

UNIVERSIDADE FEDERAL DO PARANÁ

DIEGO FRANCO MARTINS
FELIPE DE SOUZA PINHEIRO

AVALIAÇÃO DA QUALIDADE DE STREAMING DE VÍDEO EM SISTEMAS
LTE

CURITIBA

2014

DIEGO FRANCO MARTINS
FELIPE DE SOUZA PINHEIRO

AVALIAÇÃO DA QUALIDADE DE STREAMING DE VÍDEO EM SISTEMAS
LTE

Trabalho de Conclusão de Curso
apresentado como exigência parcial para
obtenção do grau de Bacharel em
Engenharia Elétrica, Ênfase em Sistema
Eletrônicos Embarcados, à Universidade
Federal do Paraná.

Orientador: Prof. Dr. Carlos Marcelo
Pedroso.

CURITIBA
2014

DIEGO FRANCO MARTINS
FELIPE DE SOUZA PINHEIRO

Título: AVALIAÇÃO DA QUALIDADE DE STREAMING DE VÍDEO EM
SISTEMAS LTE.

Trabalho de Conclusão de Curso apresentado como exigência parcial
para obtenção do grau de Bacharel em Engenharia Elétrica com ênfase em
Sistema Eletrônicos Embarcados, à Universidade Federal do Paraná.

Data: 05 de dezembro de 2014.

Orientador: Prof. Dr. Carlos Marcelo Pedroso
Universidade Federal do Paraná

Prof. Dr. Evelio Martín García Fernández
Universidade Federal do Paraná

Prof. Dr. Luis Henrique Assumpção Lolis
Universidade Federal do Paraná

AGRADECIMENTOS

Nossos agradecimentos, nesse momento, não contemplam todas as pessoas que de modo significativo fazem parte de nossa trajetória na elaboração desse trabalho. Assim, pedimos licença àquelas que não estão presentes entre essas palavras, mas estejam certas que estão em nossos pensamentos e em nossa profunda gratidão.

Ao Prof. Dr. Carlos Marcelo Pedroso pelo seu empenho e pela oportunidade de desenvolver esse trabalho sob sua orientação.

Agradecemos a todos as pessoas que conviveram conosco durante este período e foram fundamentais para esse triunfo, pois em algum momento prestaram alguma ajuda, palavras de apoio ou colaboraram de alguma forma que contribuísse com nossa formação.

No entanto, não podemos deixar de fazer um agradecimento especial aos nossos pais, que durante todas nossas vidas nos deram todo o suporte que precisávamos e nossas esposas, por todo amor e paciência.

Enfim, àqueles que por algum motivo contribuíram para a realização deste objetivo e que fizeram com que alcançássemos a nossa meta.

À todos nossa eterna gratidão.

RESUMO

Este trabalho tem por objetivo a avaliação da degradação da qualidade percebida pelos usuários, para streaming de vídeos codificados em H.264/AVC sob redes LTE, também conhecidas como 4G. Serão verificadas as influências da variação do tráfego de fundo da rede, alterando a quantidade de usuários, e da variação do tamanho do Group of Pictures (GoP) sobre a Qualidade de Experiência final do usuário. Serão utilizadas as técnicas objetivas PSNR (Relação Sinal Ruído de Pico) e SSIM (Índice de Similaridade Estrutural). Para alcançar os resultados foram utilizadas simulações computacionais, utilizando o software para simulações de redes NS3. Os resultados obtidos demonstram o aumento da degradação da qualidade percebida na medida que ocorre o aumento do tráfego de fundo. É analisado também que para níveis altos de tráfego de fundo um vídeo com valor de GoP maior tende a ter uma degradação menor se comparado a valores de GoP menores.

Palavra chave: multimídia, vídeo, qualidade percebida, LTE.

ABSTRACT

This study aims to evaluate the degradation of viewer's perceived quality for streaming of videos encoded in H.264/AVC under LTE networks, also known as 4G. The influences of the variation of the background traffic, changing the number of users, and the size of the Group of Pictures (GoP) was analyzed on the end user Quality of Experience. The video quality was evaluated using the PSNR (Peak Signal Noise Ratio) and SSIM (structural similarity). To obtain the results, computer simulations were used, with the software Network Simulator 3. The results show an increase of the degradation of viewer's perceived quality which occurs with the increase of background traffic. It is also found that for high levels of background traffic, video with high GoP value tends to have a lower degradation compared to smaller GoP values.

Key Words: Multimedia, video, perceived quality, LTE.

LISTA DE ILUSTRAÇÕES

FIGURA 1 – FLUXO DE CODIFICAÇÃO, TRANSMISSÃO E ANÁLISE DO VÍDEO	16
FIGURA 2 - CRONOLOGIA DA EVOLUÇÃO DE COMPACTAÇÃO DE VÍDEO	19
FIGURA 3 - POSIÇÃO DAS AMOSTRAS DE LUMINÂNCIA E CROMINÂNCIA NOS FORMATOS: (A) 4:4:4; (B) 4:2:2; (C) 4:2:0 MPEG1; (D) 4:2:0 MPEG-4 .	20
FIGURA 4 - DIAGRAMA DO GROUP OF PICTURES	22
FIGURA 5 - BLOCO DE RECURSOS.....	25
FIGURA 6 - ARQUITETURA DE REDE LTE.....	27
FIGURA 7 - REFERÊNCIA COMPLETA.....	29
FIGURA 8 - REFERÊNCIA REDUZIDA	29
FIGURA 9 - FLUXO GERAL QOE MONITOR.....	34
FIGURA 10 – TOPOLOGIA DA REDE	35
FIGURA 11 – (a) VIDEO SOCCER; (b) VÍDEO ICE; (c) VÍDEO CREW; (d) VÍDEO CITY.....	36
FIGURA 12 (a) – GRÁFICO VÍDEO “SOCCER”, SSIM POR VARIAÇÃO DE UE	40
FIGURA 13 (a) – GRÁFICO VÍDEO “ICE”, SSIM POR VARIAÇÃO DE UE	41
FIGURA 14 (a) – GRÁFICO VÍDEO “CREW”, SSIM POR VARIAÇÃO DE UE	43
FIGURA 15 (a) – GRÁFICO VÍDEO “CITY”, SSIM POR VARIAÇÃO DE UE...	44

LISTA DE TABELAS

TABELA 1 - HSPA x LTE	24
TABELA 2 – QOSE CLASS INDICATOR (QCI)	26
TABELA 3 - CORRELAÇÃO PSNR, MOS E SSIM	32
TABELA 4 - ATRIBUTOS DOS VÍDEOS A SEREM SIMULADOS.....	36
TABELA 5 - PARAMETROS FIXOS DA SIMULAÇÃO.....	37
TABELA 6 - VÍDEO "SOCCER", VALORES SSIM.....	39
TABELA 7 - VÍDEO "SOCCER", VALORES PSNR.....	39
TABELA 8 - VÍDEO "ICE", VALORES SSIM	40
TABELA 9 - VÍDEO "ICE", VALORES PSNR.....	40
TABELA 10 - VÍDEO "CREW", VALORES SSIM	42
TABELA 11 - VÍDEO "CREW", VALORES PSNR.....	42
TABELA 12 - VÍDEO "CITY", VALORES SSIM.....	44
TABELA 13 - VÍDEO "CITY", VALORES PSNR.....	44

SUMÁRIO

1	INTRODUÇÃO.....	14
1.1	JUSTIFICATIVA.....	14
1.2	OBJETIVO.....	14
1.2.1	Objetivo Geral.....	14
1.2.2	Objetivos Específicos	15
1.3	METODOLOGIA.....	15
1.4	ESTRUTURA.....	17
2	FUNDAMENTAÇÃO TEÓRICA.....	18
2.1	CODIFICAÇÃO DE VÍDEO.....	18
2.1.1	YUV.....	19
2.1.2	MPEG-4.....	21
2.2	LTE – LONG TERM EVOLUTION	23
2.2.1	Alocação de Recursos.....	24
2.2.2	Qualidade de serviço.....	25
2.2.3	Arquitetura de rede.....	26
2.3	STREAMING DE VÍDEO	28
2.4	AVALIAÇÃO DE QUALIDADE DE VÍDEO.....	28
2.4.1	MOS	30
2.4.2	PSNR	30
2.4.3	SSIM.....	31
3	SIMULAÇÃO	33
3.1	SOFTWARE DE SIMULAÇÃO	33
3.1.1	NS-3	33
3.1.2	QoE Monitor	33
3.2	topologia da rede.....	34
3.3	CENÁRIO DE SIMULAÇÃO	35
3.3.1	Experimentos.....	36
3.4	PARÂMETROS.....	37
4	RESULTADOS.....	38
4.1	INTERVALO DE CONFIANÇA	38

4.2	RESULTADOS DAS SIMULAÇÕES.....	38
4.3	ANÁLISE	45
5	CONCLUSÃO	47
6	REFERÊNCIAS.....	48
	APÊNDICE 1 – CÓDIGO PARA SIMULAÇÃO.....	51

LISTA DE ABREVIATURAS E/OU SIGLAS

3GPP - *3rd Generation Partnership Project*

4G – *4th Generation*

ATSV - *Advanced Television System Committee*

AVC - *Advanced Video Coding*

DL –*Downlink*

DVB-T - *Digital Video Broadcasting - Terrestrial*

ENB – *Evolved NodeB*

EPC - *Evolved Packet Core*

FTP - *File Transfer Protocol*

HD – *High Definition*

HSPA - *High Speed Packet Access*

IC – *Intervalo de Confiança*

IP – *Internet Protocol*

ISDB-T - *Integrated Services Digital Broadcasting - Terrestrial*

ITU - *International Telecommunication Union*

LTE – *Long Term Evolution*

MIMO - *Multiple-Input Multiple-Output*

MOS – *Mean Opinion Score*

MPEG - *Moving Picture Experts Group*

MSE – *Mean Square Error*

NIST - *National Institute of Standards and Technology*

NS3 – *Network Simulator 3*

OFDM - *Orthogonal frequency-division multiplexing*

OFDMA - *Orthogonal frequency-division multiplexing Access*

PAPR - *Peak to Average Power Ratio*

PGW – *Packet Data Network Gateway*

PSNR – *Peak Signal Noise Ratio*

QAM - *Quadrature Amplitude Modulation*

QoE – *Quality of Experience*

QoS – *Quality of Service*

QPSK - *Quadrature Phase Shift Keying*

RAN – *Radio Access Network*

RB – *Resource Block*

SGW – *Serving Gateway*

SBTVD – *Sistema Brasileiro de Televisão Digital*

SC-FDMA – *Single Carrier Frequency Division Multiple Access*

SSIM – *Structural Similarity*

TCP - *Transmission Control Protocol*

TTI - *Transmission Time Interval*

UDP - *UserDatagramProtocol*

UE – *UserEquipament*

UL - *Uplink*

1 INTRODUÇÃO

1.1 JUSTIFICATIVA

Segundo estudo realizado pela Cisco (2014), intitulado *Visual Networking Index*, o total de tráfego dedicado a vídeo está em franca expansão. Em 2018 a cada um segundo serão transmitidos o equivalente a um milhão de minutos em vídeos. Segundo o mesmo estudo, em 2013 66% de todo tráfego é decorrente de streaming de vídeos, e em 2018 este valor ficará próximo a 79%. Haverá também uma grande mudança na resolução dos vídeos transmitidos, como o Ultra HD (4K) com incremento de 0,1% em 2013 para 11% em 2018 do total do tráfego.

Conforme estudo divulgado pela 4G Américas (2014), *4G Mobile Broadband Evolution*, o Brasil possui 66% de todos os usuários de redes LTE na América Latina, totalizando dois milhões de usuários, sendo que a taxa de crescimento anual deverá permanecer em 100% até 2018. A quantidade de usuários de rede LTE mundial mais que dobrou entre o primeiro trimestre de 2013 e o de 2014, saindo de 104 milhões e atingindo 245 milhões.

Com a grande expansão da rede LTE (4G) no mundo e Brasil, aliado a popularização de dispositivos móveis que possibilitam a visualização de *streaming* de vídeos, tem-se a necessidade de aperfeiçoar o processo de codificação para otimizar a transmissão de dados com o menor número de perdas.

1.2 OBJETIVO

1.2.1 Objetivo Geral

Realizar análise qualitativa da degradação da qualidade de streaming de vídeo codificado com H.264/AVC sobre rede LTE.

1.2.2 Objetivos Específicos

O trabalho proposto tem os seguintes objetivos específicos:

- a) Análise das alterações de parâmetros de compactação espacial/temporal na degradação da qualidade de vídeo percebida;
- b) Definição de uma topologia de rede típica e simulação de streaming de vídeos sob LTE;
- c) Avaliação da qualidade do vídeo usando a Relação Sinal Ruído de Pico (PSNR) e Índice de Similaridade Estrutural (SSIM);
- d) Avaliação dos intervalos de confiança dos resultados utilizando ferramentas estatísticas.

1.3 METODOLOGIA

Serão utilizadas simulações computacionais para realizar a análise proposta, pois com isso tem-se a facilidade para alterar parâmetros e realizar uma grande quantidade de testes, resultando assim numa análise mais próxima de uma situação real de uso em diferentes cenários. Serão testados vídeos comumente utilizados pela comunidade de pesquisa da área, com características diferentes entre si. Os vídeos utilizados são de domínio público, disponíveis na biblioteca de vídeos para teste da Universidade de Hanôver Gottfried Wilhelm Leibniz, na Alemanha, disponível em [ftp://ftp.tnt.uni-hannover.de/pub/svc/testsequences/](http://ftp.tnt.uni-hannover.de/pub/svc/testsequences/).

Será utilizado o software de simulação *Network Simulator 3* (NS-3), para sistemas de Internet, inclusive LTE, desenvolvido para fins de pesquisa e educacionais. Tal software possui um módulo, denominado QoE Monitor, para análise métricas objetivas de transmissão de vídeos (PSNR e SSIM).

Segundo Wang (2003) às métricas para análise da qualidade de vídeo podem ser classificadas em métodos objetivos e subjetivos. As métricas subjetivas são baseadas na percepção humana. Métricas objetivas são baseadas em modelos matemáticos. Neste trabalho serão utilizadas às métricas SSIM e PSNR.

Será utilizado o Ffmpeg, uma ferramenta de código aberto e domínio público para codificação dos vídeos.

As análises de resultados serão realizadas utilizando-se de ferramentas estatísticas como média, desvio padrão, correlação e confiança.

A Figura 1 apresenta como sequência de como seguirá as simulações:

1. Codificação do arquivo de vídeo original formato YUV para MPEG4;
2. Transmissão do vídeo MPEG4 via LTE com parâmetros diferentes para cada cenário de simulação;
3. Decodificação do vídeo MPEG4 recebido para YUV;
4. Análise métrica objetiva do arquivo YUV original com o arquivo YUV recebido, calculando PSNR e SSIM.

