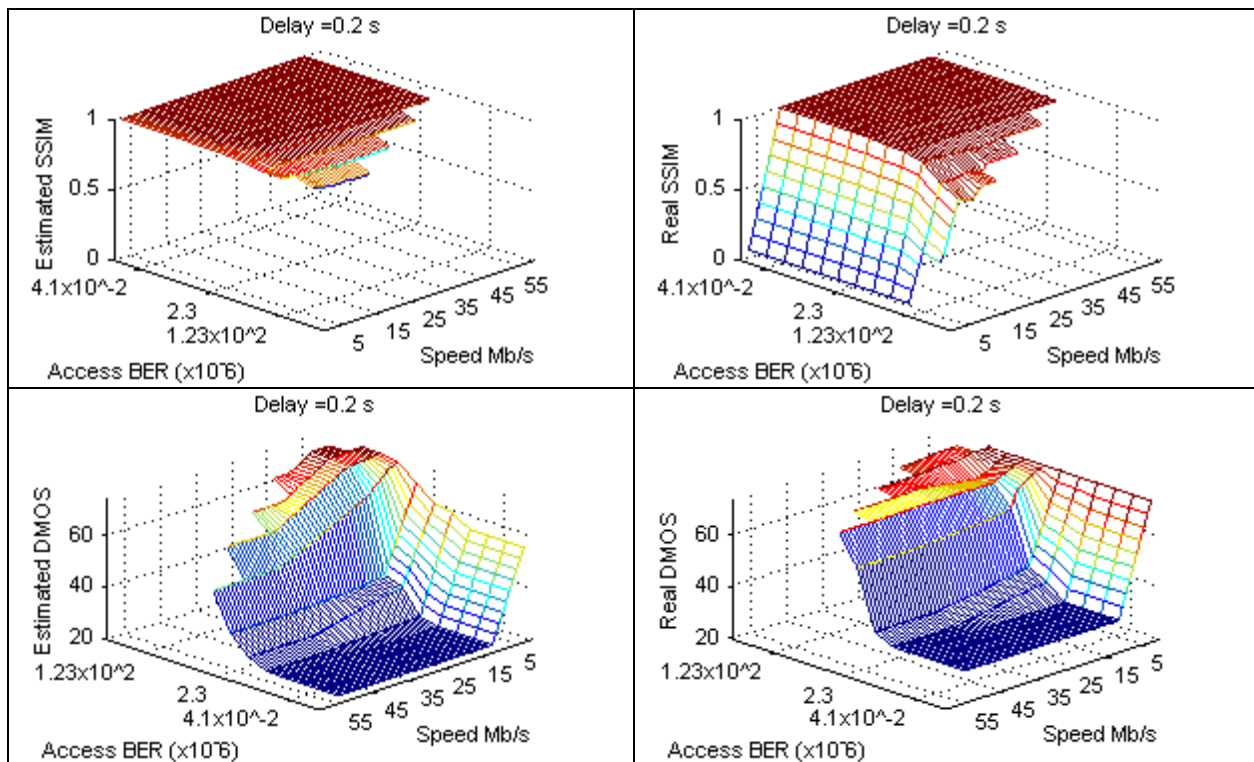


Figura A20 - SSIM e DMOS estimado e medido para um atraso fixo de 0.2s para o cenário Du7.

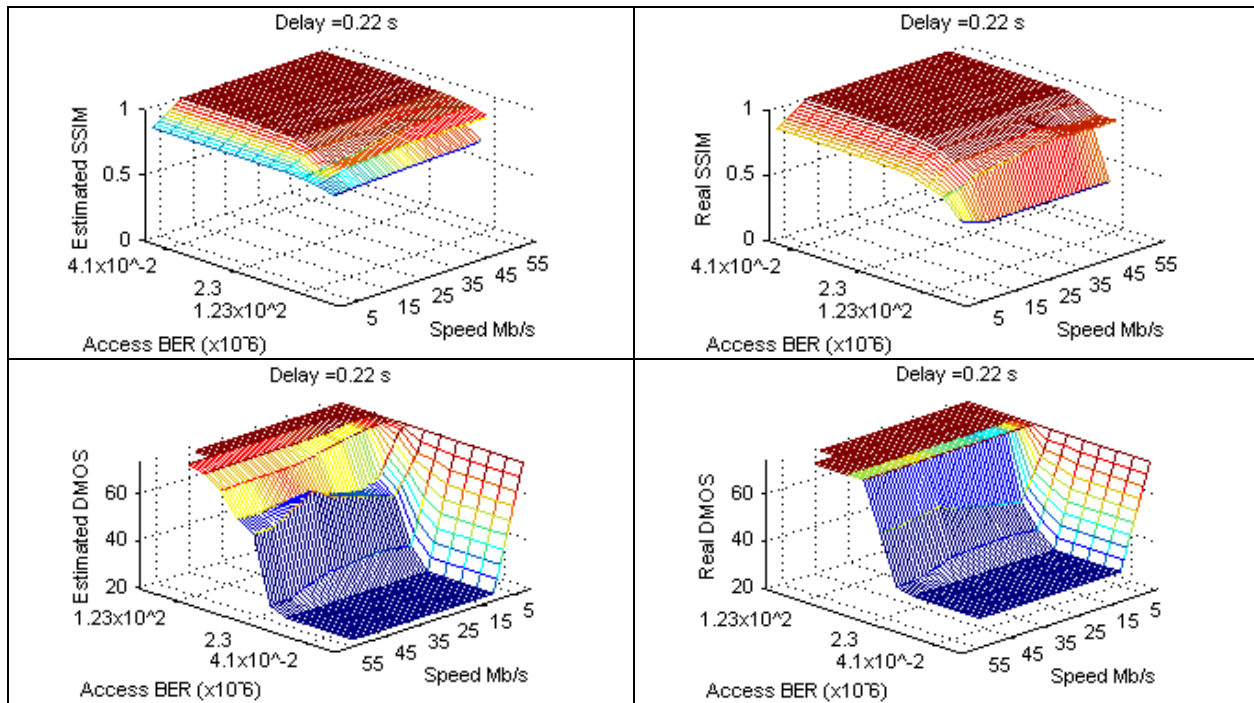


Figura A21 - SSIM e DMOS estimado e medido para um atraso fixo de 0.22s para o cenário Dp3.

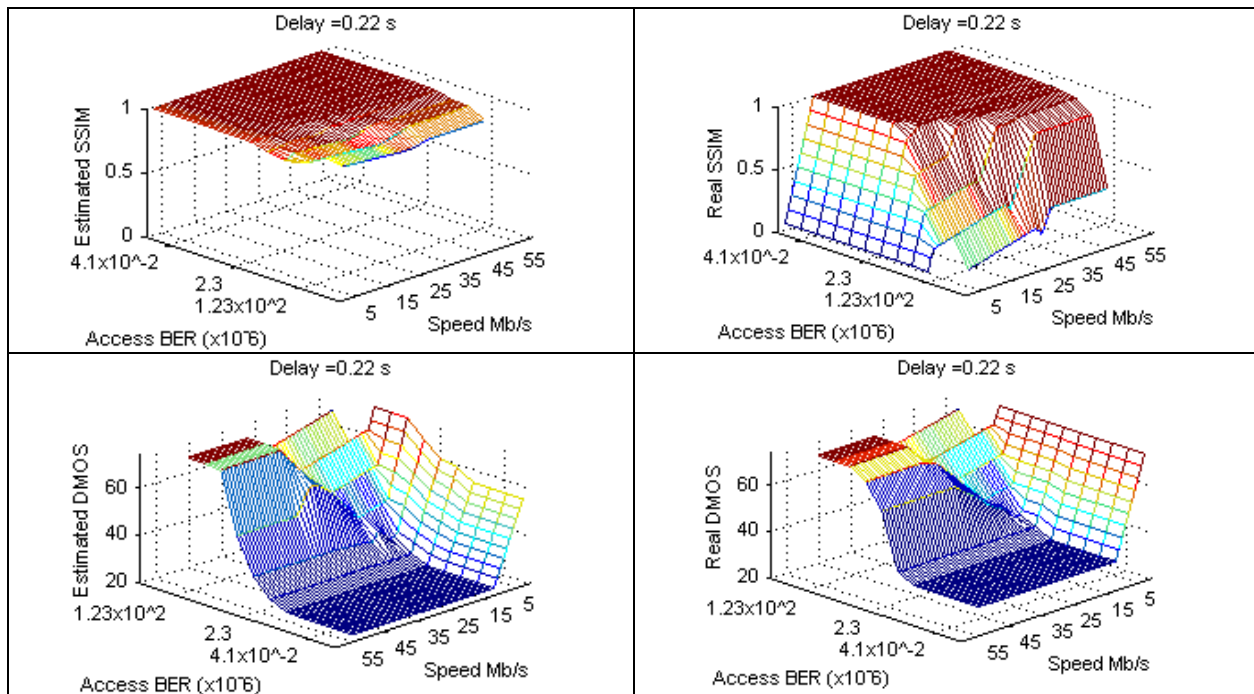


Figura A22 - SSIM e DMOS estimado e medido para um atraso fixo de 0.22s para o cenário Du7.