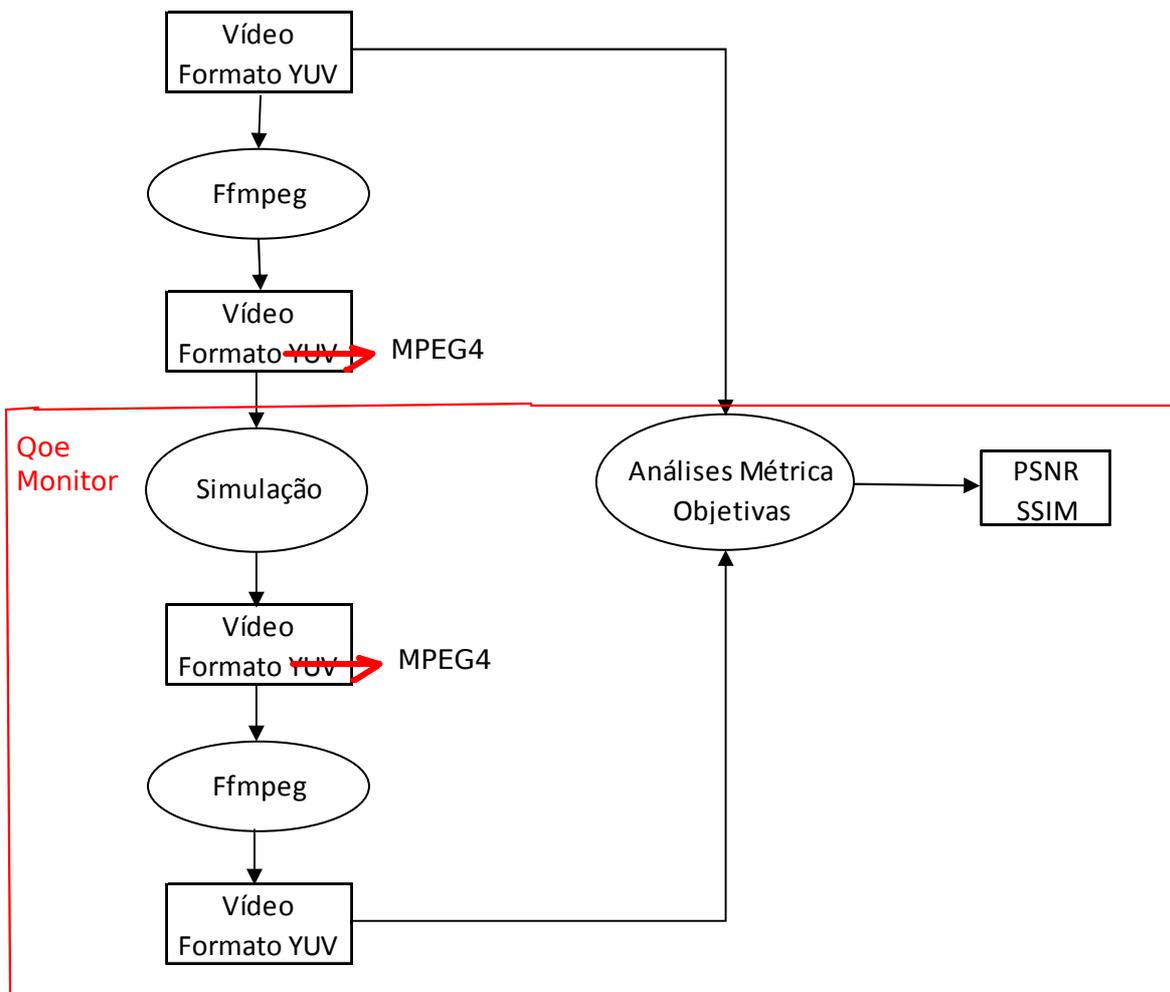


FIGURA 1 – FLUXO DE CODIFICAÇÃO, TRANSMISSÃO E ANÁLISE DO VÍDEO

1.4 ESTRUTURA

Este trabalho está dividido em cinco capítulos, sendo este primeiro o introdutório. O segundo capítulo, denominado Fundamentação Teórica, apresenta os conceitos necessários para compreensão de toda a metodologia utilizada para a execução deste projeto.

O capítulo três, demonstra o método de simulação executado, primeiramente explicando as ferramentas utilizadas, como o *Network Simulator* – NS3 e seu módulo QoE Monitor, e posteriormente todos os cenários aos quais foram submetidas as transmissões de vídeos para análise.

O quarto capítulo, denominado resultado, apresenta os resultados obtidos em todas as simulações e análise de resultados. Por último, no capítulo cinco, tem-se a conclusão, onde é abordado o cumprimento de todos os objetivos, com base nos resultados apresentados.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 CODIFICAÇÃO DE VÍDEO

Vídeos digitais possuem uma grande demanda que está em franca expansão devido a implementação do Sistema Brasileiro de Televisão Digital (SBTVD), videoconferências, streaming de vídeos na Internet, aumento na acessibilidade a dispositivos de gravação de vídeos e compartilhamentos em redes sociais. Há uma transformação na forma de comunicação digital informal, passando de textos para áudios e vídeos.

Sendo os vídeos compostos por uma sequência de imagens, muitas vezes com uma alta resolução, é necessária uma considerável quantidade de bits para representar tal vídeo, tornando sua transmissão um desafio. Para tanto foram elaboradas técnicas de compressão de vídeo, com o intuito de reduzir a quantidade de bits necessária para a representação do vídeo.

Existem dois tipos de compactação de vídeos, as sem perdas e as com perdas. No primeiro caso o vídeo reconstituído após a codificação é idêntico ao vídeo original, entretanto é possível reduzir somente em duas a três vezes a quantidade de bits para sua representação.

Segundo F. Yamada (2004) na compactação com perdas, como o MPEG-4, há uma distorção entre o vídeo original e o reconstituído, porém em sua grande maioria tal distorção não é perceptível ao olho humano. Neste tipo de codificação é possível reduzir significativamente a quantidade de bits para representação do vídeo em até cem vezes.

Ainda conforme afirma F. Yamada (2004) um sinal de vídeo possui uma taxa de bits que pode superar 1 Gbps, impossibilitando a transmissão em sistemas de TV digitais, como ATSV, DVB-T e ISDB-T, que transmitem uma taxa máxima de 20 Mbps, necessitam assim a compressão do vídeo para a transmissão. Atualmente estes padrões de transmissão digital utilizam o MPEG2 para a compressão de vídeo.

Segundo F. Yamada (2004) em 1986 a ISO (*International Standard Organization*) organizou um grupo, denominado JPEG (*Joint Photograph*

Expert Group), para desenvolver e padronizar métodos de compressão de sinal de imagens estáticas.

Em 1988 institutos desenvolveram e padronizaram sistemas de compactação de vídeo, baseados no JPEG. Na Figura 2 está representado a cronologia da elaboração destes padrões realizado pela ITU-T (*International Telecommunication Union - Telecommunication Standardization Sector*) e MPEG (Moving Picture Experts Group). O objeto de estudo deste trabalho está focado no padrão H264/MPEG-4 AVC.

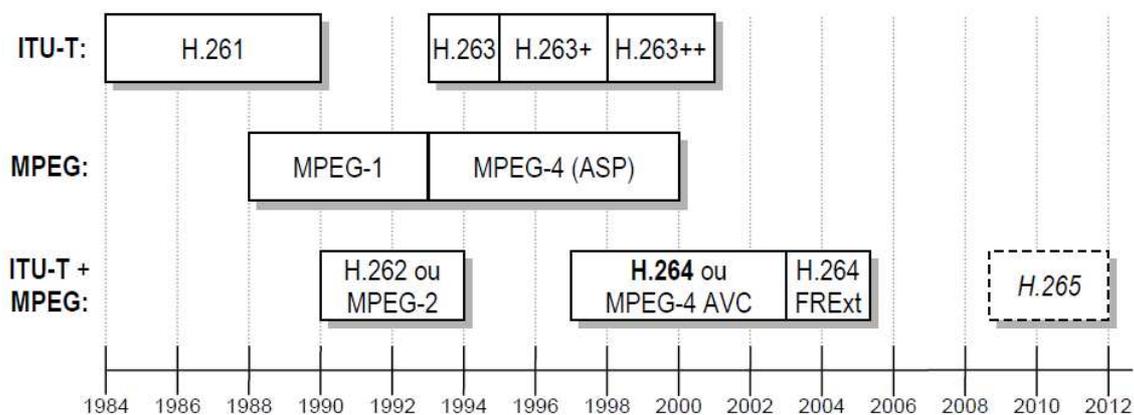


FIGURA 2 - CRONOLOGIA DA EVOLUÇÃO DE COMPACTAÇÃO DE VÍDEO
 FONTE: JENQ-NENG HWANG, (2009 p.108)

2.1.1 YUV

O padrão YUV é derivado do padrão RGB, sendo as três letras as iniciais das cores primárias, *Red* (Vermelho), *Green* (Verde) e *Blue* (Azul). Com as junções destas três cores com intensidades diferentes é possível resultar em diversas cores diferentes. Utilizando um byte para representar cada cor é possível obter mais de 16 milhões de combinações de tons, saturação e brilho.

O padrão YUV, conhecido também como CCIR 601 ou YCbCr é um padrão utilizado onde pode ser representada luminância e crominância. Onde Y representa a luminosidade (luminância), U ou Cb representa a informação da cor azul e V ou Cr representa a informação da cor vermelha.

Este padrão foi desenvolvido para transmissão de TV analógica possibilitando que o mesmo sinal seja utilizado tanto para televisores a cores e televisores preto e branco, pois a informação de luminância (Y) já é suficiente para imagens em tons de cinza.

Os valores da luminância Y e das componentes U e V podem ser correlacionados com os valores das componentes RGB, sendo suas definições e equações recomendadas pelo (ITU-R BT.601-5) para imagens em baixa resolução e pelo (ITU-R BT.709-5) para imagens em alta resolução.

O padrão YUV tem uma nomenclatura para a representação ou não da subamostragem dos sinais de *croma*, Cb e Cr. No formato 4:4:4 para cada amostra de Y há uma amostra de Cb e Cr. No formato 4:2:2 a amostragem dos sinais de croma são reduzidos pela metade horizontalmente. No formato 4:2:0 a amostragem é reduzida pela metade horizontalmente e verticalmente, resultando em quatro amostragens de Y para cada amostra de Cb e Cr.

Segundo Manoel (2007) para a representação correta da imagem procedida de uma subamostragem dos sinais de *croma*, estas devem ser interpolados corretamente com os sinais de *luma*, conforme Figura 3.

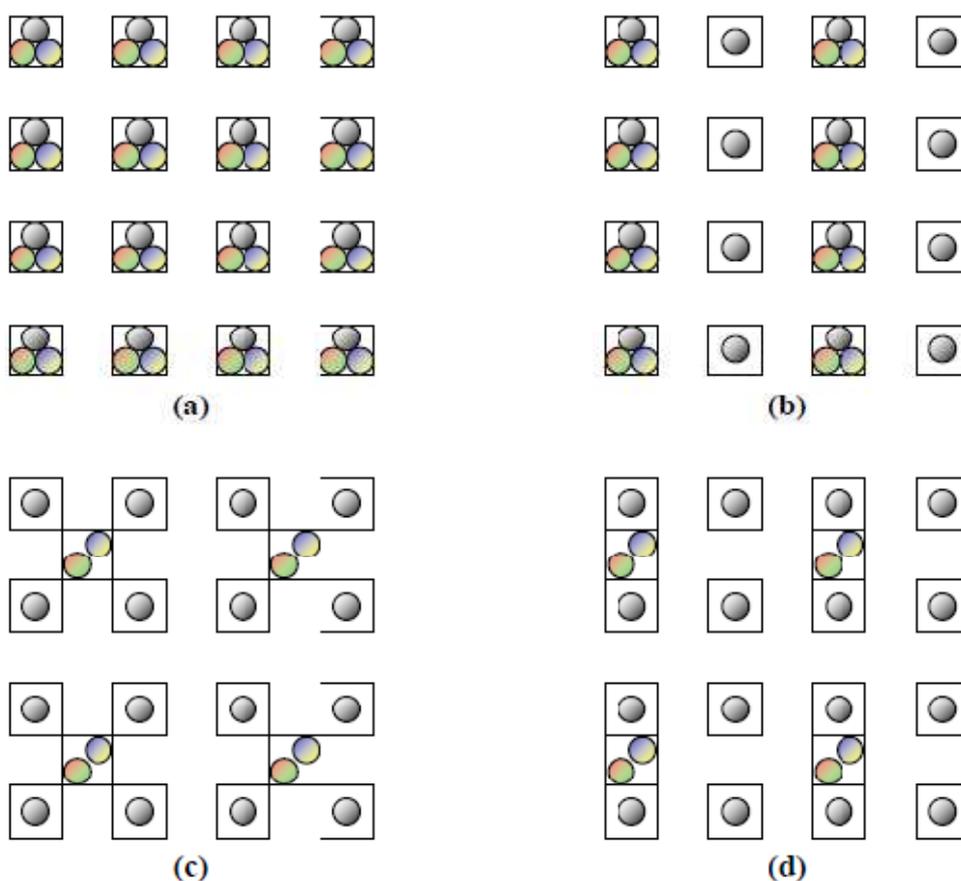


FIGURA 3 - POSIÇÃO DAS AMOSTRAS DE LUMINÂNCIA E CROMINÂNCIA NOS FORMATOS: (A) 4:4:4; (B) 4:2:2; (C) 4:2:0 MPEG1; (D) 4:2:0 MPEG-4
 FONTE: MANOEL, E (2007), p. 13.

2.1.2 MPEG-4

Como explicado na seção 2.1.1 deste documento, um padrão de codificação de vídeo tem como objetivo reduzir a quantidade de informação necessária para representar um determinado vídeo. O padrão MPEG-4 é uma evolução dos padrões MPEG-1 e MPEG-2.

A codificação MPEG-4 é uma das mais utilizadas atualmente, porém esta sofreu várias evoluções. O tema de estudo deste trabalho será baseado na codificação MPEG-4 Parte 10, conhecida também como MPEG-4 AVC (*AdvancedVideoCoding*) ou H.264.

Segundo artigo da Axis (2008) o H.264 é utilizado na codificação de Blue-Ray, serviços de IPTV (BBC *i player*, Sri Lanka Telecom, etc.) e também pelo Sistema Brasileiro de Televisão Digital (ISDB-Tb) padrão de teledifusão digital também utilizado no Peru, Argentina, Equador e outros países da América Latina. É utilizado também na maioria dos sistemas de transmissão de TV sobre o protocolo IP (IPTV).

Segundo Manoel (2007) a melhoria de desempenho do MPEG-4 em relação ao MPEG-2 é considerável, tendo uma taxa de bits 50% menor, mantendo a qualidade semelhante.

Existem dois tipos de redundância nos vídeos, as quais podem ser utilizadas para compactação. São elas: espacial e temporal. A espacial leva em consideração informações similares no mesmo quadro, como por exemplo um fundo preto. Já a redundância temporal é baseada em mais de um quadro, levando em consideração as semelhanças numa sequência de quadros, como uma paisagem de fundo.

Para explorar a redundância temporal, o MPEG utiliza três tipos de frames, denominados I, B e P.

Os quadros do tipo I (*IntraCoded Frame*) contém o maior número de informações e não depende dos demais para serem decodificados. É o quadro que inicia a sequência de frames, denominada GoP (*Group of Pictures*).

Quadros do tipo P (*PredictiveCoded Pictures*), são codificados a partir de quadros I ou quadros P, ou seja, são dependentes do quadro anterior. É observada a redundância temporal, baseado no movimento, para compactação.

Quadros do tipo B (*BipredictiveCoded Picture*), são os quadros que possuem a menor quantidade de informação, codificados a partir de quadros I e P anteriores e posteriores, e utilizam também a compensação do movimento para compactação.

Os quadros do tipo I são os mais importantes devido ao seu alto grau de informação em relação aos demais tipos de frames.

Conforme a Figura 4 apresenta, o GOP inicia-se sempre em um frame I tendo em sequência frames P e/ou B. O tamanho do GOP é formado por dois parâmetros, M e N, sendo M a menor distância entre frames I ou P e N o número de frames B consecutivos.

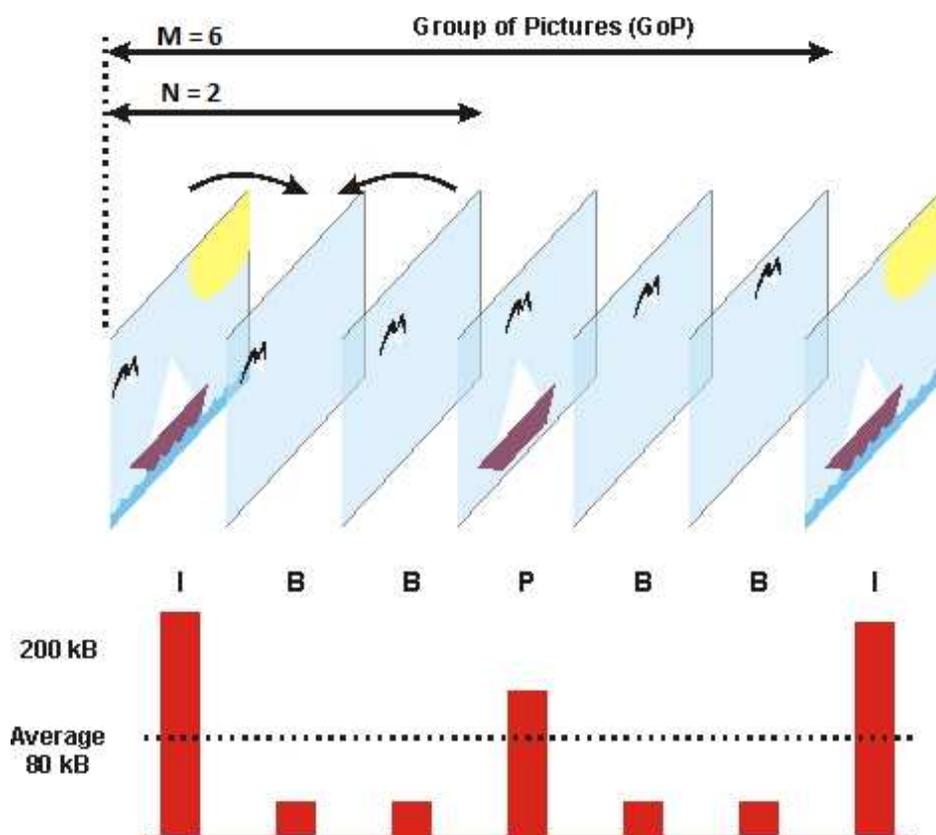


FIGURA 4 - DIAGRAMA DO GROUP OF PICTURES
 FONTE: PRICE, D. INTRODUCTION TO MPEG 2ND EDITION, (2012)

Neste presente trabalho os *Group of Pictures* serão configurados somente com frames I e P, ou seja, não serão utilizados frames B.

2.2 LTE – LONG TERM EVOLUTION

O *LongTermEvolution* (LTE) é atualmente a tecnologia utilizada para implementação da 4ª geração de comunicações móveis via rádio (4G), sendo introduzido na *Release 8 e 9* do 3º *GenerationPartnership Project* (3GPP).

O LTE usa a multiplexação por divisão ortogonal de frequência (OFDM) para downlink e acesso múltiplo por divisão de frequência de portadora única (SC-FDMA) como sua tecnologia de acesso via rádio, juntamente com avançadas tecnologias de antenas. Ao integrar tecnologias de ponta, como a entrada múltipla e saída múltipla (MIMO) e múltiplo acesso por divisão ortogonal de frequência (OFDMA), conforme afirma Zhou (2013), o LTE suporta uma taxa de pico de downlink de 300 Mbps, e uma taxa máxima de uplink de 75 Mbps, possibilitando um atraso menor que 5 ms na rede de acesso rádio (RAN) em condições ideais.

O OFDM divide o espectro em várias subportadoras, com largura de banda menor, de forma que seja permitido maior robustez à interferência. Assim, no LTE são utilizados subportadoras com banda de 15 kHz, permitindo um melhor uso do espectro.

A tecnologia de acesso LTE utiliza diferentes níveis de modulação QAM, podendo ser QPSK, 16-QAM ou 64-QAM, de acordo com as condições do canal.

Uma motivação para o *uplink* ter a multiplexação em SC-FDMA é devido o OFDM possuir grandes variações de potência, o que contribui para o aumento significativo do PAPR (*Peak to Average Power Ratio*). Como o UE é alimentado com bateria, é essencial que seu consumo de energia seja otimizado. Desta forma, a utilização do SC-FDMA contribui para o aproveitamento de potência dos equipamentos móveis. Contudo, o SC-FDMA se baseia no OFDM e também utiliza múltiplas portadoras espaçadas de 15 kHz.

Com a melhor capacidade de mobilidade dos terminais móveis em grandes velocidades de deslocamento, o LTE oferece uma boa oportunidade para os usuários utilizarem aplicações de intensa largura de banda, como streaming de vídeo e vídeo conferência.

Conforme Campana (2010) a tecnologia anterior ao LTE, denominada de *High-Speed Packet Access* (HSPA) definida no 3GPP Release 5, pode atingir até 14,4 Mbit/s de *downlink* em condições ideais e uplink de 5,7 Mbit/s. As vantagens do LTE com relação a tecnologia anterior são evidenciadas na Tabela 1, que apresenta uma comparação entre os requisitos do HSPA e LTE.

TABELA 1 - HSPA x LTE

	Downlink		Uplink	
	Release-6 HSPA	LTE	Release-6 HSPA	LTE
Pico de taxa de transmissão (Mbps)	14,4	144	5,7	57
Eficiência Espectral (bps/Hz)	0,75	1,84	0,26	0,67
Cell edge user throughput (bps/Hz)	0,006	0,0184	0,006	0,015

Fonte: JIMAA(2014)

Devido ao sistema LTE ser baseado em arquitetura da rede IP, caracterizado por diminuição de elementos no sistema, baixa latência na transmissão de dados, grande flexibilidade e redução no custo de operação, a implementação do 4G torna-se atraente e viável as operadoras.

2.2.1 Alocação de Recursos

A menor unidade de recurso que o pode ser alocada a um UserEquipment (UE) no OFDMA é chamado de *resourceblock* (RB), conforme mostrado na Figura 5. Especificamente, um RB no domínio do tempo é um intervalo de tempo de transmissão denominado *transmission time interval* (TTI), que tem a duração de dois intervalos de tempo e uma subportadora no domínio da frequência. Então, cada intervalo de tempo é 0,5 ms e uma subportadora é 180 kHz. A alocação de RBs é realizada pela eNodeB, que é o elemento de hardware central conectado com os UEs através da rede rádio.

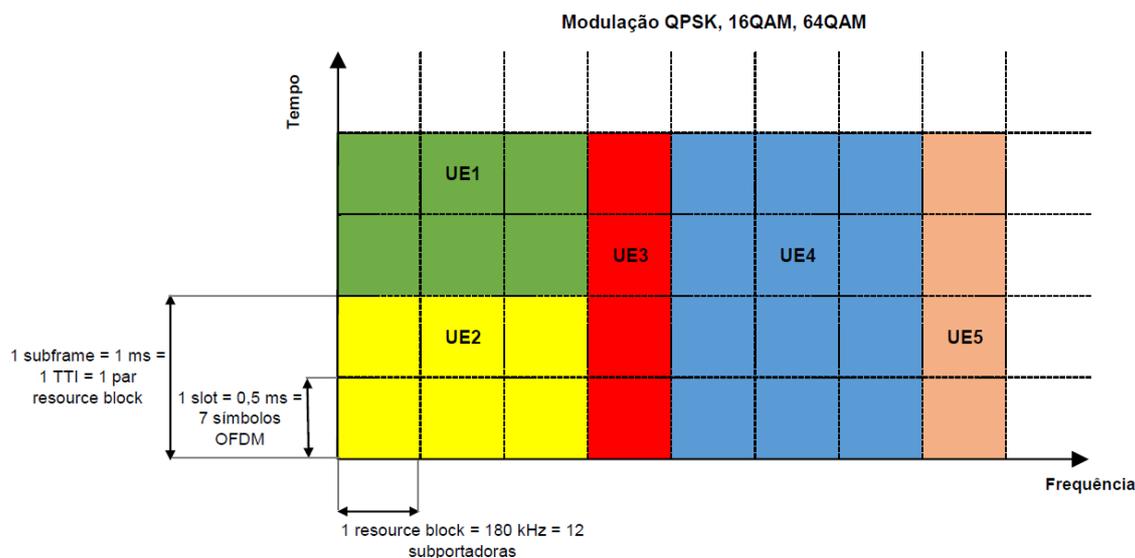


FIGURA 5 - BLOCO DE RECURSOS

Desta forma, é possível uma distribuição flexível de RB a diferentes UEs, tanto no domínio do tempo como no domínio da frequência. Isto agrega muitas vantagens em comparação às tecnologias anteriores, pois o compartilhamento nos domínios da frequência e do tempo tendem a tornar mais eficientes a distribuição de recursos.

2.2.2 Qualidade de serviço

Um aspecto importante de qualquer rede de pacotes é um mecanismo para garantir a diferenciação de fluxo de pacotes com base nos seus requisitos de QoS. Aplicativos como o streaming de vídeo ou de vídeo conferência possuem necessidades especiais de QoS e devem receber atendimento diferenciado através da rede.

No LTE, o QoS é estabelecido entre o *Packet Gateway* (PGW) e o usuário. Cada portadora do EPC é associada a um perfil de QoS, composta por uma portadora de rádio e um canal de acesso, a rede pode dar prioridade a pacotes conforme necessidade.

Ao receber um pacote IP da internet, o PGW executa a classificação dos pacotes com base em parâmetros, tais como as regras recebidas, e os envia através do canal de acesso adequado. Com base no canal de acesso, o eNodeB pode mapear os pacotes para a apropriada portadora de rádio.

Para cada portadora com um tipo de recurso *Guaranteed Bit Rate* (GBR), significa que a largura de banda da portadora será garantida. Quando o tipo do recurso é não GBR (NON-GBR) a largura de banda da portadora será com melhor esforço.

A Tabela 2 apresenta o *QoSClassIndicator* (QCI) que especifica o parâmetro correspondente e aplicações comuns.

TABELA 2 – QOS CLASS INDICATOR (QCI)

QCI	Tipo de recurso	Prioridade	Atraso do pacote	Taxa de perda de pacote	Aplicação
1	GBR	2	100ms	10^{-2}	Conversação de voz
2	GBR	4	150ms	10^{-3}	Conversação por vídeo
3	GBR	3	50ms	10^{-3}	Jogos de tempo real
4	GBR	5	300ms	10^{-6}	<i>Streaming</i> de vídeo
5	NON-GBR	1	100ms	10^{-6}	IMS <i>Signaling</i>
6	NON-GBR	6	300ms	10^{-6}	Vídeo, www, email, ftp
7	NON-GBR	7	100ms	10^{-3}	Jogos interativos
8	NON-GBR	8	300ms	10^{-6}	Vídeo, www, email, ftp
9	NON-GBR	9	300ms	10^{-6}	Vídeo, www, email, ftp

Apesar do LTE ter disponível recurso para aplicar qualidade de serviço no tráfego de dados, não há obrigatoriedade na legislação vigente para que as operadoras de acesso móvel utilizem tal recurso em transmissões de vídeo. Desta forma, este trabalho não contempla o QoS nas simulações realizadas.

2.2.3 Arquitetura de rede

Os conceitos do LTE foram projetados para suportar o uso de serviços baseados numa arquitetura de rede IP. A Figura 6 apresenta este modelo.

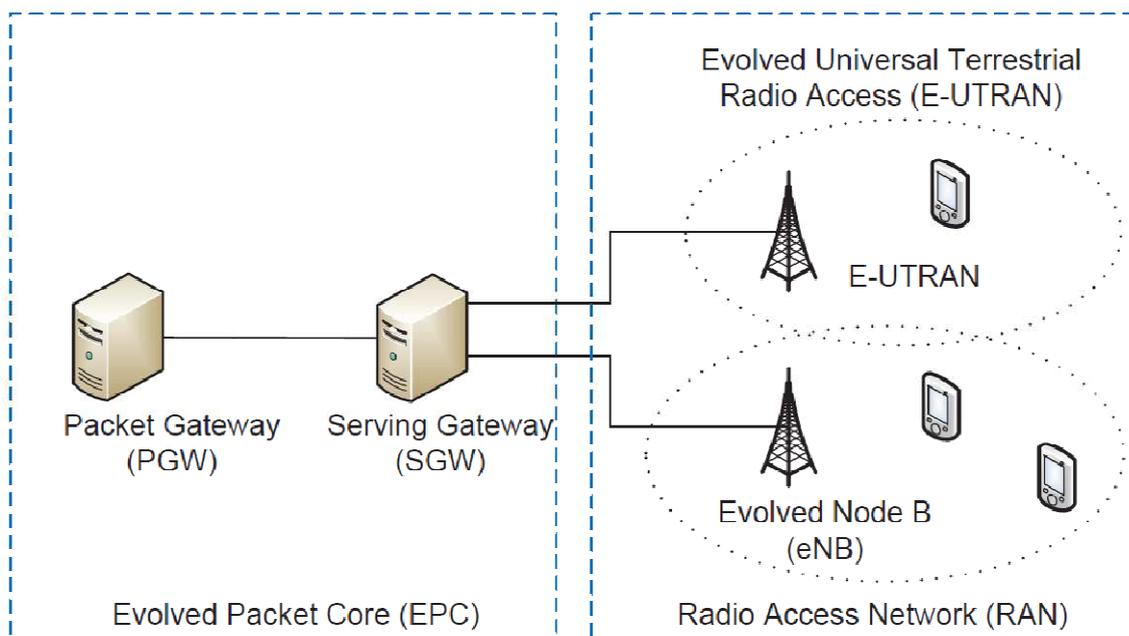


FIGURA 6 - ARQUITETURA DE REDE LTE
 FONTE: NOSENSON (2009)

Segundo Nossenson (2009), a rede LTE consiste basicamente na divisão de duas partes principais, o *Evolved Packet Core* (EPC) e *Radio Access Network* (RAN). O EPC é o principal responsável pela gestão da mobilidade, estabelecendo conexão da RAN com a internet pública. Dentro do EPC, estão o *Serving Gateway* (SGW) e *Packet Gateway* (PGW). Esta arquitetura plana com menos nós envolvidos, reduz latências e melhora o desempenho.

A função de alto nível de SGW é a gestão do tunelamento e de comutação. O SGW faz parte da infraestrutura de rede mantida nas instalações centrais de operação, tendo um papel muito menor nas funções de controle.

O PGW é o roteador de borda entre o EPC e rede externa. Também atua como ponto de ligação de IP para o UE, executando a gestão do tráfego, conforme exigido pelos serviços em questão.

No RAN se encontra o *evolved universal terrestrial radio access* (E-UTRA) como a interface aérea que fornece transmissão sem fio para os UEs.

Conforme cita Zhou (2013), um elemento chave do LTE que afeta o desempenho do sistema é o escalonador de recursos na camada de controle de acesso ao meio (MAC), destinado a alocar recursos de rádio para UEs de modo que vários requerimentos de QoS de downlink e uplink sejam satisfeitos. Este elemento está localizado na eNodeB.

2.3 STREAMING DE VÍDEO

Streaming de vídeo ou outro tipo de multimídia, se configura como a transmissão deste tipo de conteúdo, dividindo-os em pacotes, enviando de forma contínua para reprodução ao usuário. Normalmente essa transmissão é *multicast*, ou seja, a transmissão é feita por um servidor para diversos usuários simultaneamente.