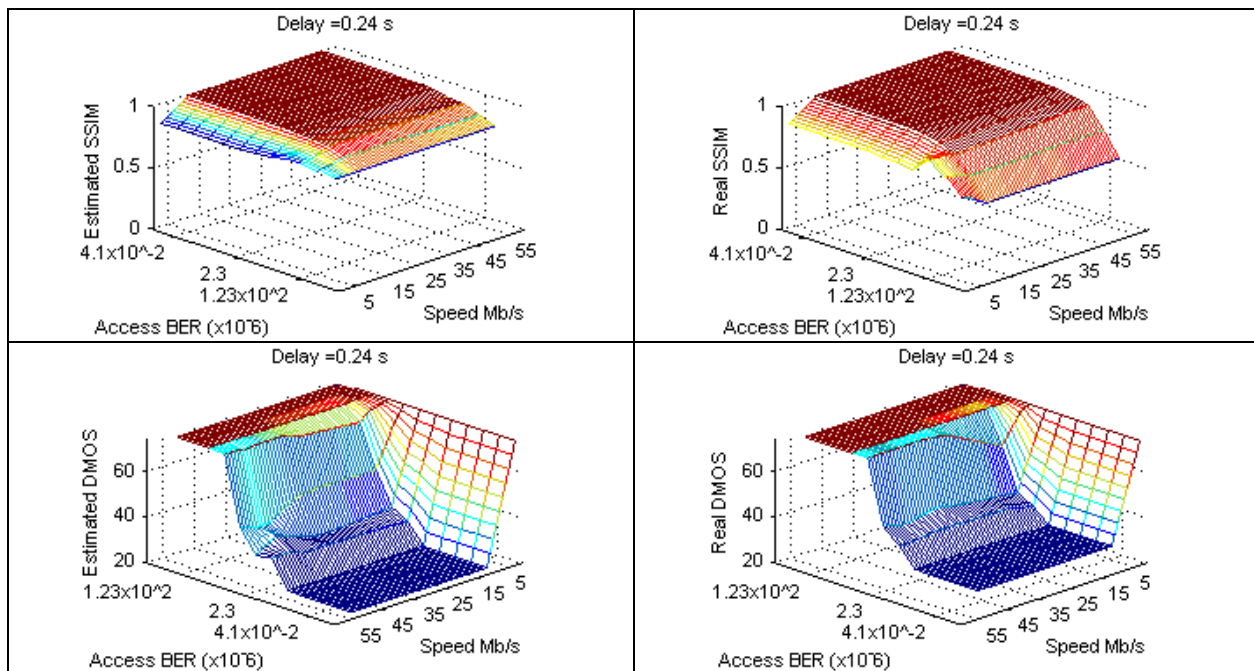


Figura A23 - SSIM e DMOS estimado e medido para um atraso fixo de 0.24s para o cenário Dp3.

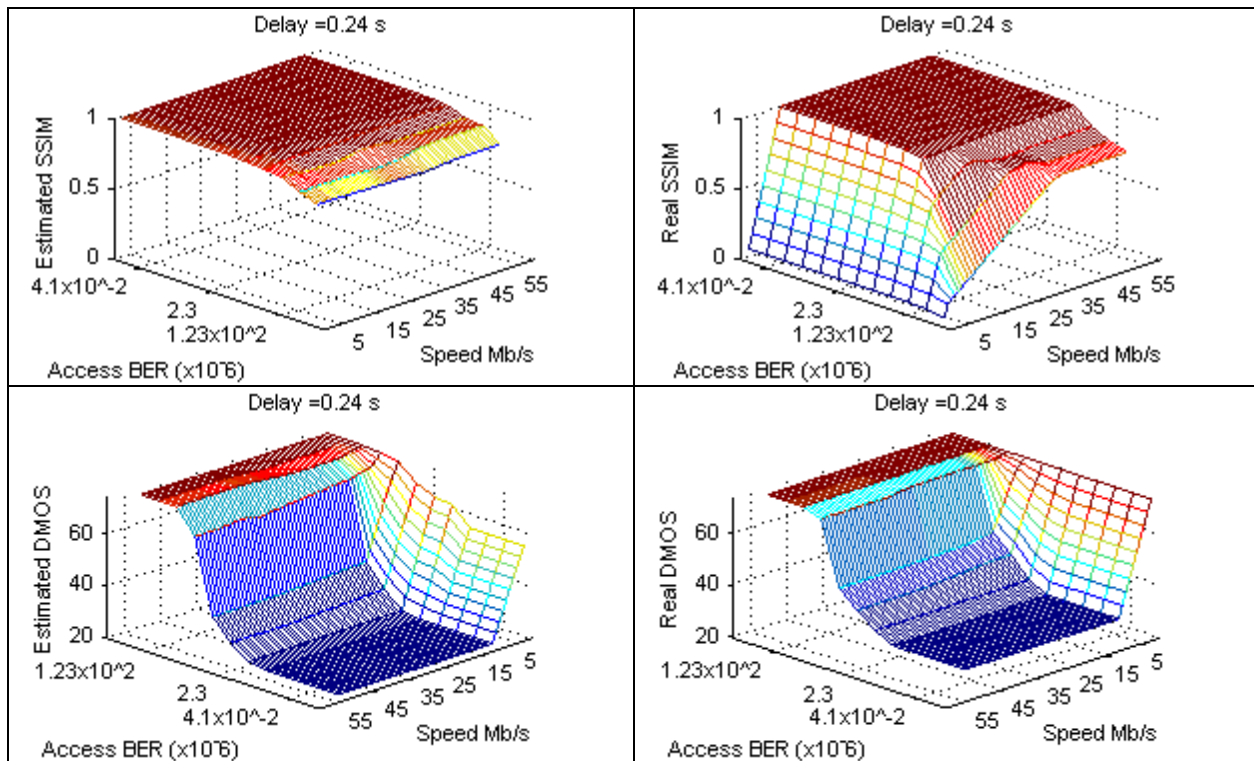


Figura A24 - SSIM e DMOS estimado e medido para um atraso fixo de 0.24s para o cenário Du7.

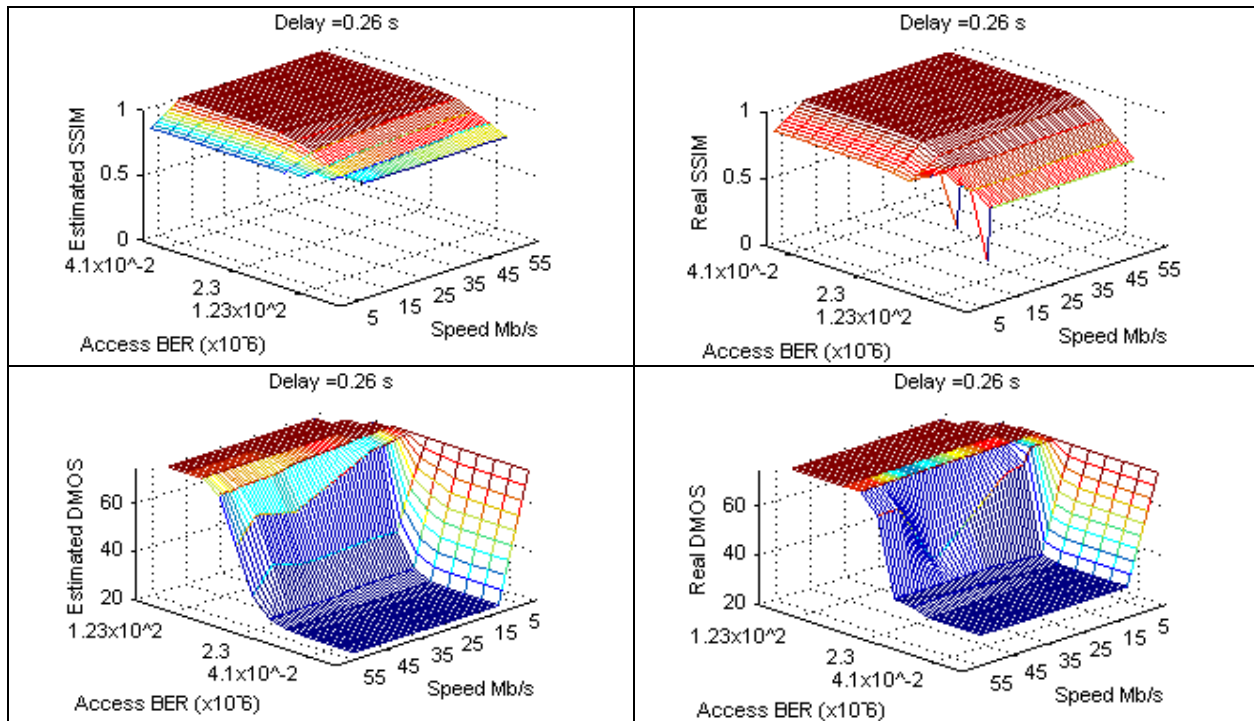


Figura A25 - SSIM e DMOS estimado e medido para um atraso fixo de 0.26s para o cenário Dp3.