Devido a popularização de serviços de Internet de alta velocidade, o número de acessos para reprodução de mídia via streaming tem crescido consideravelmente, conforme apresentado na seção 1.1, este tipo de tráfego já ultrapassa 60% de todo tráfego global e continua em franca expansão. Dentro os principais fornecedores deste tipo de serviço temos o Youtube e o Netflix.

Segundo Greengrass (2009) durante o processo de transmissão pode ocorrer perdas de pacotes, devido a características da rede, dentre elas a latência dos pacotes maiores que o tempo máximo do buffer de reprodução. Dentre os principais defeitos que afetam o QoE tem-se: mosaico (*blocking*), pixelização, fantasmas e congelamentos de quadros conforme apresenta Greengrass (2009).

2.4 AVALIAÇÃO DE QUALIDADE DE VÍDEO

A avaliação da qualidade de vídeo permite mensurar por diferentes técnicas objetivas ou subjetivas o nível de satisfação do espectador no que tange a qualidade de reprodução do vídeo.

Conforme relata Cerqueira (2009) as métricas para QoE podem ser divididas em subjetiva e objetiva, podendo também ser divididas em métricas diretas e indiretas.

As métricas subjetivas são baseadas em testes reais de exibição de sequência de vídeos a espectadores, onde estes avaliam a qualidade dos vídeos em questões. Os ambientes de testes e espectadores devem ser controlados, para tanto a ITU (*International Telecommunication Union*) faz recomendações em ITU-T P.910 (2008) e ITU-R BT.500 (2012).

Em Cerqueira (2009) define métricas objetivas como um modelo desenvolvida para estimar a QoE através de modelamentos matemáticos.

Modelos objetivos têm como vantagem de serem rápidos de serem produzidos, podendo assim fazê-los em grande escala.

Conforme recomendação da ITU-TJ.143 (2000) existem três abordagens para métricas objetivas:

Referência completa (*FullReference FR*): Nesta abordagem estão presentes o vídeo original e o vídeo recebido, a comparação e cálculo da qualidade é feita entre estas duas fontes. A Figura 7 representa um diagrama de funcionamento da métrica de referência completa.

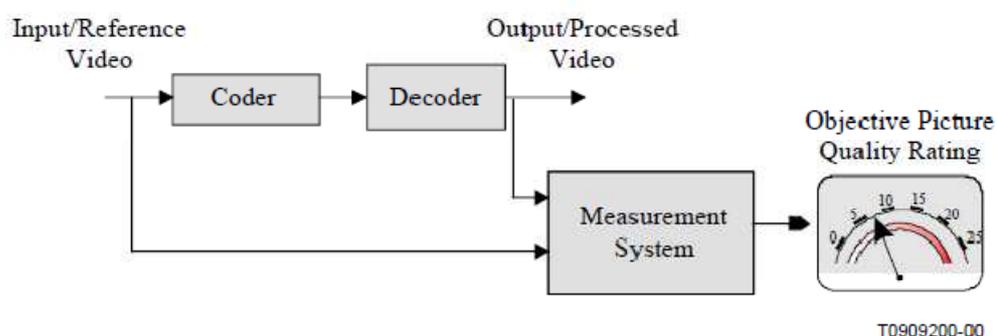


FIGURA 7 - REFERÊNCIA COMPLETA
FONTE: RECOMENDAÇÃO ITU-T J.143 (2000)

Referência reduzida (*ReducedReference RR*): Neste método é enviado junto com o vídeo uma quantidade limitada de dados em paralelo, sendo estes dados utilizadas para cálculo da experiência do usuário. A Figura 8 representa um diagrama de funcionamento da métrica de referência reduzida.

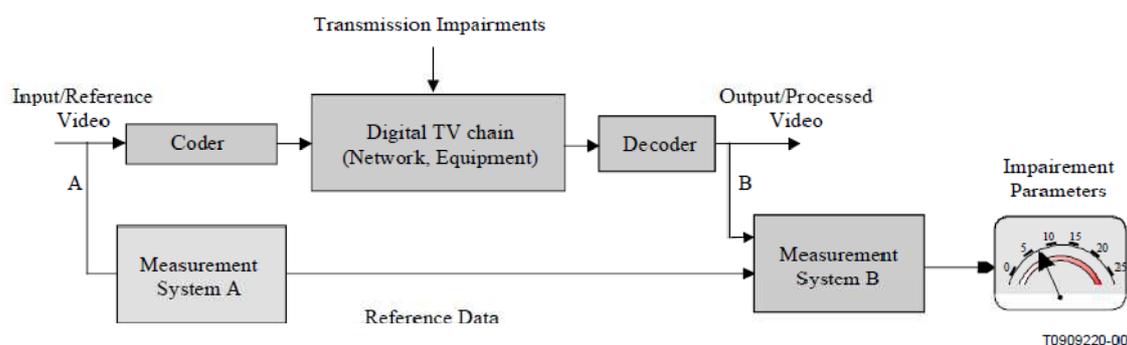


FIGURA 8 - REFERÊNCIA REDUZIDA
FONTE: RECOMENDAÇÃO ITU-T J.143 (2000)

Sem referência (*No Reference NR*): A avaliação é realizada através do vídeo recebido, não tendo uma referência do vídeo original.

Segundo Cerqueira (2009) métricas diretas levam em consideração fatores que afetam diretamente a percepção do usuário, tal como o áudio ou vídeo recebido. Estas são obtidas através de perdas de pacotes, desempenho da rede de transmissão, tipos de codificação, etc. As principais métricas diretas são: *PeakSignaltoNoiseRatio* (PSNR), *StructuralSimilarity* (SSIM), *VideoQualityMetric* (VQM) e *MeanOpinion Score* (MOS).

Métricas indiretas são baseadas na experiência do usuário como um todo, porém não está relacionada necessariamente ao vídeo, afetando principalmente a experiência do usuário sobre o serviço. Pode-se citar como um exemplo de influência nas métricas indiretas o tempo de resposta entre o usuário clicar no vídeo até sua reprodução.

2.4.1 MOS

O MOS (*MeanOpinion Score*) é um método subjetivo para cálculo do QoE, ele consiste em obter um valor para a qualidade de vídeo através das opiniões dos telespectadores.

Para obtenção de um resultado mais fidedigno é necessário que todas as variáveis dos testes sejam controladas, incluindo desde o ambiente de exibição, a tela da projeção, distância e ângulo para a tela, vídeos a serem testados e inclusive os próprios espectadores. Tais recomendações estão descritas em ITU-R BT.500-13 (2012).

Os espectadores avaliam os vídeos atribuindo notas de 1 a 5, onde o maior valor representa uma melhor qualidade

2.4.2 PSNR

PSNR (*PeakSignaltoNoiseRatio*), Relação Sinal Ruído de Pico, é definido como a relação entre a máxima energia de um sinal e seu ruído, para a análise da qualidade de vídeo é utilizando como referência vídeo original e vídeo degradado. Esta é uma técnica objetiva, com referência completa e direta. A unidade para sua representação é o dB (decibel).

O valor de PSNR é baseado no erro quadrático médio (*Mean Square Error*), MSE, definido como:

$$\text{MSE} = \frac{1}{lc} \sum_{i=1}^l \sum_{j=1}^c [Y_{0(i,j)} - Y_{r(i,j)}]^2 \quad (2.1)$$

onde l e c representam o número de linhas e colunas do quadro, $Y_{0(i,j)}$ representa o valor de luminância de um pixel específico do vídeo original e $Y_{r(i,j)}$ a luminância do vídeo recebido. O mesmo procedimento pode ser realizado com os valores do croma Cb e do croma Cr

O PSNR pode ser obtido através:

$$\text{PSNR} = 20 \log_{10} \left(\frac{\text{MAX}}{\text{MSE}} \right) \quad (2.2)$$

onde MAX é o valor máximo do pixel, baseado no número de bits para a sua representação.

Quanto maior o valor do PSNR menor a degradação do vídeo, este pode variar de zero a infinito, porém normalmente se limita um valor máximo para o cálculo, neste presente trabalho o limite do valor máximo do PSNR foi definido em 99 dB.

2.4.3 SSIM

SSIM (*Structural similarity*), índice de similaridade estrutural, conforme afirmam Wang (2004) e Cantarutti (2013) é uma técnica métrica objetiva, com referência completa e direta que leva em consideração a medição da distorção estrutural da imagem, diferente do PSNR que leva em conta a medição do erro. O método leva em conta para a medição da degradação a correlação de perdas, distorção de luminância e distorção de contraste.

$$\text{SSIM}_{(x,y)} = \frac{(2\mu_x\mu_y+c_1)(2\sigma_{xy}+c_2)}{(\mu_x^2+\mu_y^2+c_1)(\sigma_x^2+\sigma_y^2+c_2)} \quad (2.2)$$

onde a média de x e y é representado por μ_x e μ_y , σ_x^2 e σ_y^2 representa a variância e σ_{xy} a covariância entre x e y . As variáveis c_1 e c_2 são utilizadas para estabilizar a divisão para valores baixos, são baseadas nos valores do número de bits para representar um pixel, conforme estabelece Wang (2004).

Sua concepção é baseada na ideia que os pixels possuem forte interdependência, principalmente quando estão extremamente próximos. O valor de SSIM pode variar de 0 a 1, sendo 1 onde a imagem recebida é exatamente igual a imagem original, ou seja, não há degradação.

Na Tabela 3 está representada a correlação entre MOS, PSNR e SSIM facilitando assim a análise dos resultados em correlação aos diferentes tipos de métricas

TABELA 3 - CORRELAÇÃO PSNR, MOS E SSIM

Qualidade	MOS	PSNR (dB)	SSIM
Excelente	5	>37	>0,99
Bom	4	31 - 37	0,95 – 0,99
Razoável	3	25 - 31	0,88 – 0,95
Pobre	2	20 - 25	0,5 – 0,88
Péssimo	1	<20	<0,5

FONTE: ZINNER, A et al. (2010) Modificado pelos autores (2014)

3 SIMULAÇÃO

3.1 SOFTWARE DE SIMULAÇÃO

3.1.1 NS-3

O *software* de simulação utilizado neste trabalho foi o *Network Simulator* versão 3, também conhecido por ns-3. Este simulador de redes é baseado em eventos discretos e foi desenvolvido especialmente para pesquisa e uso educacional. O ns-3 é um *software* livre, sob a licença GNU GPLv2 e está disponível ao público.

O ns-3 permite o desenvolvimento pela comunidade de modelos de simulação de protocolos ou tecnologias de acesso, sendo assim possível construir qualquer cenário que se deseja estudar e simular de forma muito aproximado da realidade o comportamento, medições e aferimentos de protocolos ou tecnologias de redes.

3.1.2 QoE Monitor

Conforme afirma Saladino (2012) QoE Monitor é um módulo para o *software* de simulação de redes *Network Simulator 3*(NS3) capaz de calcular a qualidade de experiência para streaming de vídeos através de técnicas objetivas, sendo elas PSNR (*PeakSignal-to-NoiseRatio*) e SSIM (*StructuralSimilarity*).

Este módulo é um *software* aberto, elaborado em C++, desenvolvido por Daniela Saladino, Alessandro Paganelli e Maurizio Casoni da Universidade de Módena e Reggio Emília na Itália.

A ferramenta suporta diversos formatos de vídeos, tal como, AVI, YUV, MP4, WMV e VOB. Independentemente do formato de entrada é criado pela ferramenta um arquivo YUV original e um arquivo YUV recebido, este último baseado no vídeo recebido no mesmo formato do enviado. Após o envio total é realizada o cálculo de PSNR e SSIM.

Os resultados de PSNR e SSIM ao final da transmissão são representados em planilhas eletrônicas. São calculados valores de PSNR para as componentes Y, U e V.

Todos os valores de SSIM e PSNR são calculados e apresentados quadro a quadro. Como resultado também são apresentados os valores de *jitter* tempos de envio e recebimento de cada pacote.

No artigo apresentado por Saladino (2012), é realizada uma comparação com os resultados do QoE monitor com a ferramenta Evalvid, sendo esta amplamente utilizada pela comunidade para análise da qualidade de experiência. Tal comparação resultou em um erro máximo de 0,002 dB para PSNR e 0,02 para SSIM.

A FIGURA 9 representa um fluxo geral do funcionamento da ferramenta.

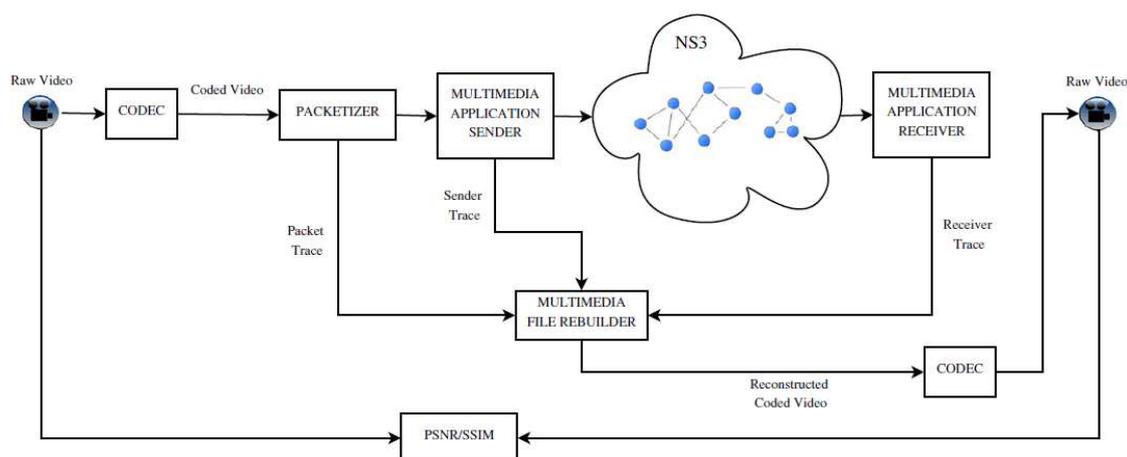


FIGURA 9 - FLUXO GERAL QOE MONITOR
 FONTE: SALADINO, D. (2012)

3.2 TOPOLOGIA DA REDE

A FIGURA10 representa o diagrama de rede dos cenários de simulação. Nesta topologia adotada, o *streaming* de vídeo que será enviado pelo elemento VideoServer para um único UE concorrerá o tráfego com diversos outros *userequipments* que estarão recebendo dados em uma transferência de arquivo utilizando o protocolo TCP.

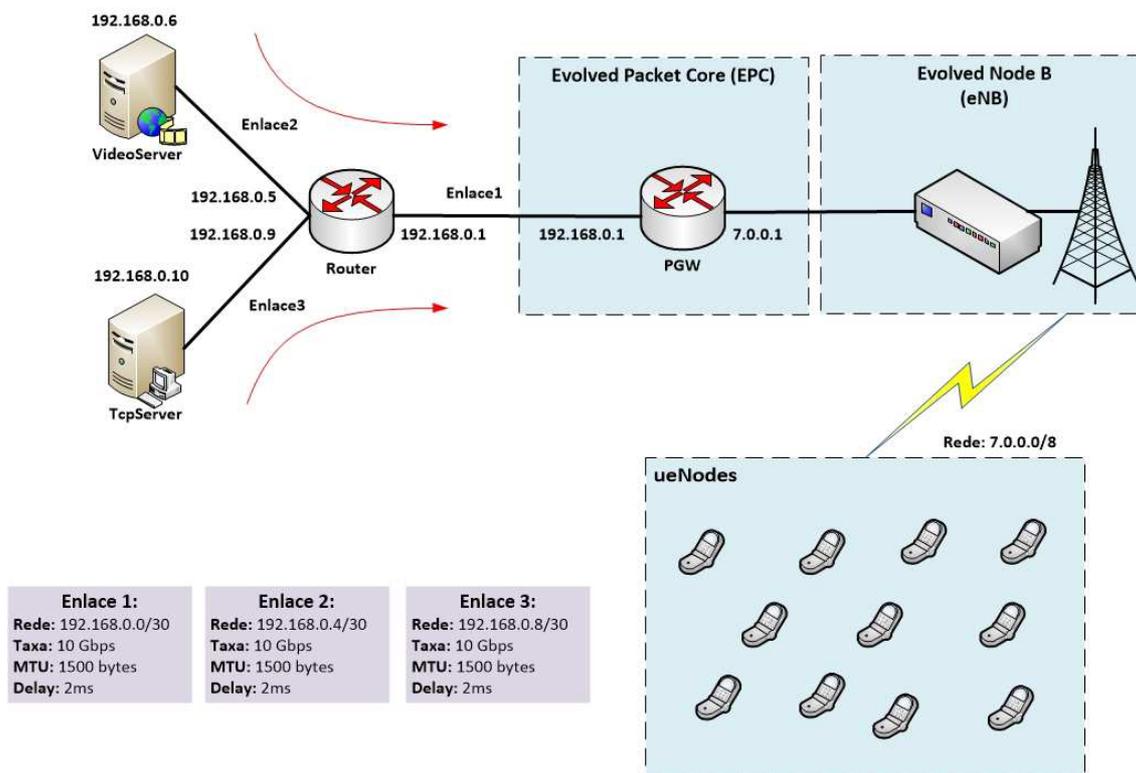


FIGURA10 – TOPOLOGIA DA REDE

3.3 CENÁRIO DE SIMULAÇÃO

O cenário de simulação proposto consiste na alteração dos valores de tamanho do GoP dos vídeos a serem transmitidos e número UEs competindo como tráfego de fundo.

O tamanho do GoP utilizado variou entre 15, 30, 45 e 90. Devido a uma limitação do módulo QoE monitor não serão utilizados frames B.

Para a codificação do vídeo do formato YUV para MPEG-4, será utilizada a seguinte linha de comando da ferramenta ffmpeg:

O número de UEs utilizados para gerar o tráfego de fundo será de 10, 20, 30 e 40, de forma a permitir a avaliação dos efeitos do escalonamento de recursos na degradação da qualidade de vídeo.

Os vídeos utilizados são de domínio público, disponíveis na biblioteca de vídeos para teste da Universidade de Hanôver Gottfried Wilhelm Leibniz, na Alemanha, disponível em: <ftp://ftp.tnt.uni-hannover.de/pub/svc/testsequences/>.

3.3.1 Experimentos

Os experimentos consistem na transmissão dos vídeos representados na Figura 11 com os atributos representados na Tabela 4. Serão realizadas cinco rodadas de simulações por variação de atributo de GoP e UE concorrentes, totalizando oitenta simulações para cada vídeo.

TABELA 4 - ATRIBUTOS DOS VÍDEOS A SEREM SIMULADOS

Descrição	Atributo			
	Soccer	Ice	Crew	City
Vídeo	Soccer	Ice	Crew	City
Resolução	4CIF (704X576)	4CIF (704X576)	4CIF (704X576)	4CIF (704X576)
Frames	300 Frames	240 Frames	300 Frames	300 Frames
Cadência	30 fps	30 fps	30 fps	30 fps
Característica do Vídeo	Vídeo com moderado grau de movimento	Vídeo com pouca movimentação de câmera	Vídeo com grande movimentação de fundo e de câmera	Vídeo com movimentação de câmera moderada.
GoP	15; 30; 45; 90	15; 30; 45; 90	15; 30; 45; 90	15; 30; 45; 90
UE concorrentes	10; 20; 30; 40	10; 20; 30; 40; 60	10; 20; 30; 40	10; 20; 30; 40; 60
Bit Rate GoP 15	2868 kbps	1303 kbps	2980 kbps	2996 kbps
Bit Rate GoP 30	2621 kbps	1184 kbps	2801 kbps	2594 kbps
Bit Rate GoP 45	2526 kbps	1152 kbps	2745 kbps	2473 kbps
Bit Rate GoP 90	2477 kbps	1107 kbps	2644 kbps	2351 kbps



(a)



(b)



(c)



(d)

FIGURA 11—(a) VIDEO SOCCER; (b) VÍDEO ICE; (c) VÍDEO CREW; (d) VÍDEO CITY
 FONTE: SERVIDOR UNIVERSIDADE HANNOVER (2014)

3.4 PARÂMETROS

Durante todas as simulações alguns parâmetros permaneceram fixos para garantir a homogeneidade dos resultados. Os parâmetros da rede LTE, transmissão e simulação são apresentados na Tabela 5.

TABELA 5 - PARAMETROS FIXOS DA SIMULAÇÃO

Parâmetro	Atributo
PARÂMETROS LTE	
Largura de Banda para Downlink	20 MHz
DL Earfcn	2500 MHz
UL Earfcn	2500 MHz
Potência de Transmissão EU	24 dBm
Potência de Transmissão Enb	32 dBm
PARÂMETROS DE SIMULAÇÃO E TRANSMISSÃO	
Jitter Buffer	200 ms
Unidade Máxima de Transmissão (MTU)	1400 bytes
Duração Simulação	200 s
Início da Transmissão	2 s
Fim da Transmissão	100 s
Início da Recepção	1 s
Fim da Recepção	101 s

4 RESULTADOS

4.1 INTERVALO DE CONFIANÇA

Para garantir resultados mais confiáveis foram utilizadas ferramentas estatísticas baseada nas cinco rodadas de simulações com sementes diferentes. Com base nesta gama de valores é possível chegar à um intervalo de confiança para o resultado.

Foram coletados os dados de cinco rodadas de simulações para cada variação do parâmetro em estudo. Para todos os resultados foram considerados um grau de confiança de 97,5%.

Para cada configuração de simulação foram utilizadas as equações abaixo que respectivamente calculam o desvio padrão (equação 4.1) e o intervalo de confiança (equação 4.2):

$$\sigma = \sqrt{\sum_{i=1}^n \frac{(x_i - \bar{x})^2}{n-1}} \quad (4.1)$$

$$\mu = \bar{x} \pm z_{\frac{\alpha}{2}} \times \frac{\sigma(x)}{\sqrt{n}} \quad (4.2)$$

4.2 RESULTADOS DAS SIMULAÇÕES

Nas tabelas que seguem estão representados os resultados para os vídeos propostos com os seus respectivos intervalos de confiança. Também são apresentados gráficos com os valores de SSIM de cada vídeo para variação de UEs e GoP, não serão apresentadas em gráficos as informações de PSNR pois as mesmas são similares aos resultados do SSIM. Nos gráficos foram traçadas uma linha limite de 0,88 que represente o limiar do valor 3 de MOS.

Os resultados do vídeo Soccer são apresentados nas Tabelas 6 e 7, onde é possível observar um aumento na degradação da qualidade em relação

ao aumento do tráfego de fundo, conforme mostra a Figura 12 (a) e (b). Com 30 UEs todos os GoP simulados estão abaixo do limite de 0,88.

Com relação a variação do GoP é observada uma redução da degradação para valores de GoP maiores, isto devido à uma maior compactação do vídeo. Esta característica é acentuada com o aumento do tráfego concorrente.

TABELA 6 - VÍDEO "SOCCER", VALORES SSIM

GOP	SSIM			
	UE			
	10	20	30	40
15	0,97 ± 0	0,85 ± 0,04	0,77 ± 0,04	0,76 ± 0,02
30	0,97 ± 0	0,92 ± 0,01	0,87 ± 0,01	0,81 ± 0,07
45	0,97 ± 0	0,94 ± 0,02	0,87 ± 0,03	0,84 ± 0,03
90	0,97 ± 0	0,96 ± 0,01	0,86 ± 0,03	0,82 ± 0,02

TABELA 7 - VÍDEO "SOCCER", VALORES PSNR

GOP	PSNR			
	UE			
	10	20	30	40
15	40,3 dB ± 0	35,8 dB ± 1	32,3 dB ± 1,4	31,9 dB ± 0,3
30	40,1 dB ± 0	38,2 dB ± 0,3	36 dB ± 0,4	33,8 dB ± 2,3
45	40,1 dB ± 0	39,1 dB ± 0,7	36,2 dB ± 0,6	35,0 dB ± 1,4
90	40,0 dB ± 0	39,8 dB ± 0,4	37,7 dB ± 0,7	35,5 dB ± 1,4

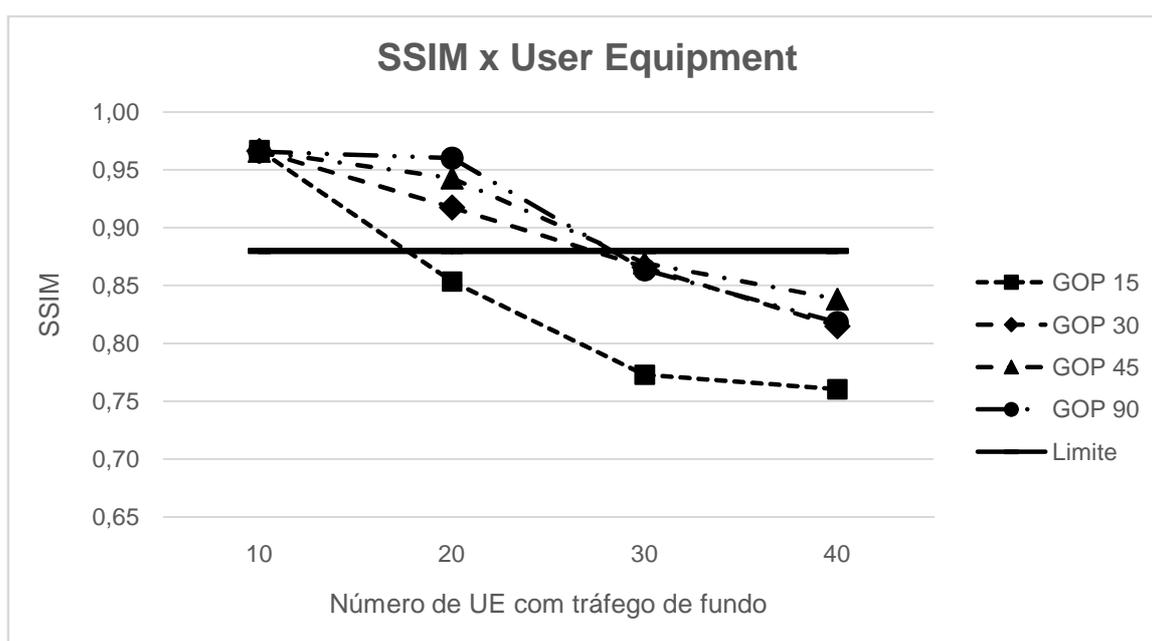


FIGURA 12(a) – GRÁFICO VÍDEO "SOCCER", SSIM POR VARIAÇÃO DE UE

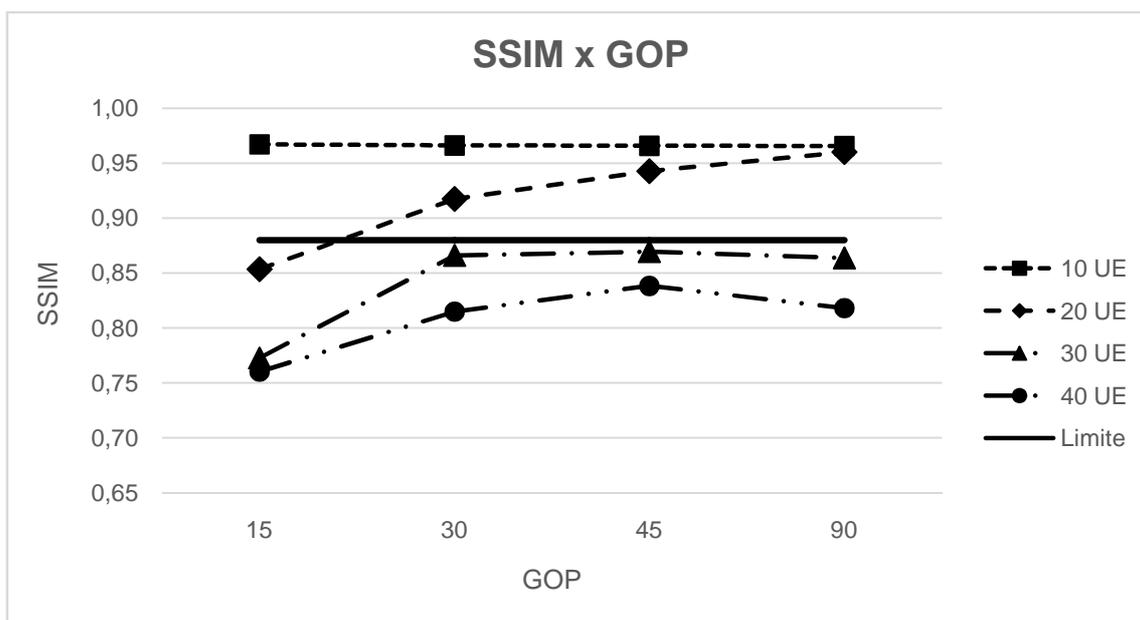


FIGURA 12 (b) – GRÁFICO VÍDEO "SOCCER", SSIM POR VARIAÇÃO DE GOP

Os resultados do vídeo Ice são apresentados nas Tabelas 8 e 9, onde não é verificado qualquer alteração na qualidade percebida para os diferentes cenários, conforme mostra a Figura 13 (a) e (b). Isto devido ao seu baixo valor de *bitrate*, pois com a característica do vídeo (fundo totalmente branco) é possível alcançar um alto nível de compactação espacial.