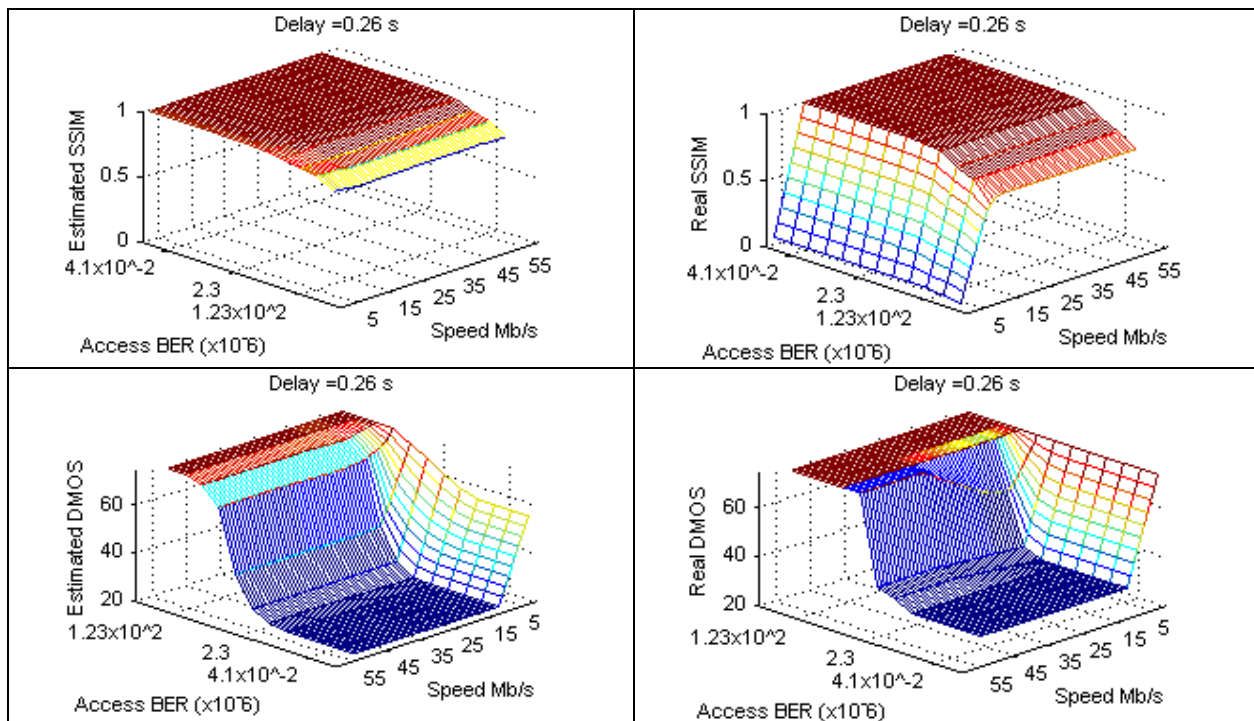


Figura A26 - SSIM e DMOS estimado e medido para um atraso fixo de 0.26s para o cenário Du7.

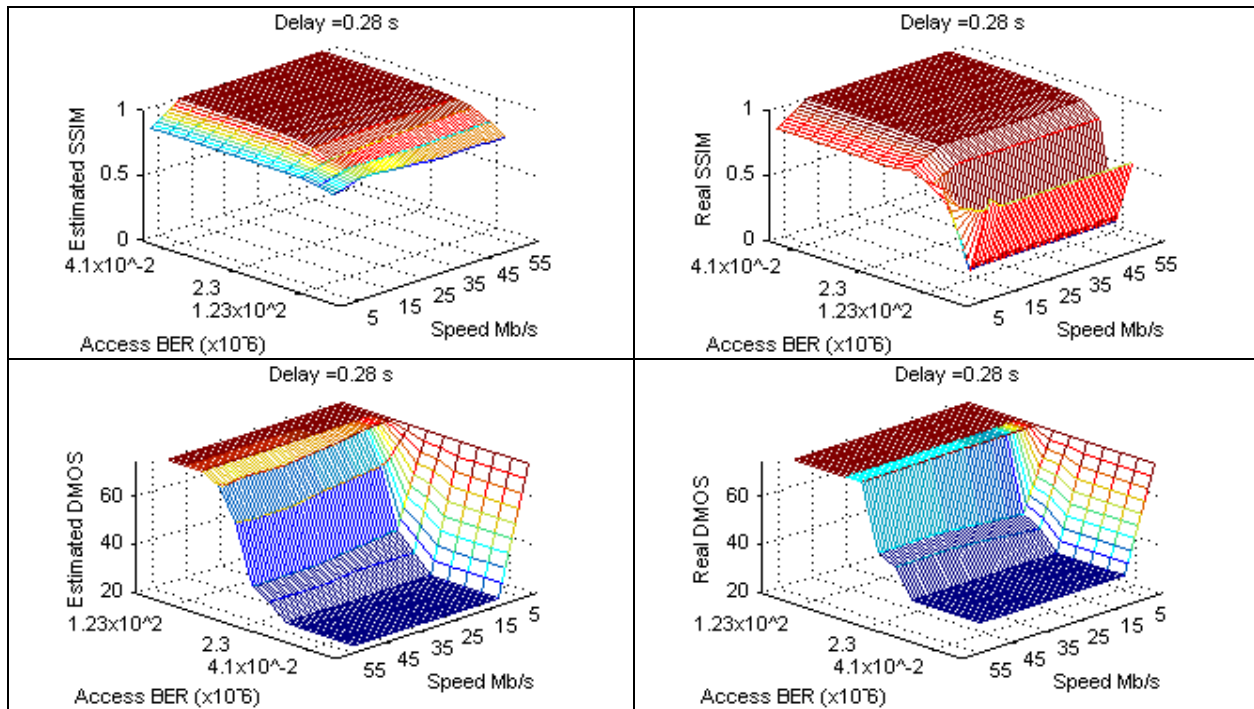


Figura A27 - SSIM e DMOS estimado e medido para um atraso fixo de 0.28s para o cenário Dp3.

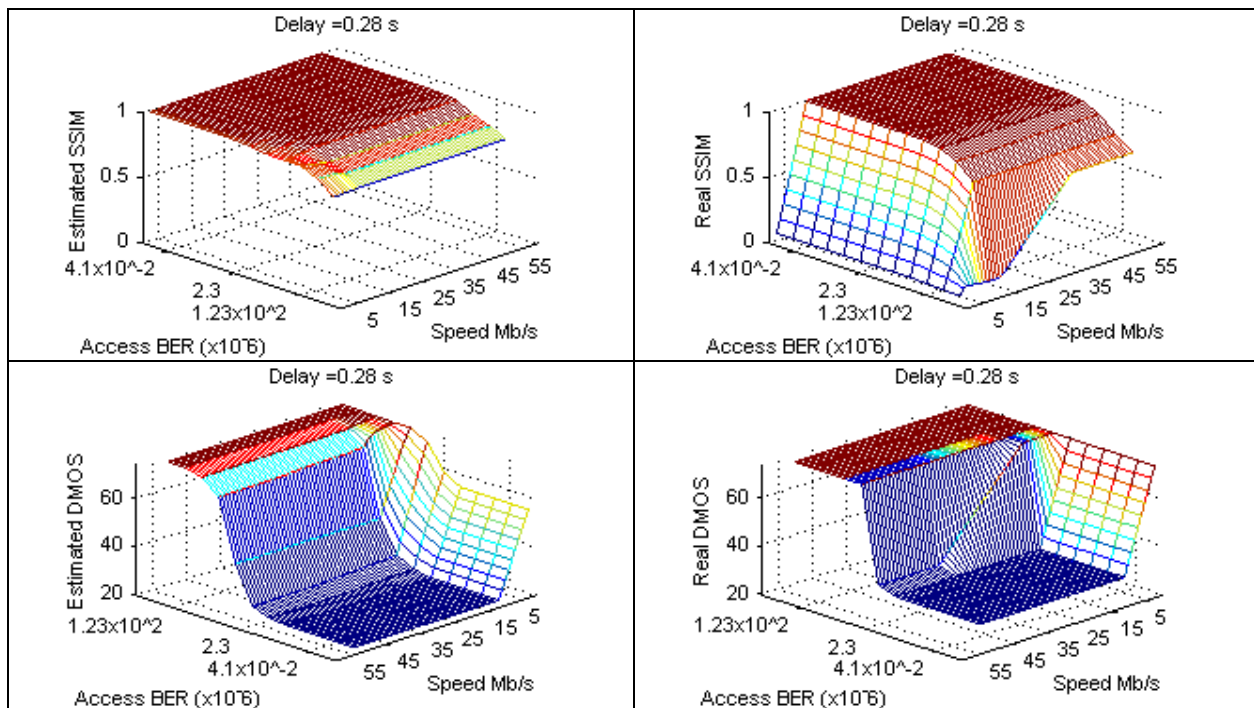


Figura A28 - SSIM e DMOS estimado e medido para um atraso fixo de 0.28s para o cenário Du7.