TABELA 8 - VÍDEO "ICE", VALORES SSIM

GOP	SSIM				
	UE				
	10	20	30	40	60
15	0,98 ± 0	0,98 ± 0	0,98 ± 0	0,98 ± 0	0,98 ± 0
30	0,98 ± 0	0,98 ± 0	0,98 ± 0	0,98 ± 0	0,98 ± 0
45	0,98 ± 0	0,98 ± 0	0,98 ± 0	0,98 ± 0	0,98 ± 0
90	0,98 ± 0	0,98 ± 0	0,98 ± 0	0,98 ± 0	0,98 ± 0

TABELA 9 - VÍDEO "ICE", VALORES PSNR

GOP	PSNR				
	UE				
	10	20	30	40	60
15	47,5 dB ± 0				
30	46,1 dB ± 0				
45	45,0 dB ± 0				
90	43,8 dB ± 0				

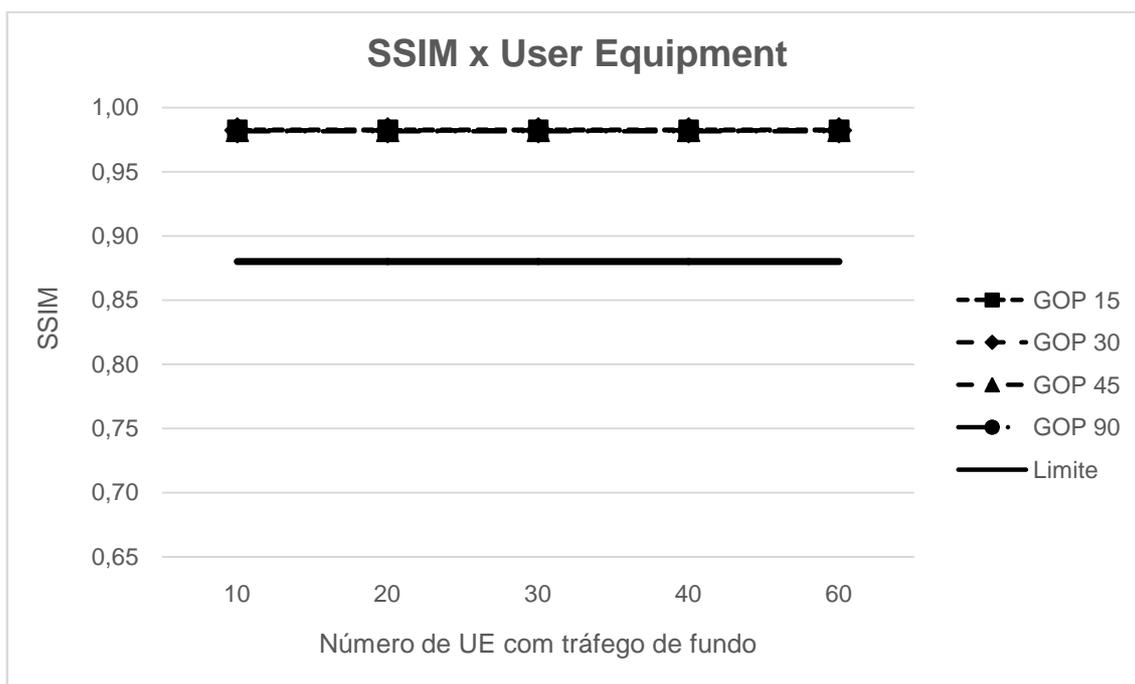


FIGURA 13(a) – GRÁFICO VÍDEO “ICE”, SSIM POR VARIAÇÃO DE UE

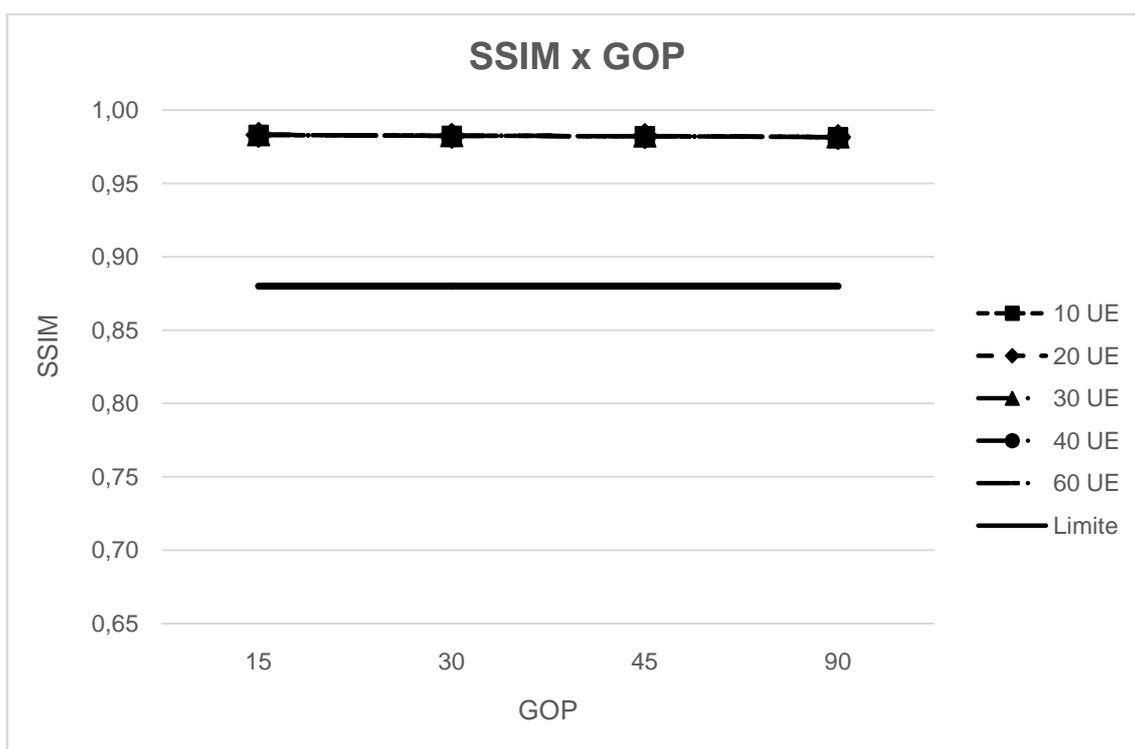


FIGURA 13 (b) – GRÁFICO VÍDEO “ICE”, SSIM POR VARIAÇÃO DE GOP

Os resultados do vídeo Crew são apresentados nas Tabelas 10 e 11, sendo possível observar resultados similares ao vídeo soccer, conforme mostra

a Figura 14 (a) e (b). Verificado um aumento na degradação da qualidade em relação ao aumento do tráfego de fundo, com 20 UEs somente o vídeo com GoP 90 está acima do limite de 0,88.

Com relação a variação do GoP é observada uma redução da degradação para valores de GoP maiores, isto devido à uma maior compactação do vídeo. Esta característica é acentuada com o aumento do tráfego concorrente.

TABELA 10 - VÍDEO "CREW", VALORES SSIM

GOP	SSIM			
	UE			
	10	20	30	40
15	0,97 ± 0	0,80 ± 0,03	0,75 ± 0,05	0,70 ± 0,02
30	0,97 ± 0	0,85 ± 0,03	0,76 ± 0,06	0,73 ± 0,05
45	0,97 ± 0	0,85 ± 0,02	0,79 ± 0,02	0,75 ± 0,05
90	0,97 ± 0	0,89 ± 0,02	0,80 ± 0,04	0,76 ± 0,03

TABELA 11 - VÍDEO "CREW", VALORES PSNR

GOP	PSNR			
	UE			
	10	20	30	40
15	39,8 dB ± 0	33,5 dB ± 0,8	31,6 dB ± 1,5	29,8 dB ± 0,7
30	39,7 dB ± 0	35,2 dB ± 1,1	32,1 dB ± 1,6	30,8 dB ± 2,3
45	39,6 dB ± 0	35,2 dB ± 1,3	33,5 dB ± 1,2	31,8 dB ± 1,9
90	39,6 dB ± 0	36,8 dB ± 0,4	33,7 dB ± 1,3	32,3 dB ± 1,2

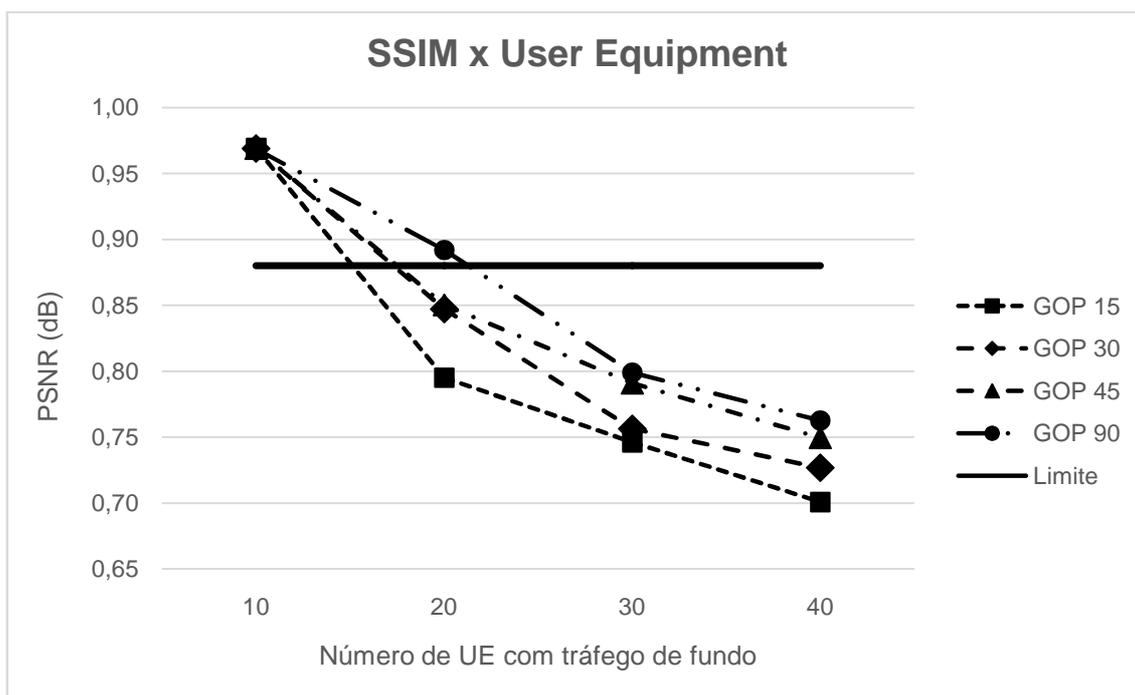


FIGURA 14 (a) – GRÁFICO VÍDEO “CREW”, SSIM POR VARIAÇÃO DE UE

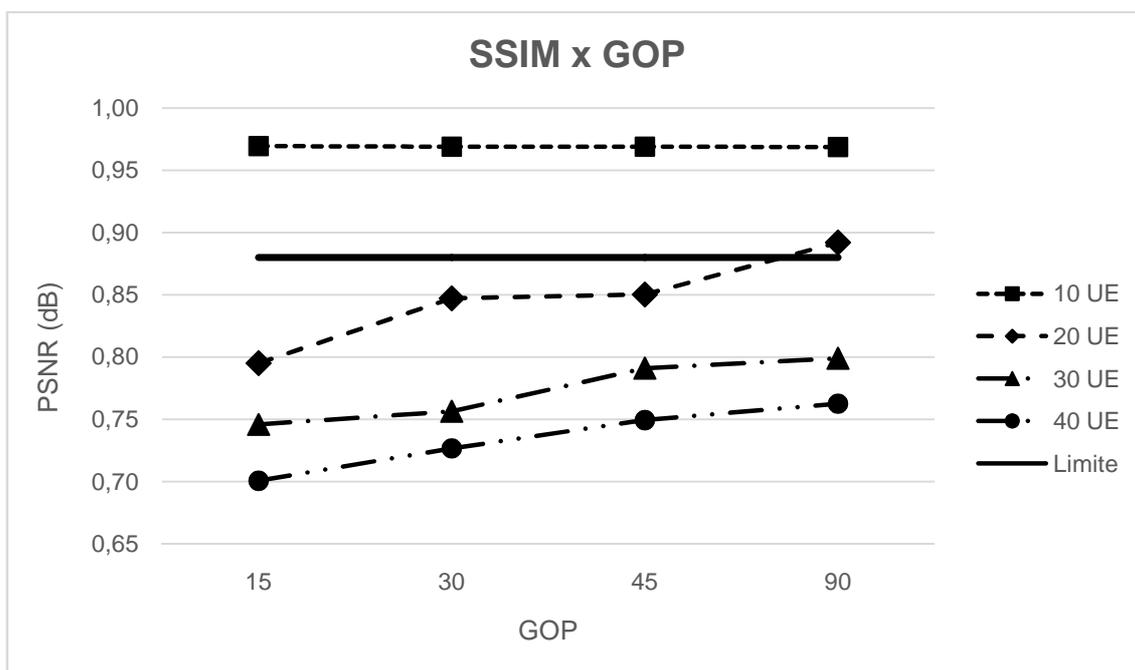


FIGURA 14 (b) – GRÁFICO VÍDEO “CREW”, SSIM POR VARIAÇÃO DE GOP

Devido a problemas de compatibilidade do módulo QoE monitor com alguns vídeos codificados, não foi possível contabilizar os resultados para o vídeo City com GoP de 15 e 30. O desenvolvedor do módulo, Saladino (2012),

relata que novas versões serão desenvolvidas para consertar tal falha, pois o mesmo ainda se encontra em fase beta.

As Tabelas 12 e 13 apresentam os resultados do vídeo City, na qual as análises da degradação são similares aos vídeos Crew e Soccer, porém com uma degradação um pouco menor devido aos valores de *bitratemenores* em relação a esses respectivos vídeos, conforme mostra a Figura 15 (a) e (b).

TABELA 12 - VÍDEO "CITY", VALORES SSIM
SSIM

GOP	UE				
	10	20	30	40	60
45	0,98 ± 0	0,97 ± 0,01	0,88 ± 0,02	0,80 ± 0,02	0,79 ± 0,02
90	0,98 ± 0	0,98 ± 0	0,90 ± 0,01	0,87 ± 0,04	0,79 ± 0,04

TABELA 13 - VÍDEO "CITY", VALORES PSNR
PSNR

GOP	UE				
	10	20	30	40	60
45	39,4 dB ± 0	39,2 dB ± 0,4	35,8 dB ± 0,8	33,5 dB ± 0,5	33,3 dB ± 0,9
90	39,3 dB ± 0	39,3 dB ± 0	36,6 dB ± 0,1	35,9 dB ± 1,7	32,5 dB ± 1,7

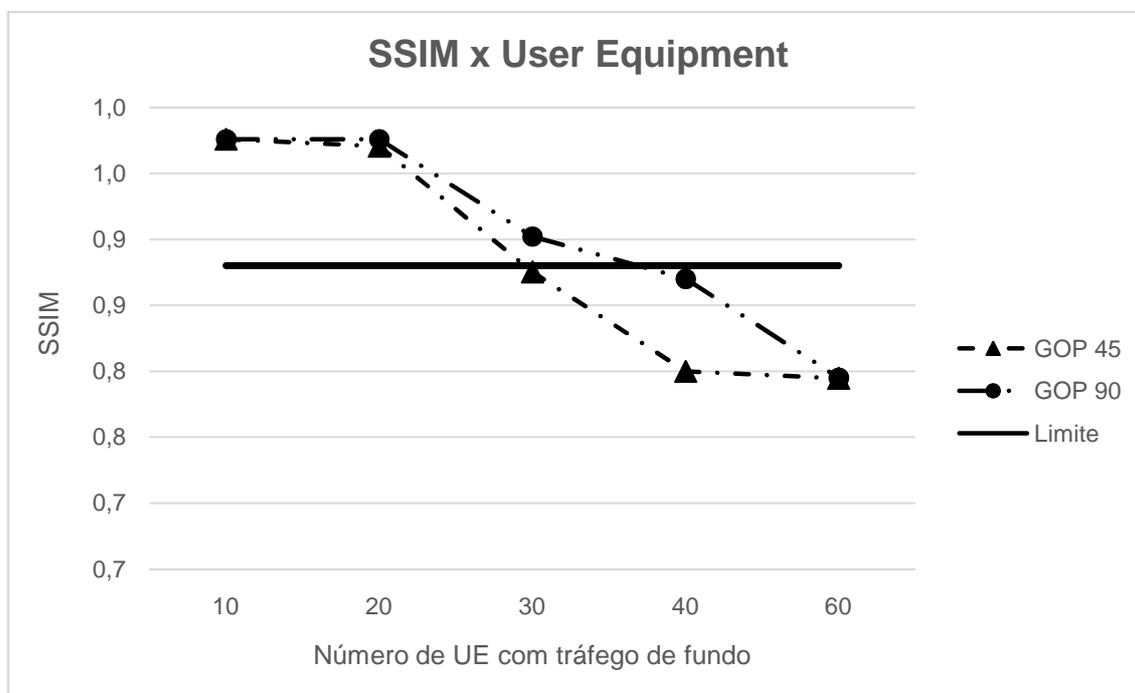


FIGURA 15(a) – GRÁFICO VÍDEO "CITY", SSIM POR VARIAÇÃO DE UE

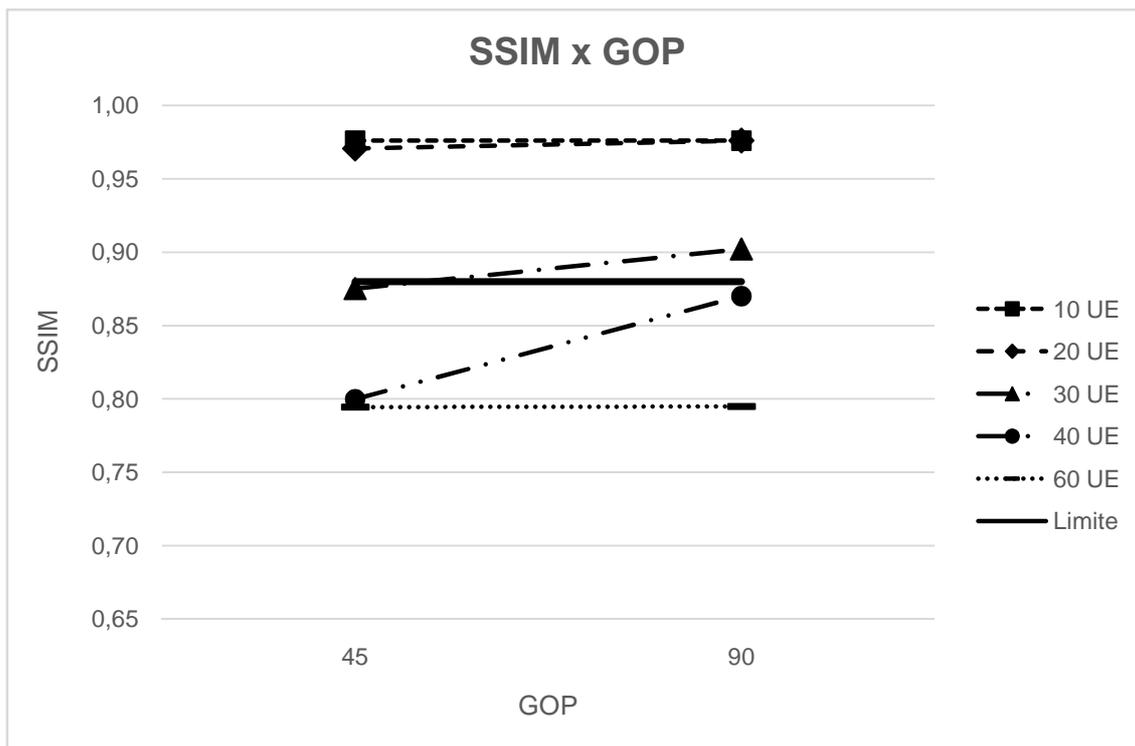


FIGURA 15 (b) – GRÁFICO VÍDEO “CITY”, SSIM POR VARIAÇÃO DE GOP

4.3 ANÁLISE

Analisando os resultados apresentados nos itens anteriores é possível observar o comportamento da degradação da qualidade com as alterações dos parâmetros estabelecidos.

Observou-se em todos os vídeos, com exceção ao “Ice”, um aumento da degradação a partir do aumento do número de UEs ativos produzindo tráfego de fundo, ou seja, a degradação é diretamente proporcional ao tráfego de fundo. Para os vídeos “Soccer” e “Crew” é observado que a partir de 30 UEs os valores de SSIM estão abaixo do limite de 0,88 para todos os valores de GOP, esta característica ocorre com 40 UEs para o vídeo “City” devido este possuir um menor *bitrate* em relação ao “Soccer” e Crew.

Para o vídeo “Ice” não é observada este efeito de degradação, devido a sua característica de baixa movimentação de câmera e fundo de imagem com cor branca dominante, com isto é alcançado uma alta compactação, resultando em valores de *bitrates* muito menores em comparação aos demais vídeos conforme apresentados na Tabela 4.

Ao verificar o nível de degradação, baseado no SSIM, para os vídeos “soccer” e “crew”, é observada a variação de bom (0,97 para ambos com 10 UE) à pobre (“crew”: 0,70 para GoP 15 e 40 UE; “soccer”: 0,76 para GoP 15 e 40 UE). Para o vídeo “city” também há uma queda de bom com 10 UEs (0,98 com GoP 45 e 90) para pobre com 60 UEs (0,79 com GoP 45 e 90).

No que tange aos valores de GoP observou-se uma melhora na qualidade do vídeo na medida que os valores de GoP aumentam, isso se acentua com o aumento do número de UEs ativos produzindo tráfego de fundo. Isto é devido aos menores valores de *bitrate* para altos valores de GoP, assim é possível transmitir o vídeo com menos banda de Internet afetando menos a experiência do usuário.

Os níveis de degradação e suas variações são diferentes para cada vídeo, pois cada um possuem uma característica de redundância espacial e temporal entre quadros diferente. Isto se caracteriza vertiginosamente com o vídeo “Ice” em comparação aos demais vídeos.

5 CONCLUSÃO

Com o aumento do número de usuários da rede LTE e de usuários de streaming de vídeo é necessário criar mecanismos que contribuam para a experiência do espectador mesmo para um alto nível de tráfego na rede.

Neste trabalho foi proposto uma análise do nível de degradação percebida pelo usuário para diferentes vídeos, valores de GoP para codificação em H.264 e número de usuário ativos produzindo tráfego de fundo. Foram utilizadas métricas objetivas como PSNR e SSIM utilizando o software de simulação de redes NS3.

Os resultados apresentados indicam um nível degradação diretamente proporcional à quantidade de UEs gerando tráfego de fundo e inversamente proporcional ao valor de GoP. Porém isto se apresenta com variações diferentes influenciadas pelas características próprias de cada vídeo, entretanto para vídeos com alto grau de redundância não é observado variações na degradação devido aos baixos valores de *bitrate*.

É interessante observar que para cada cenário há uma configuração ideal que nos indicam uma melhor experiência para o espectador, devido a variáveis da rede e de codificação.

Como trabalho futuro pretende-se a ampliação dos testes para diferentes resoluções de vídeos como HD e FullHD, tendo em vista o aumento de demanda para tais resoluções e sua perspectiva futura de crescimento. É necessário considerar também uma gama maior de configuração para valores de usuário de tráfego de fundo e valores de GoP.

6 REFERÊNCIAS

AXIS COMMUNICATIONS. **H.264 video compression standard, new possibilities within video surveillance**, 2008.

CANTARUTTI, R. **Modelo para previsão da qualidade de experiência na Transmissão de vídeo sobre IP**. Dissertação de Mestrado, Universidade Federal do Paraná, 2013.

CAMPANA, M. A.; VÁZQUEZ, E.; VINYES, J. **Performance Modeling of Web Access over HSPA Networks**. 2010 International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), p 710–716, 2010.

FFMPEG. **Aboutffmpeg**. Disponível em: <http://ffmpeg.org/about.html>, acessado em 30/04/2014.

GREENGRASS, J.; EVANS, J.; BEGEN, A. C. **Not all packets are equal, part I**. IEEE Internet Computing, Vol.:13, 2009.

GROSS, J; et al., **Cross-Layer Optimization of OFDM Transmission Systems for MPEG-4 Video Streaming**, ComputerCommunications, vol. 27, no. 11, pp. 1044–1055, 2004.

ITU - Telecommunication Union. **Recommendation ITU-R BT.709-5 (04/2002)Parameter values for the HDTV standards for production and international programme exchange International**, 2002.

ITU - Telecommunication Union. **Recommendation ITU-R BT.601-5 Studio Encoding Parameters Of Digital Television For Standard 4:3 And Wide-Screen 16:9 Aspect Ratios**, 1995.

ITU - Telecommunication Union. **Recommendation BT.500-13, “Methodology for the subjective assessmentof the quality of television pictures”**, 2012.

ITU - Telecommunication Union. **Recommendation P.910, “Subjective video quality assessment methods for multimedia applications”**, 2008.

ITU - Telecommunication Union. **Recommendation J.143, “User requirements for Objective Perceptual Video Quality Measurements in Digital Cable Television”**, 2000.

JIMAA, S. **LTE-A an Overview and Future Research Areas**. Wireless and Mobile Computing, Networking and Communications (WiMob), 2011 IEEE 7th International Conference. p. 395–399, 2014.

JENQ-NENG HWANG. **Multimedia Networking From Theory to Practice**. Cambridge University Press; Ed. 1. 2009.

MANOEL, E. T. M., **Codificação de Vídeo H.264 – Estudo de Codificação Mista de Macroblocos**. Dissertação de Mestrado, Universidade Federal de Santa Catarina, 2007

NETO, A. et al. **Enhancing Smart Grid with Session-Oriented Communication System to Truly Support Reliability and Robustness**. Smart Grid and Renewable Energy, p. 260-165, Novembro 2012.

NOSENSEN, R. **Long-term evolution network architecture**. 2009 IEEE International Conference on Microwaves, Communications, Antennas and Electronics Systems, p. 1–4, nov. 2009.

PERNES, D.; NEVES, D. **Análise de Cobertura e Capacidade em Redes Móveis LTE de Quarta Geração (4G)**. Proc URSI Seminar of the Portuguese Committee, Lisboa, Portugal, Vol. 1, pp. 1 - 1, Novembro, 2012.

PRICE D., HARTE L., **Introduction to MPEG 2nd Edition**. Althos Publishing, 2012.

SALADINO, D.; PAGANELLI A.; CASONI M. **A tool for multimedia quality assessment in NS3: QoE Monitor**. University of Modena and Reggio Emilia, Simulation Modelling Practice and Theory, Volume 32, Pages 30–41. Março, 2013

WANG, Z.; R. SHEIKH; C. BOVIK. **Objective video quality assessment**. The University of Texas at Austin. Broadcasting, IEEE Transactions, Volume:57, p. 165 – 182, 2003.

WANG, Z.; BOVIK, A.C.; SHEIKH, H.R.; SIMONCELLI, E.P. **Image quality assessment: from error visibility to structural similarity**. Image Processing, IEEE Transactions, Volume: 13, p. 600-612. 2004.

YAMADA, F.; et al. **Parte I – sistemas de TV digital**. Revista Mackenzie de Engenharia e Computação, Vol.5, No.5, 2004.

ZINNER, T. et al. **Towards QoE Management for Scalable Video Streaming**. University of Wuerzburg. 21th ITC Specialist Seminar on Multimedia Applications - Traffic, Performance and QoE. 2010.

ZHOU, D. et al. **Evaluation of TCP performance with LTE downlink schedulers in a vehicular environment**. 2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC), p. 1064–1069, jul. 2013.

APÊNDICE 1 – CÓDIGO PARA SIMULAÇÃO

```

/*=====
=====
*   Título: AVALIAÇÃO DA QUALIDADE DE STREAMING DE VÍDEO EM
SISTEMAS LTE
*
*   Trabalho de Conclusão de Curso apresentado como exigência
parcial
*   para obtenção do grau de Bacharel em Engenharia Elétrica,
Ênfase
*   Sistemas Eletrônicos Embarcados, à Universidade Federal do
Paraná.
*
*   Alunos:           Diego Franco Martins
*                   Felipe de Souza Pinheiro
*
*   Orientador: Prof. Dr. Carlos Marcelo Pedroso
*/

#include "ns3/internet-stack-helper.h"
#include "ns3/ipv4-address-helper.h"
#include "ns3/ipv4-interface-container.h"
#include "ns3/ipv4-routing-table-entry.h"
#include "ns3/ipv4-global-routing-helper.h"
#include "ns3/simulation-dataset.h"
#include "ns3/h264-packetizer.h"
#include "ns3/multimedia-application-sender.h"
#include "ns3/multimedia-application-receiver.h"
#include "ns3/multimedia-file-rebuilder.h"
#include "ns3/error-model.h"
#include "ns3/mpeg4-container.h"
#include "ns3/psnr-metric.h"
#include "ns3/ssim-metric.h"
#include "ns3/nstime.h"
#include "ns3/core-module.h"
#include "ns3/internet-module.h"
#include "ns3/network-module.h"
#include "ns3/point-to-point-module.h"
#include "ns3/udp-client-server-helper.h"
#include "ns3/packet-sink-helper.h"
#include "ns3/flow-monitor-module.h"
#include "ns3/applications-module.h"
#include "ns3/lte-helper.h"
#include "ns3/epc-helper.h"
#include "ns3/mobility-module.h"
#include "ns3/point-to-point-helper.h"
#include "ns3/config-store.h"
#include "ns3/flow-monitor-helper.h"
#include "ns3/tcp-socket.h"
#include "ns3/olsr-module.h"
#include "ns3/dsdv-module.h"
#include <iostream>
#include <cstring>
#include <cassert>

using namespace ns3;

NS_LOG_COMPONENT_DEFINE ("Simulacao - QoE Monitor / LTE");

```

```

/* Simple FFMpeg wrapper for encoding/decoding purposes */
intFfmpeg(std::string inputFileNames, std::string outputFileNames)
{
std::stringstream command;
command<< "ffmpeg -i " <<inputFileNames
<< " " <<outputFileNames;

/* I execute the command */
intreturnCode = system(command.str().c_str());
if (returnCode != 0)
{
std::cout<< "Error while calling FFMpeg, abort.\n";
}
returnreturnCode;
}

/*****
/*
/*          Rotina Principal          *
/*
/*
/*****/

int
main (intargc, char *argv[])
{

/*****
/* Inicialização das variáveis *
/*****/

uint16_t seed = 1; // Default do seed
uint16_t numberOfNodes1 = 15; // Default de UE para tráfego de
fundo
uint16_t numberOfNodes2 = 1; // Default de UE que recebe o vídeo
std::string name ("TCC_versao_Final");

CommandLinecmd;
cmd.AddValue("seed", "Seed of simulation", seed);
cmd.AddValue("numberOfNodes1", "Number of UE1 pairs",
numberOfNodes1);
cmd.Parse(argc, argv);

ns3::SeedManager::SetSeed(seed);

std::stringjitterBufferLength("200ms");
std::stringsimulationDuration("200s");
std::stringtransmitterStartTime("2s");
std::stringtransmitterStopTime("100s");
std::stringreceiverStartTime("1s");
std::stringreceiverStopTime("101s");
unsignedintmtu = 1400;

/*****
/*          Extract the coded filename          *
/*****/

```

```

std::stringcodedFilename(argv[1]);

    /*****
    /* Opções para simulação *
    /*****/

boolenablePsnr = true;
boolenableSsim = true;

    /*****
    /*      Filenames creation      *
    /*****/

size_textPosition = codedFilename.rfind(".mp4");
std::stringfileIdentifier = codedFilename.substr(0, extPosition);

std::stringrawFilename = fileIdentifier + ".yuv";
std::stringreceivedFilename = fileIdentifier + ".received.mp4";
std::stringreceivedRawFilename = fileIdentifier + ".received.yuv";

std::stringtraceFileID = fileIdentifier + "-trace";
std::stringmetricFile = fileIdentifier + "-metric";

    /*****
    /*      QoE monitor setup      *
    /*****/

SimulationDataset* dataset = new SimulationDataset();

dataset->SetOriginalRawFile(rawFilename);
dataset->SetOriginalCodedFile(codedFilename);
dataset->SetReceivedCodedFile(receivedFilename);
dataset->SetReceivedReconstructedFile(receivedRawFilename);
dataset->SetTraceFileId(traceFileID);

    Mpeg4Container mpeg4ReadingContainer = Mpeg4Container(dataset-
>GetOriginalCodedFile(),

Container::READ, AVMEDIA_TYPE_VIDEO);
mpeg4ReadingContainer.InitForRead();
dataset-
>SetSamplingInterval(mpeg4ReadingContainer.GetSamplingInterval());

H264Packetizer videoPacketizer(mtu, dataset);

    /*****
    /* Cria elementos principais do LTE *
    /*****/

Ptr<LteHelper>LteObjeto = CreateObject<LteHelper> ();
Ptr<EpcHelper>EpcObjeto = CreateObject<EpcHelper> ();
LteObjeto ->SetEpcHelper (EpcObjeto);

    /*****
    /*      Parâmetros do LTE      *
    /*****/