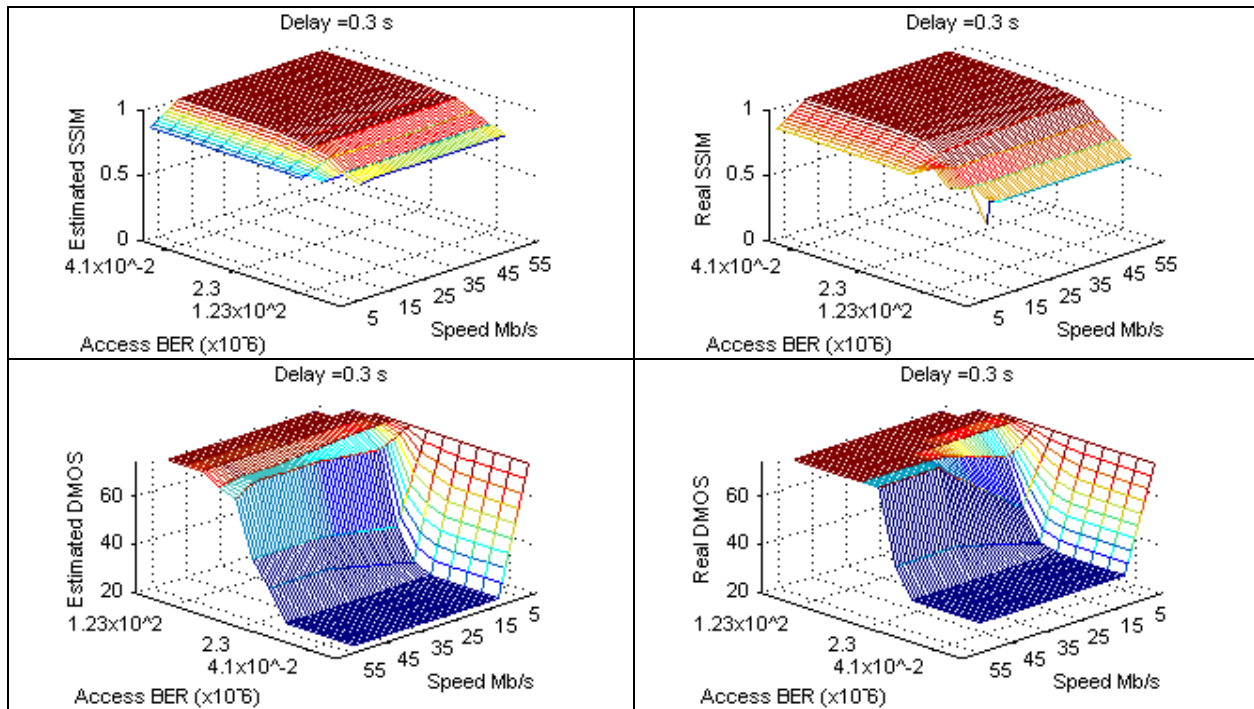


Figura A29 - SSIM e DMOS estimado e medido para um atraso fixo de 0.3s para o cenário Dp3.

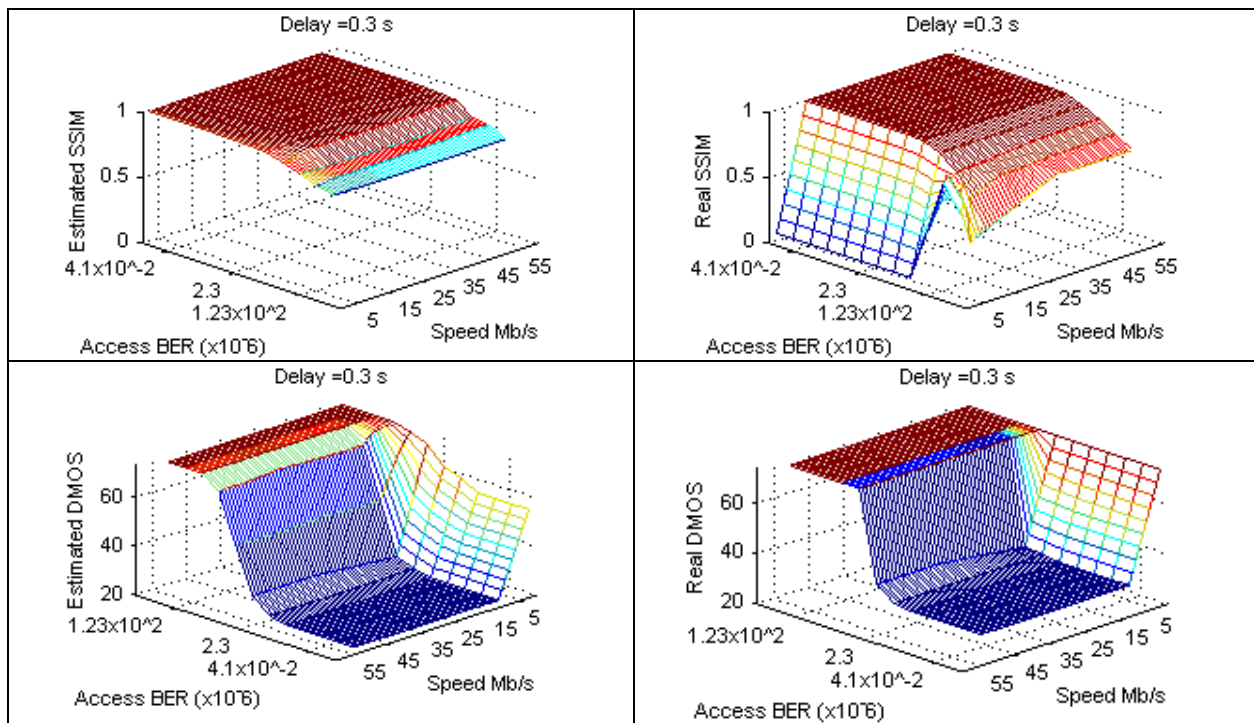


Figura A30 - SSIM e DMOS estimado e medido para um atraso fixo de 0.3s para o cenário Du7.