```

```

LteObjeto ->SetEnbAntennaModelType ("ns3::CosineAntennaModel");
LteObjeto ->SetEnbAntennaModelAttribute ("Orientation", DoubleValue
(0));
LteObjeto ->SetEnbAntennaModelAttribute ("Beamwidth", DoubleValue
(60));
LteObjeto ->SetEnbAntennaModelAttribute ("MaxGain", DoubleValue
(0.0));

ConfigStoreinputConfig;
inputConfig.ConfigureDefaults();
inputConfig.SetFilename("input-default.txt");

Ptr<Node>pgw = EpcObjeto->GetPgwNode ();

    /******
    /*      Cria os elementos      *
    /*******/

NodeContainerDevices;
Devices.Create (3);
Ptr<Node> Router = Devices.Get(0);
Ptr<Node>VideoServer = Devices.Get(1);
Ptr<Node>TcpServer = Devices.Get(2);

    /******
    /* Cria conexão entre os elementos *
    /*******/

PointToPointHelper p2ph;
p2ph.SetDeviceAttribute ("DataRate", DataRateValue (DataRate
("10Gbps"))); // Enlace = 10Gbps
p2ph.SetDeviceAttribute ("Mtu", UIntegerValue (1500));
// MTU = 1500 bits
p2ph.SetChannelAttribute ("Delay", StringValue("2ms"));
// Delay do canal = 2ms

NetDeviceContainerEnlace[3];

Enlace[0] = p2ph.Install (Router, pgw);
Enlace[1] = p2ph.Install (Router, VideoServer);
Enlace[2] = p2ph.Install (Router, TcpServer);

    /******
    /* Cria rede entre os elementos *
    /*******/

InternetStackHelper internet;
internet.Install (Devices);

Ipv4AddressHelper ipv4h;
ipv4h.SetBase ("192.168.0.0", "255.255.255.252");

Ipv4InterfaceContainer interfacesIP[3];

for(unsigned inti = 0; i< 3; i++)
{
interfacesIP[i] = ipv4h.Assign(Enlace[i]);
}

```

```

ipv4h.NewNetwork();
    }

    Ipv4StaticRoutingHelper ipv4Roteamento;
    Ptr<Ipv4StaticRouting> RotaEstaticaEnlace1 =
    ipv4Roteamento.GetStaticRouting (Router->GetObject<Ipv4> ());
    RotaEstaticaEnlace1 ->AddNetworkRouteTo (Ipv4Address ("7.0.0.0"),
    Ipv4Mask ("255.0.0.0"), 1);

    Ptr<Ipv4StaticRouting> RotaEstaticaEnlace2 =
    ipv4Roteamento.GetStaticRouting (pgw->GetObject<Ipv4> ());
    RotaEstaticaEnlace2->SetDefaultRoute
    (interfacesIP[0].GetAddress (0), 2);

    Ptr<Ipv4StaticRouting> RotaDefault1 =
    ipv4Roteamento.GetStaticRouting (VideoServer->GetObject<Ipv4> ());
    RotaDefault1->SetDefaultRoute (interfacesIP[1].GetAddress (0),
    1);

    Ptr<Ipv4StaticRouting> RotaDefault2 =
    ipv4Roteamento.GetStaticRouting (TcpServer->GetObject<Ipv4> ());
    RotaDefault2->SetDefaultRoute (interfacesIP[2].GetAddress (0), 1);

    /*****
    /*      Cria os elementos Móveis      *
    *****/

    NodeContainer ueNodes1;
    NodeContainer ueNodes2;
    NodeContainer enbNodes;
    enbNodes.Create(1);
    ueNodes1.Create(numberOfNodes1);
    ueNodes2.Create(numberOfNodes2);

    /*****
    /*      Instala o modelo de Mobilidade      *
    *****/

    Ptr<ListPositionAllocator> positionAlloc0 =
    CreateObject<ListPositionAllocator> ();
    positionAlloc0->Add (Vector(0, 0, 0));

    Ptr<UniformRandomVariable> rand;
    rand = CreateObject<UniformRandomVariable>();

    Ptr<ListPositionAllocator> positionAlloc1 =
    CreateObject<ListPositionAllocator> ();
    for (uint16_t i = 0; i< numberOfNodes1; i++)
    {
        positionAlloc1->Add (Vector(rand->GetInteger(0,1000), rand-
    >GetInteger(0,1000), 0));
    }
    Ptr<ListPositionAllocator> positionAlloc2 =
    CreateObject<ListPositionAllocator> ();
    for (uint16_t i = 0; i< numberOfNodes2; i++)
    {
        positionAlloc2->Add (Vector(rand->GetInteger(0,500), rand-
    >GetInteger(0,500), 0));
    }

```

```

}

MobilityHelper mobility0;
mobility0.SetMobilityModel("ns3::ConstantPositionMobilityModel");
mobility0.SetPositionAllocator(positionAlloc0);
mobility0.Install(enbNodes);

MobilityHelper mobility1;
mobility1.SetMobilityModel("ns3::ConstantPositionMobilityModel");
mobility1.SetPositionAllocator(positionAlloc1);
mobility1.Install(ueNodes1);

MobilityHelper mobility2;
mobility2.SetMobilityModel("ns3::ConstantPositionMobilityModel");
mobility2.SetPositionAllocator(positionAlloc2);
mobility2.Install(ueNodes2);

/*****
 * Instala Dispositivos LTE nos nós *
 *****/

NetDeviceContainer enbDevs;
NetDeviceContainer ueDevsA;
NetDeviceContainer ueDevsB;
enbDevs = LteObjeto->InstallEnbDevice (enbNodes);
ueDevsA = LteObjeto->InstallUeDevice (ueNodes1);
ueDevsB = LteObjeto->InstallUeDevice (ueNodes2);

/*****
 * Instala IP nos UEs *
 *****/

internet.Install (ueNodes1);
Ipv4InterfaceContainer uE_Interface1;
uE_Interface1 = EpcObjeto->AssignUeIpv4Address (NetDeviceContainer
(ueDevsA));

internet.Install (ueNodes2);
Ipv4InterfaceContainer uE_Interface2;
uE_Interface2 = EpcObjeto->AssignUeIpv4Address (NetDeviceContainer
(ueDevsB));

/*****
 * Defini EPC como gateway para os UEs *
 *****/

for (uint32_t u = 0; u < ueNodes1.GetN (); u++)
{
Ptr<Node>ueNode_tcp = ueNodes1.Get (u);
Ptr<Ipv4StaticRouting> ueStaticRouting1 =
ipv4Roteamento.GetStaticRouting (ueNode_tcp->GetObject<Ipv4> ());
ueStaticRouting1->SetDefaultRoute (EpcObjeto-
>GetUeDefaultGatewayAddress (), 1);
}

for (uint32_t u = 0; u < ueNodes2.GetN (); u++)
{
Ptr<Node>ueNode_video = ueNodes2.Get (u);

```

```

Ptr<Ipv4StaticRouting> ueStaticRouting2 =
ipv4Roteamento.GetStaticRouting (ueNode_video->GetObject<Ipv4> ());
    ueStaticRouting2->SetDefaultRoute (EpcObjeto-
>GetUeDefaultGatewayAddress (), 1);
    }

    /*****
    /*      Anexaos UEs no eNB      *
    /*****/

LteObjeto->Attach (ueDevsA, enbDevs.Get (0));
LteObjeto->Attach (ueDevsB, enbDevs.Get (0));

    /*****
    /*      Transmissão do video      *
    /*****/

    /* Receiver side application setup
    * First I create an mpeg4 container for the received file
    (output) */
    Mpeg4Container mpeg4Container = Mpeg4Container(dataset-
>GetReceivedCodedFile(),
                                                    Container::WRITE,
AVMEDIA_TYPE_VIDEO);

    /* Now I take the codec context from the reading context (input)
    and I copy
    * its settings to the output codec context */
mpeg4Container.SetCodecContext(mpeg4ReadingContainer.GetCodecContext());

    /* I copy the stream's settings from the reading context (input)
    to the output context */
mpeg4Container.SetStream(mpeg4ReadingContainer.GetStream());

    /* Now I can initialize the output container for writing */
mpeg4Container.InitForWrite();

    /* Create the receiver's multimedia file rebuilder */
MultimediaFileRebuilderfileRebuilder(&mpeg4Container, dataset);

    /* Create the receiver's side application */
Ptr<MultimediaApplicationReceiver>multimediaReceiver =
CreateObject<
MultimediaApplicationReceiver> (ueNodes2.Get(0), dataset,
Time(std::string(jitterBufferLength)));

multimediaReceiver->SetupReceiverPort(400);
multimediaReceiver->SetupFileRebuilder(&fileRebuilder);

ueNodes2.Get(0)->AddApplication(multimediaReceiver);

multimediaReceiver->SetStartTime(Time(receiverStartTime));
multimediaReceiver->SetStopTime(Time(receiverStopTime));

    /* Create the sender's side application */

```

```

Ptr<MultimediaApplicationSender>multimediaSender =
CreateObject<MultimediaApplicationSender>
                                                    (VideoServer,
&videoPacketizer, dataset);

multimediaSender-
>SetupDestinationAddress(uE_Interface2.GetAddress(0), 400);
VideoServer->AddApplication(multimediaSender);

multimediaSender->SetStartTime(Time(transmitterStartTime));
multimediaSender->SetStopTime(Time(transmitterStopTime));

    /******
    /*      Trafego de Fundo em TCP      *
    /******/

uint16_tulPort = 2000;
for (uint32_t u = 0; u < ueNodes1.GetN (); u++)
{

    ApplicationContainerctSenderContainer, ctReceiverContainer;

        /* Sender side */
BulkSendHelperctTcpSender("ns3::TcpSocketFactory", InetSocketAddress
(uE_Interface1.GetAddress(u), ulPort));
ctTcpSender.SetAttribute("MaxBytes", UintegerValue(0));
ctSenderContainer = ctTcpSender.Install(TcpServer);

        /* Receiver side */

PacketSinkHelperctTcpReceiver("ns3::TcpSocketFactory", InetSocketAddress
ress (Ipv4Address::GetAny(), ulPort));
ctReceiverContainer = ctTcpReceiver.Install(ueNodes1.Get(u));

        ulPort++;

ctSenderContainer.Start(Seconds (1));
ctSenderContainer.Stop(Time(receiverStopTime));

ctReceiverContainer.Start(Seconds (1));
ctReceiverContainer.Stop(Time(receiverStopTime));
}

/* =====*/

    /* Flow monitor setup */
FlowMonitorHelperflowmon_helper;
Ptr<FlowMonitor> monitor = flowmon_helper.InstallAll();
monitor->SetAttribute("DelayBinWidth", ns3::DoubleValue(0.0005));
monitor->SetAttribute("JitterBinWidth", ns3::DoubleValue(0.0005));
monitor->SetAttribute("PacketSizeBinWidth", ns3::DoubleValue(20));

    /* Actual simulation start */
std::cout<< "Starting simulation\n";

    Simulator::Stop (Time(simulationDuration));
    Simulator::Run();

```

```

    /* Print traces to file */
    dataset->PrintTraces(true);

    /* Decoding received file */
    std::cout<< "Received file decoding with FFMpeg...\n";
    assert(Ffmpeg(receivedFilename, receivedRawFilename) == 0);

    /* Decode the original h264 file to perform video comparison */
    //std::cout<< "Original YUV (raw) file decoding with
    FFMpeg...\n";
    //assert(Ffmpeg(codedFilename, rawFilename) == 0);

    if (enablePsnr)
    {
        /* Computing PSNR */
        std::cout<< "PSNR computing...";
        std::cout.flush();

        PsnrMetricpsnr;
        psnr.EvaluateQoe(rawFilename, receivedRawFilename);

        /* Print the metric output without any header */
        psnr.PrintResults(metricFile.c_str(), true);
        std::cout<< " done!\n";
    }

    if (enableSsim)
    {
        /* Computing SSIM */
        std::cout<< "SSIM computing...";
        std::cout.flush();

        SsimMetricssim;
        ssim.EvaluateQoe(rawFilename, receivedRawFilename);

        /* Print the metric output without any header */
        ssim.PrintResults(metricFile.c_str(), true);
        std::cout<< " done!\n";
    }

    /* Flow monitor post-processing */
    Ptr<Ipv4FlowClassifier> classifier =
    DynamicCast<Ipv4FlowClassifier> (flowmon_helper.GetClassifier ());
    std::map<FlowId, FlowMonitor::FlowStats> stats = monitor-
    >GetFlowStats();

    for (std::map<FlowId, FlowMonitor::FlowStats>::const_iterator
    iterator = stats.begin (); iterator != stats.end (); iterator++)
    {
        /* Extract the classifier info. for the current flow */
        Ipv4FlowClassifier::FiveTuple tuple = classifier->FindFlow
        (iterator->first);

        uint16_tdestinationPort = tuple.destinationPort;
        uint16_tsourcePort = tuple.sourcePort;
        Ipv4Address sourceAddress = tuple.sourceAddress;
        Ipv4Address destinationAddress = tuple.destinationAddress;
    }

```

```

        /* I extract loss, throughput (in bps), delay, jitter and
        packet size */
        doublesenderStreamDuration = (iterator->second.timeLastTxPacket -
        iterator->second.timeFirstTxPacket).GetSeconds();
        doublereceiverStreamDuration = (iterator->second.timeLastRxPacket -
        iterator->second.timeFirstRxPacket).GetSeconds();
        doublesampleLoss = (iterator->second.txPackets - iterator-
        >second.rxPackets)/(double (iterator->second.txPackets));
        doublesampleThroughput = ((iterator->second.rxBytes * 8)/(double
        ((iterator->second.timeLastRxPacket -
        iterator->second.timeFirstRxPacket).GetSeconds())));
        doublesampleDelay = (iterator-
        >second.delaySum).GetSeconds()/(double (iterator-
        >second.rxPackets));
        doublesampleJitter = (iterator-
        >second.jitterSum).GetSeconds()/(double (iterator->second.rxPackets
        - 1));
        doublesamplePacketSize = (iterator->second.rxBytes)/(double
        (iterator->second.rxPackets));
        uint32_tsampledPacketSent = iterator->second.txPackets;
        uint32_tsamplePacketReceived = iterator->second.rxPackets;

        std::cout<< "Throughput, delay, jitter, loss, packet size, packet
        sent, packet received\n";
        std::cout<< "Flow "<< iterator->first << " ( " <<sourceAddress<<
        ":" <<sourcePort<<
        " -> " <<destinationAddress<< ":" <<destinationPort<< " )\n";
        std::cout<<std::fixed
        <<std::setw(6) <<std::setprecision(0) <<sampleThroughput<< ", "
        <<std::setw(8) <<std::setprecision(6) <<sampleDelay<< ", "
        <<std::setw(8) <<std::setprecision(6) <<sampleJitter<< ", "
        <<std::setw(8) <<std::setprecision(6) <<sampleLoss<< ", "
        <<std::setw(5) <<std::setprecision(0) <<samplePacketSize<< ", "
        <<std::setw(8) <<std::setprecision(0) <<samplingPacketSent<< ", "
        <<std::setw(8) <<std::setprecision(0) <<samplePacketReceived<<
        "\n";

        std::cout.precision(6);
        std::cout<< "Sender stream duration: " <<senderStreamDuration<<
        "\n";
        std::cout<< "Receiver stream duration: " <<receiverStreamDuration<<
        "\n";
    }

    delete dataset;
    Simulator::Destroy();
    return 0;
}

```