Anexo B

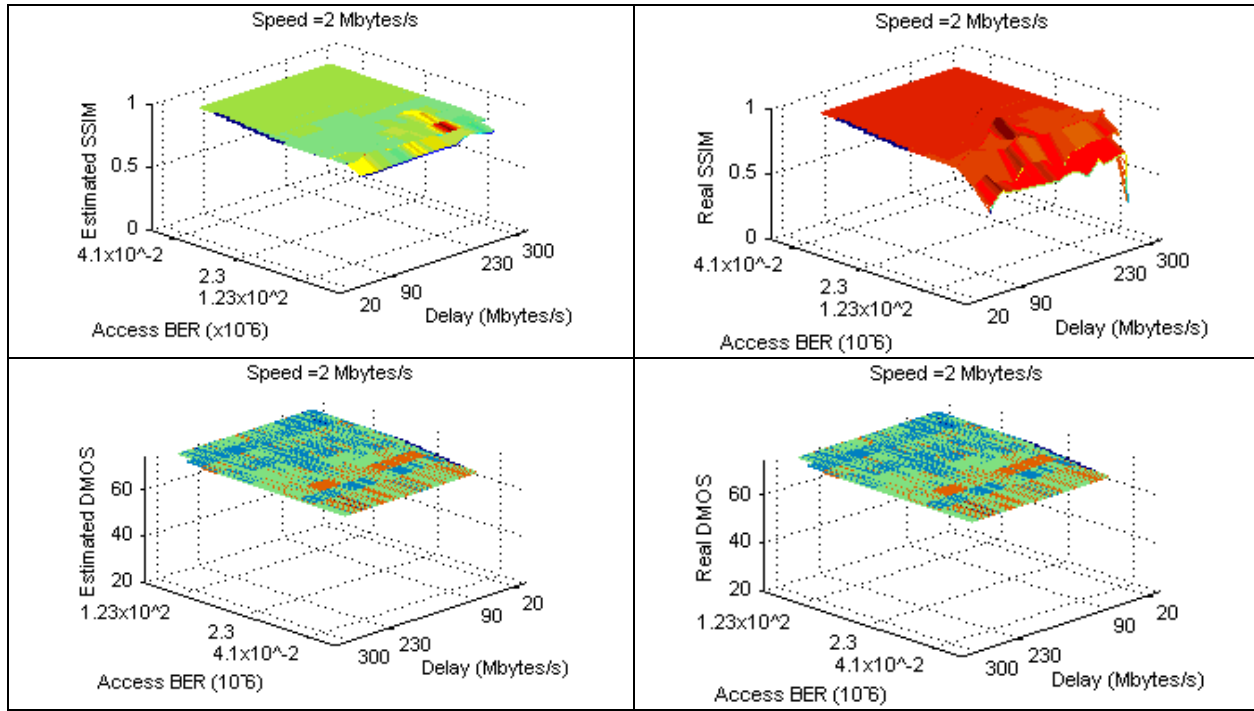


Figura B1 - SSIM e DMOS estimado e medido para uma velocidade fixa de 2 Mbytes/s para o cenário Dp3.

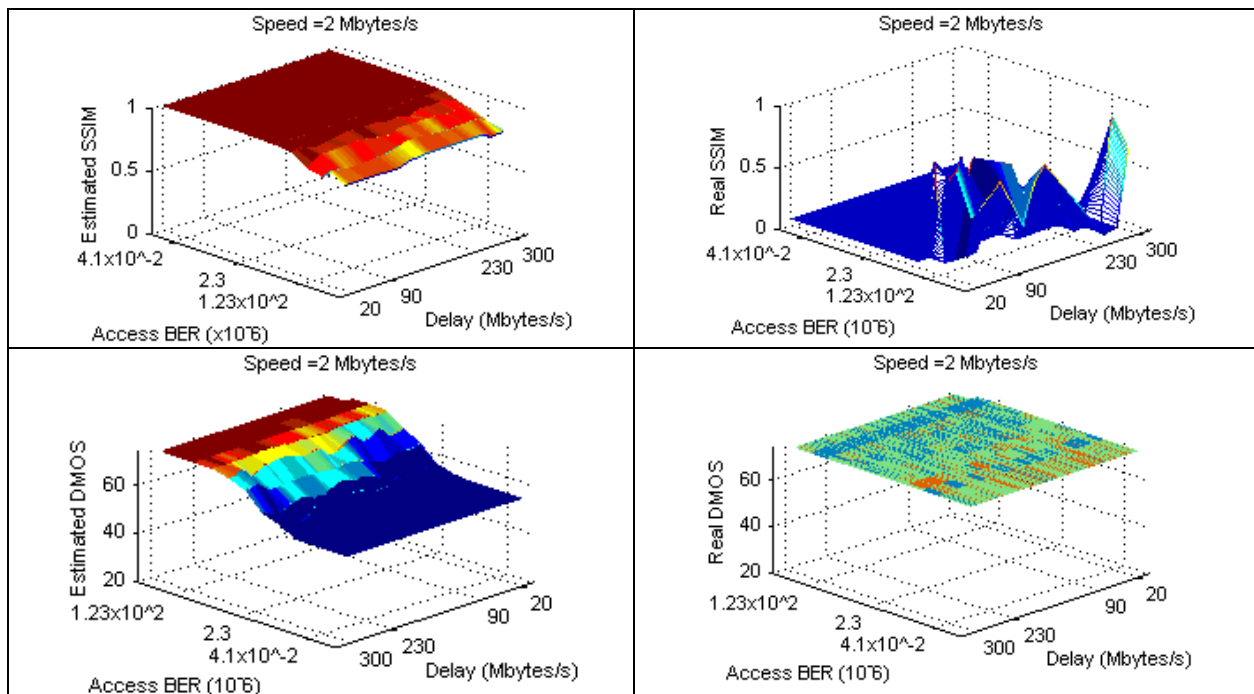


Figura B2 - SSIM e DMOS estimado e medido para uma velocidade fixa de 2 Mbytes/s para o cenário Du7.

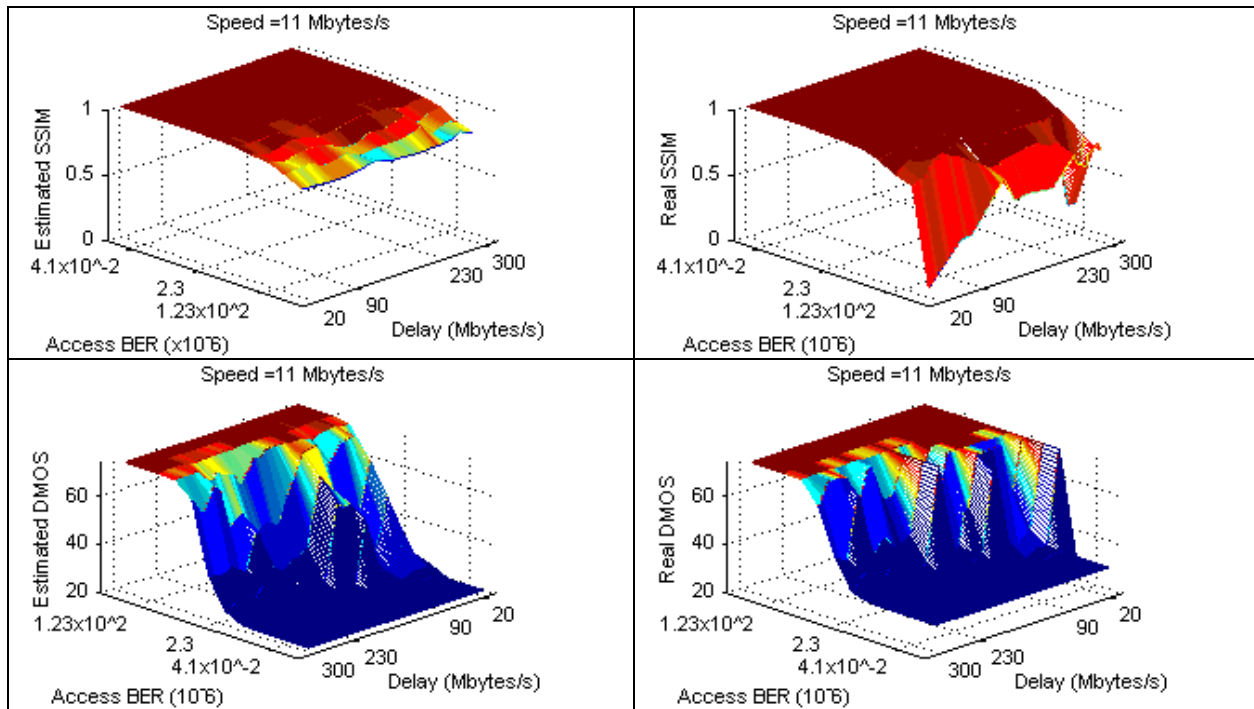


Figura B3 - SSIM e DMOS estimado e medido para uma velocidade fixa de 11 Mbytes/s para o cenário Dp3.

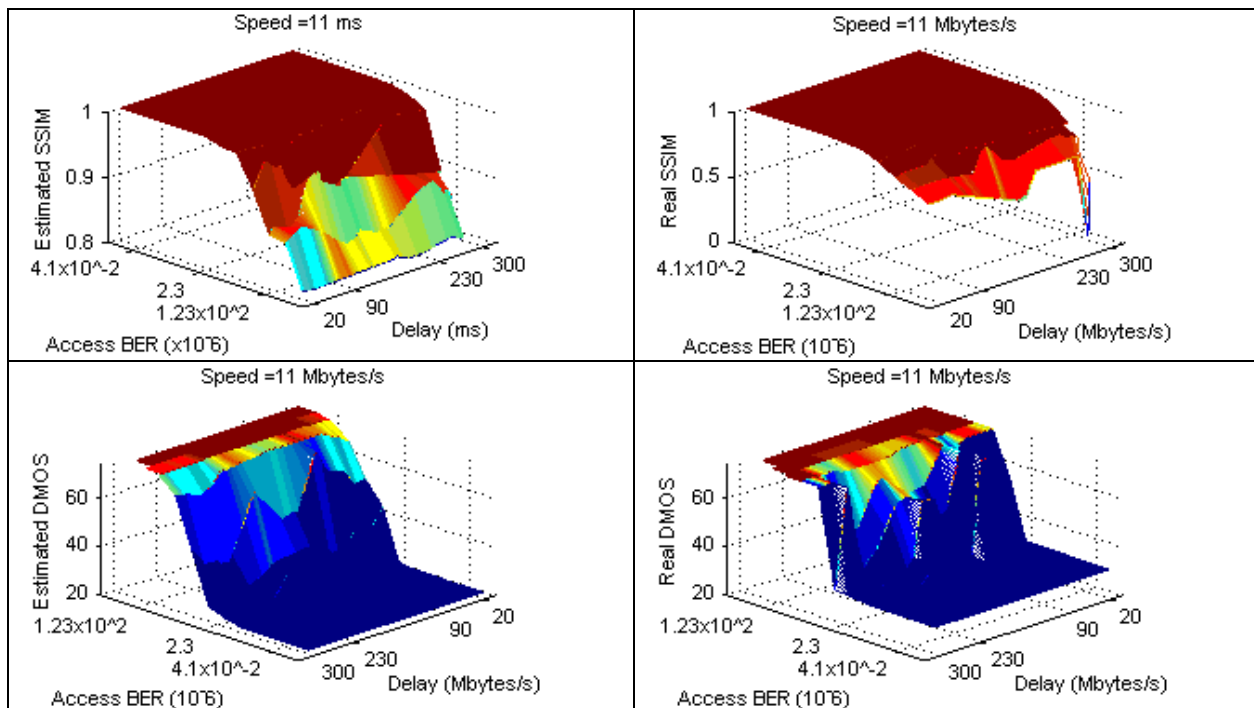


Figura B4 - SSIM e DMOS estimado e medido para uma velocidade fixa de 11 Mbytes/s para o cenário Du7.

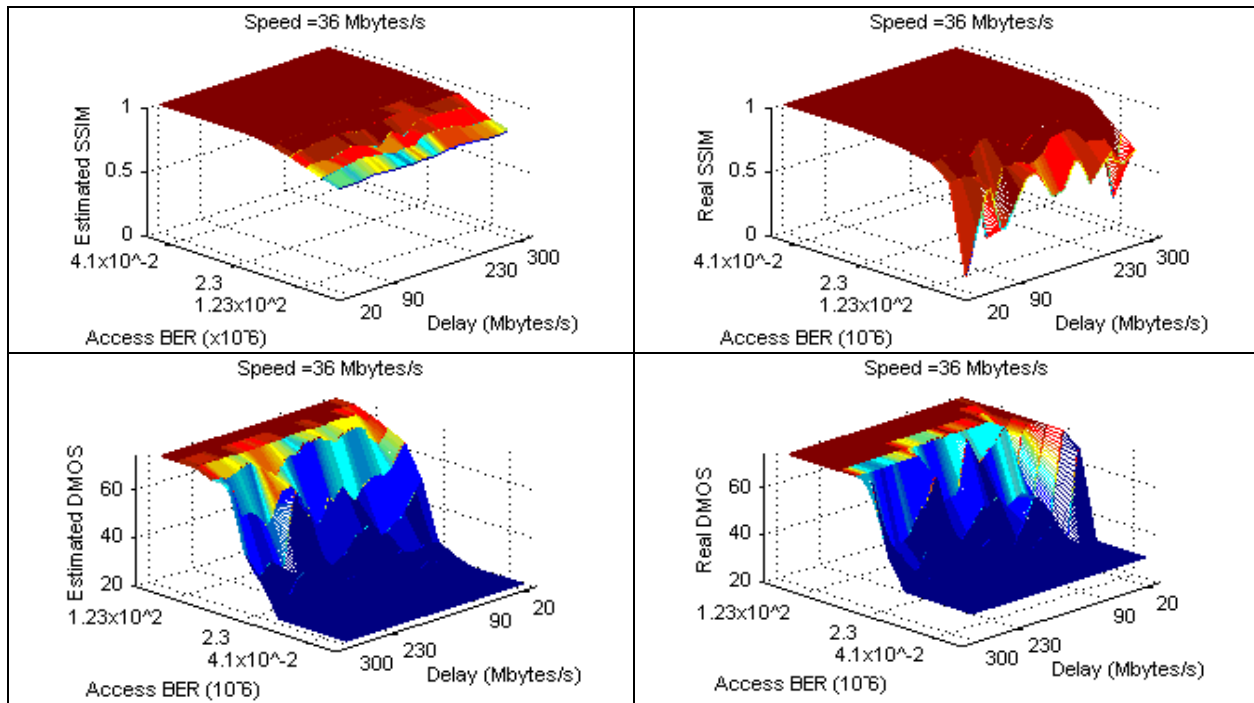


Figura B5 - SSIM e DMOS estimado e medido para uma velocidade fixa de 36 Mbytes/s para o cenário Dp3.

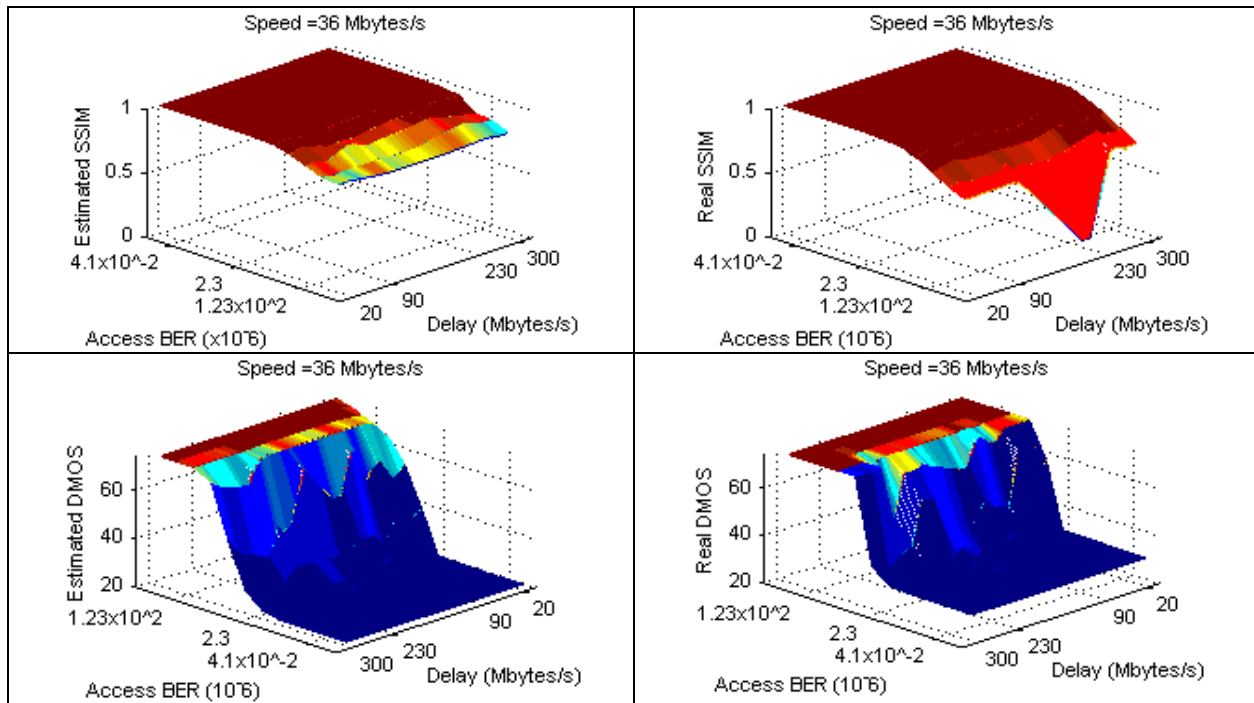


Figura B6 - SSIM e DMOS estimado e medido para uma velocidade fixa de 36 Mbytes/s para o cenário Du7.

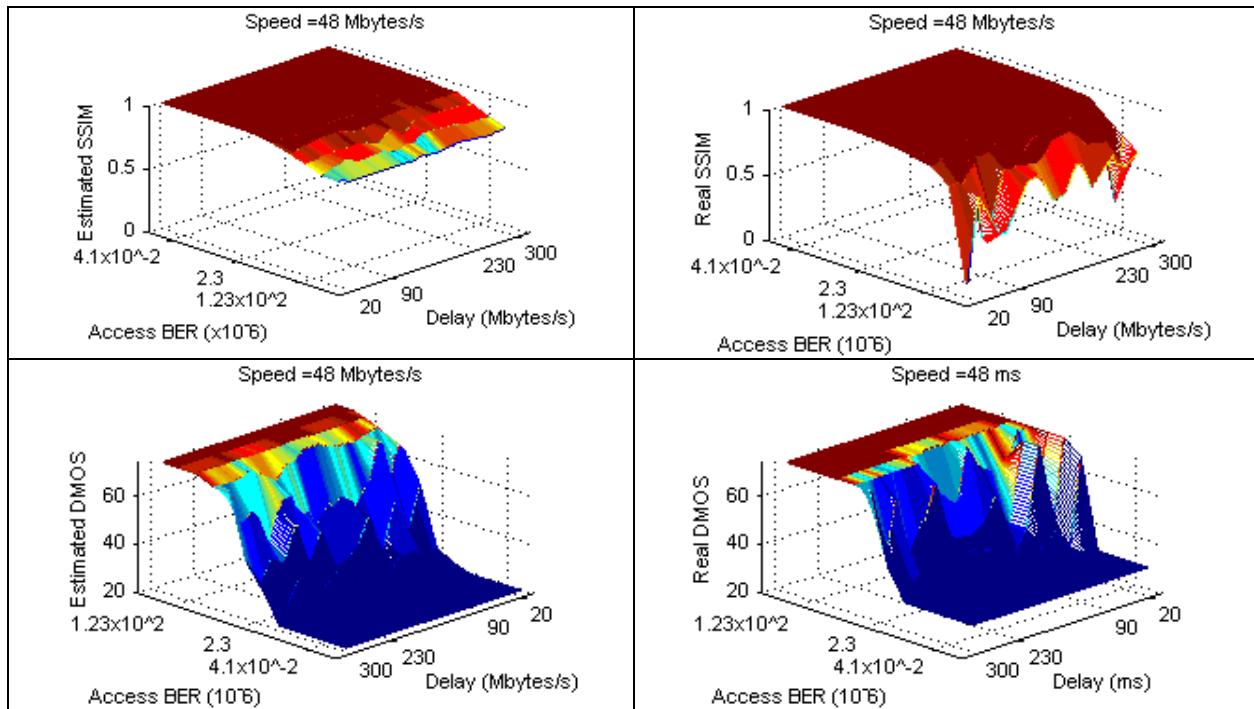


Figura B7 - SSIM e DMOS estimado e medido para uma velocidade fixa de 48 Mbytes/s para o cenário Dp3.

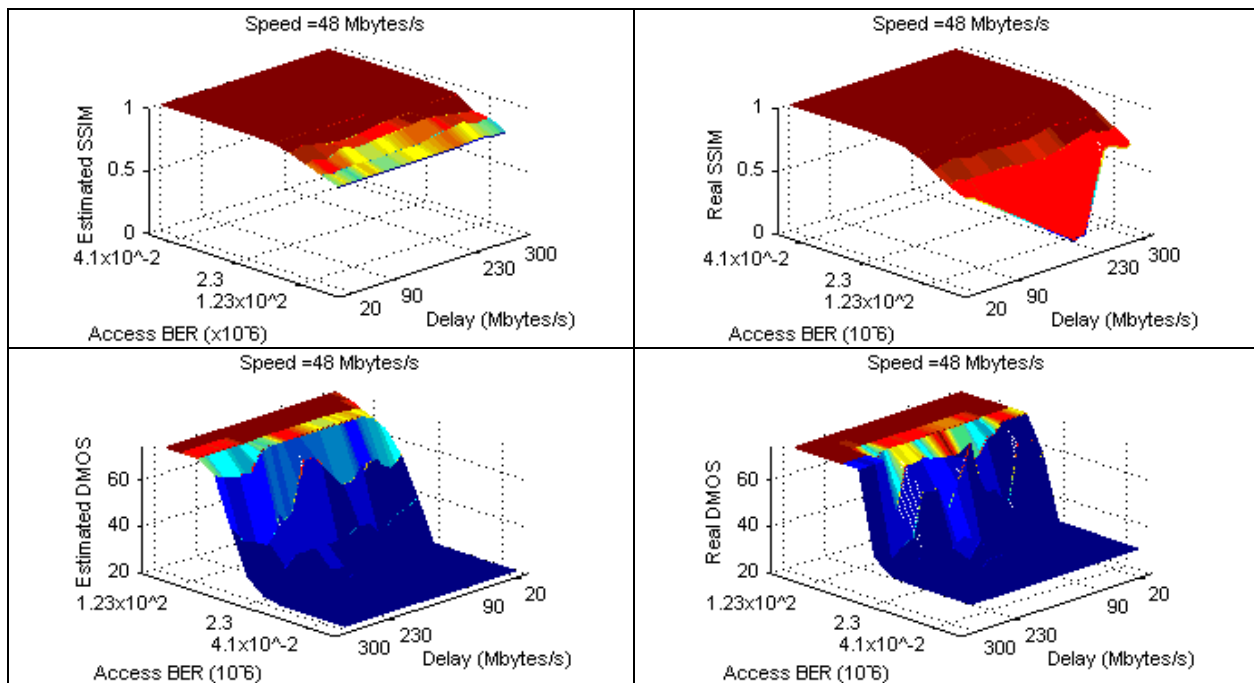


Figura B8 - SSIM e DMOS estimado e medido para uma velocidade fixa de 48 Mbytes/s para o cenário Du7.

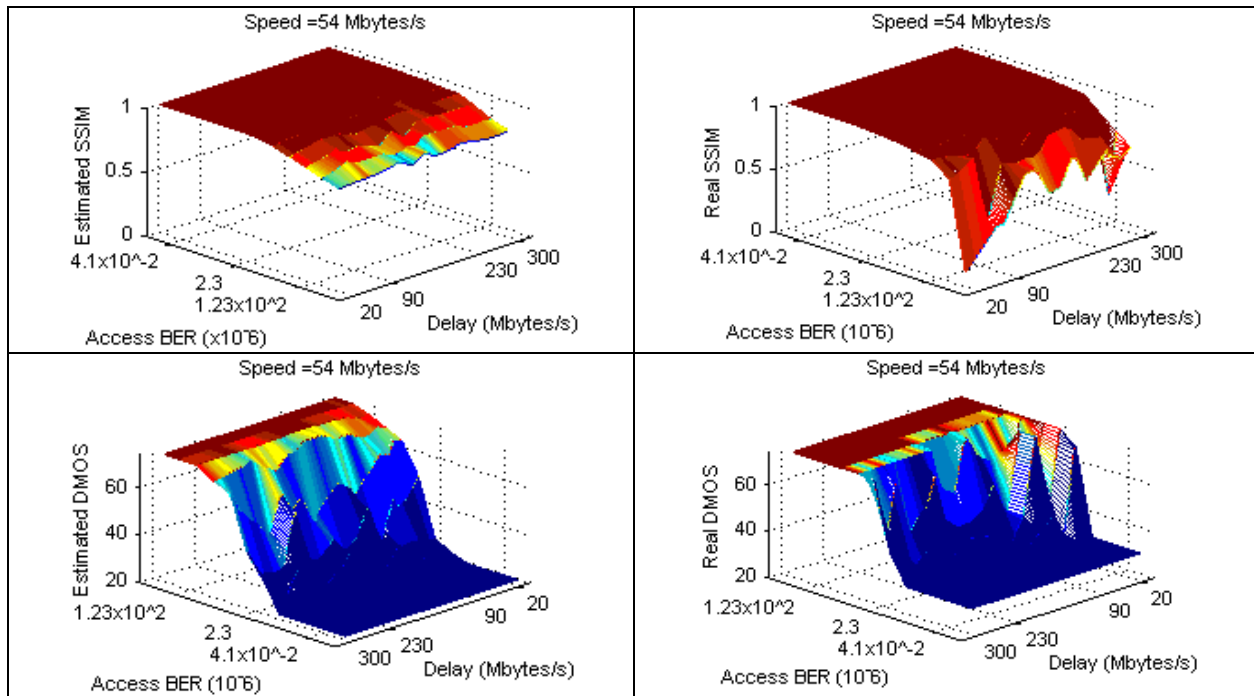


Figura B9 - SSIM e DMOS estimado e medido para uma velocidade fixa de 54 Mbytes/s para o cenário Dp3.

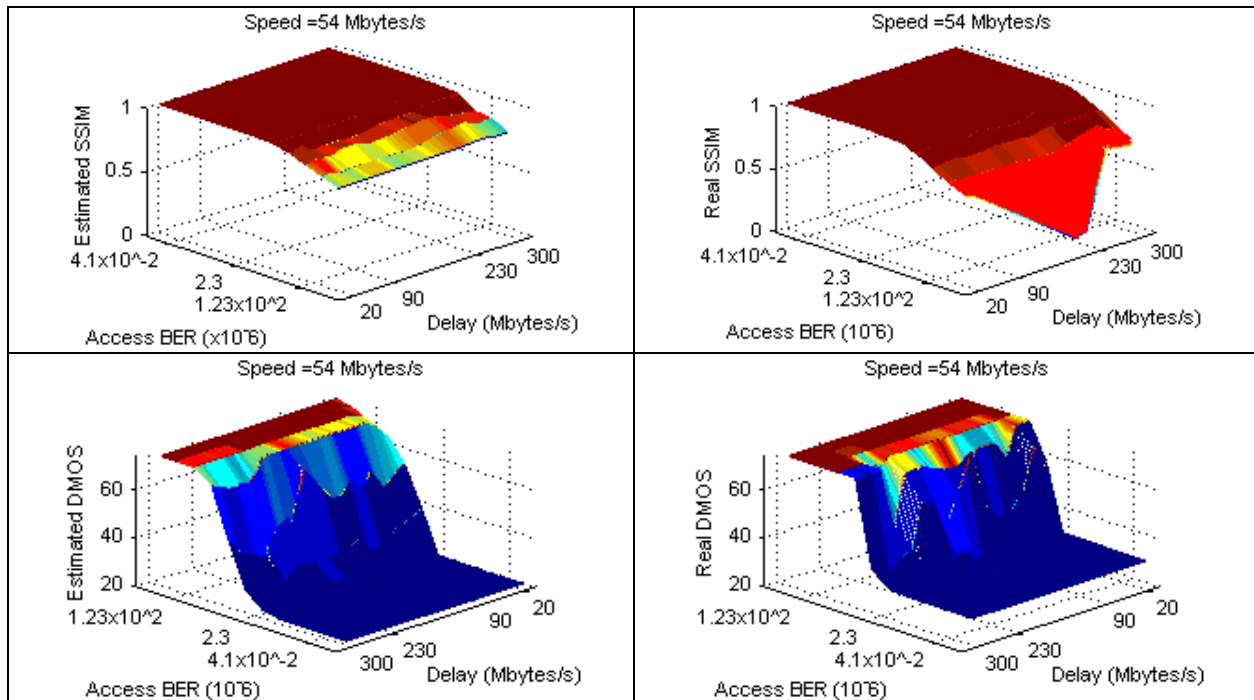


Figura B10 - SSIM e DMOS estimado e medido para uma velocidade fixa de 54 Mbytes/s para o cenário Du7.

Anexo C

Network Element	Impairment / Interferer	Parameter Range
Core		
	Number of Switches	3 to 15
	Total Core Link Delay (ms)	10 to 250
Access (pick one technology)		
GPON	Access Rate Down (Mbit/s)	5 to 50
	Access Rate Up (Mbit/s)	2 to 35
	Residual BER	10^{-12} to 10^{-9}
	Delay (ms)	1
DSL	Access Rate Down (Mbit/s)	3 to 33
	Access Rate Up (Mbit/s)	1 to 3
	Residual BER	10^{-8} to 10^{-6}
	Delay (ms)	1
Managed Bandwidth		
	IPTV HD Stream 1 - CBR (qty)	0 to 1 (10 core-only)
	Downstream Rate (Mbit/s)	8
	IPTV HD Stream 2 - VBR (qty)	0 to 1 (10 core-only)
	Downstream Rate (Mbit/s)	8
	IPTV SD Stream 1 - CBR (qty)	0 to 1 (10 core-only)
	Downstream Rate (Mbit/s)	2
	IPTV SD Stream 2 - VBR (qty)	0 to 1 (10 core-only)
	Downstream Rate (Mbit/s)	2
	VoIP/Fax (qty)	0 to 1 (10 core-only)
	Rate down/up (Mbit/s)	0.064 / 0.064
Residual Bandwidth		
	Peer-to-peer Rate Down / Up	1.89 / 0.768
	HTTP Rate Down / Up	1.7 / 0.007
	OTT1 Rate Down / Up	7 / 0.128
	OTT2 Rate Down / Up	1 / 0.025
	VoIP/FoIP Rate Down / Up	0.064 / 0.064

Figura C1 - Tabela com as gamas de valores possíveis atribuídos aos parâmetros dos elementos de rede [26].

Network Element	Impairment / Interferer	PCAP Avg BW (Mb/s)	Partially Managed					
			p1	p2	p3	p4	p5	p6
Core								
	Number of Switches		3	4	5	8	10	12
	Total Core Link Delay (ms)		10	25	50	75	100	200
Access (pick one technology)								
GPON	Access Rate Down (Mbit/s)		35	35	25	15	5	15
	Access Rate Up (Mbit/s)		35	15	25	5	2	5
	Residual BER		1.E-12	1.E-12	1.E-11	1.E-09	1.E-09	1.E-09
DSL	Access Rate Down (Mbit/s)		24	18	12	6	3	6
	Access Rate Up (Mbit/s)		3	1.5	1.5	1	1	1
	Residual BER		1.E-08	1.E-07	1.E-07	1.E-06	1.E-06	1.E-06
	Managed Bandwidth	(QoS Voice=1 Video=2)						
	IPTV HD Stream 1 - CBR (qty)	8						
	IPTV HD Stream 2 - VBR (qty)	8						
	IPTV SD Stream 1 - CBR (qty)	2	1	1	1	1	0	1
	IPTV SD Stream 2 - VBR (qty)	2	1	1	1	1	1	1
	VoIP/Fax (qty)	0.064/0.064	1	1	1	1	1	1
	Residual Bandwidth	QoS = 7						
	Peer-to-peer Rate Down	1.89		10%	50%	25%	20%	25%
	Peer-to-peer Rate Up	0.285		135%	135%	135%	135%	135%
	HTTP Rate Down	1.58		10%	100%	40%	20%	40%
	HTTP Rate Up	0.66		10%	50%	50%	60%	50%
	OTT1 Rate Down	6.678	100%	100%				
	OTT1 Rate Up	0.099	100%	100%				
	OTT2 Rate Down	2.66	100%	100%	100%			
	OTT2 Rate Up	0.051	100%	100%	100%			
	VoIP/Fax Rate Down	0.064	100%	100%	100%	100%	100%	100%
	VoIP/Fax Rate Up	0.064	100%	100%	100%	100%	100%	100%

Figura C2 - Casos de Teste padrão configurados no simulador para uma rede Partially-Managed. Apresenta os valores dos parâmetros para cada um deles [26].

Network Element	Impairment / Interferer	PCAP Avg BW (Mb/s)	Unmanaged							Core
			u1	u2	u3	u4	u5	u6	u7	c1
Core										
	Number of Switches		3	4	5	8	10	12	15	3
	Total Core Link Delay (ms)		10	25	50	75	100	200	250	150
Access (pick one technology)										
GPON	Access Rate Down (Mbit/s)		35	35	25	15	5	15	5	1000
	Access Rate Up (Mbit/s)		35	15	25	5	2	5	2	1000
	Residual BER		1.E-12	1.E-12	1.E-11	1.E-09	1.E-09	1.E-09	1.E-09	1.E-12
DSL	Access Rate Down (Mbit/s)		24	18	12	6	3	6	3	1000
	Access Rate Up (Mbit/s)		3	1.5	1.5	1	1	1	1	1000
	Residual BER		1.E-08	1.E-07	1.E-07	1.E-06	1.E-06	1.E-06	1.E-06	1.E-12
	Managed Bandwidth	(QoS Voice=1 Video=2)								
	IPTV HD Stream 1 - CBR (qty)	8								10
	IPTV HD Stream 2 - VBR (qty)	8								10
	IPTV SD Stream 1 - CBR (qty)	2								10
	IPTV SD Stream 2 - VBR (qty)	2								10
	VoIP/Fax (qty)	0.064/0.064								10
	Residual Bandwidth	QoS = 7								
	Peer-to-peer Rate Down	1.89		10%	50%	25%	20%	25%	20%	1000%
	Peer-to-peer Rate Up	0.285		135%	135%	135%	135%	135%	135%	2000%
	HTTP Rate Down	1.58		10%	100%	40%	20%	40%	20%	1000%
	HTTP Rate Up	0.66		10%	50%	50%	60%	50%	60%	1000%
	OTT1 Rate Down	6.678	100%	100%						500%
	OTT1 Rate Up	0.099	100%	100%						500%
	OTT2 Rate Down	2.66	200%	200%	200%	100%	50%	100%	50%	500%
	OTT2 Rate Up	0.051	200%	200%	200%	100%	50%	100%	50%	500%
	VoIP/Fax Rate Down	0.064	100%	100%	100%	100%	100%	100%	100%	1000%
	VoIP/Fax Rate Up	0.064	100%	100%	100%	100%	100%	100%	100%	1000%

Figura C3 - Casos de Teste padrão configurados no simulador para uma rede Unmanaged. Apresenta os valores dos parâmetros para cada um deles [26].