

Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Coordenação de Pós-Graduação em Ciência da Computação

Avaliação de Qualidade de Vídeos Transmitidos via
Vídeo *Streaming* em Ambientes Residenciais

Luiz Teixeira Machado Neto

Dissertação submetida à Coordenação do Curso de Pós-Graduação em Ciência da Computação da Universidade Federal de Campina Grande - Campus I como parte dos requisitos necessários para obtenção do grau de Mestre em Ciência da Computação.

Área de Concentração: Ciência da Computação

Linha de Pesquisa: Redes de Computadores

Reinaldo Gomes

Campina Grande, Paraíba, Brasil

©Luiz Teixeira Machado Neto, 01/03/2015

Resumo

A utilização de serviços de *streaming* cresceu bastante nos últimos anos, por meio de sistemas como Youtube, Hulu, Netflix, Vimeo, etc. Utilizando o *stream*, os vídeos são transmitidos e exibidos em tempo real e em qualquer lugar do mundo por meio da internet. Muitos usuários destes sistemas os utilizam em sua residência onde é bastante comum encontrar uma rede sem fio (devido à mobilidade que se pode alcançar por meio deste tipo de rede). Apesar de mais mobilidade, uma rede sem fio está mais suscetível a interferências do que a rede cabeada e, por isso, um vídeo pode ter sua imagem degradada com mais facilidade. Neste trabalho, é avaliada a transmissão de vídeos via *stream* para descobrir se o vídeo realmente é degradado pela transmissão; se tipos de conteúdos diferentes afetam a qualidade do vídeo recebido; e se de acordo com o padrão de compressão utilizado, é possível observar melhorias na qualidade do vídeo recebido. Por meio de uma abordagem experimental com um *design* de experimentos fatorial completo, foram feitas transmissões de vídeos utilizando o H.264, o HEVC e o MPEG-4; padrões mais utilizados atualmente. Além de definir os padrões, foram definidas outras variáveis: porcentagem de ocupação do canal de transmissão (com o objetivo de avaliar a degradação dos vídeos de acordo com a competição que a rede está sofrendo); potência do sinal de transmissão (com o objetivo de avaliar o impacto da qualidade do sinal da rede no vídeo recebido); quantidade de movimento no vídeo (para avaliar se a quantidade de movimento que o vídeo exhibe impacta na sua qualidade). Nos experimentos, foi utilizado um ambiente residencial que conta as interferências de outras redes, exatamente como pode acontecer em um ambiente real. Para definir as porcentagens de ocupação, foram realizados experimentos para medição da capacidade máxima de transmissão da rede de testes. Em se tratando da quantidade de movimento, foi necessário fazer uma classificação prévia dos vídeos de acordo com características espaciais e temporais de cada vídeo. Os vídeos foram separados em três categorias e dentro dessas categorias, três vídeos foram escolhidos aleatoriamente para participar dos experimentos. Os resultados mostram que o HEVC obteve a melhor média para as métricas de qualidade de vídeo escolhidas, sendo o *codec* que menos perde qualidade numa transmissão sem fio. Também foi possível observar

que a quantidade de movimento foi o parâmetro que mais influenciou na qualidade do vídeos nos experimentos realizados.

Abstract

The use of streaming services has grown significantly in recent years, through systems such as Youtube, Hulu, Netflix, Vimeo, etc. Using the stream, videos are transmitted and displayed in real time and from anywhere in the world via the Internet. Many users of these systems use the same in their homes where it is quite common to find a wireless network (due to the mobility we can achieve through this type of computer network). In spite of having more mobility, a wireless network is more susceptible to interference than the wired network so a video can have its picture degraded more easily just because it is transmitted over a wireless network. We evaluate the transmission of videos via stream to find out whether the video is actually degraded by transmission; if different types of content affect the quality of the received video; and if there are compression standarts (H.264 and MPEG-4 HEVC, the most currently used) which ensure a better received video quality. Through an experimental approach with a design of full factorial experiments, several transmissions of videos were made in the three chosen standarts. In addition to defining the standarts, other variables were defined as: transmission channel occupancy percentage (in order to evaluate the degradation of videos according to occupation); power transmission signal (in order to assess the impact of network signal quality in the video received); amount of motion in the video (to evaluate whether the amount of motion the video displays impacts on its quality). For the experiments we used a residential environment that has all the interference from other networks, just as it can happen in a real environment. To set the occupancy percentages, experiments were performed to measure the maximum transmission capacity of the test network. Concerning the quantity of movement, it was necessary to make a preliminary classification of videos according to spatial and temporal characteristics of each one. The videos were separated into three categories and within these categories, three videos were chosen at random to participate in the experiments. The results show that the HEVC achieved the highest average for quality metrics of the chosen videos, and the codec is the one that loses less quality in a wireless transmission. The amount of movement affects the quality of the received video, and the greater the amount of motion, the bigger the loss of image quality.

Agradecimentos

Ao Professor Reinaldo Gomes, pela confiança, pela oportunidade de trabalhar ao seu lado e por todas as orientações na execução deste trabalho.

Aos colegas do Laboratório de Teleinformática da Universidade Federal de Campina Grande.

À Professora Joseana Fachine, pelas orientações sobre inteligência artificial e classificação.

À Ana Clara, por tudo que foi feito para que eu conseguisse chegar até aqui.

À Luiz Teixeira Machado, por ser a figura de um Pai e me proporcionar a oportunidade de ser o que sou hoje.

À Elza Cavalcanti Machado, por ser a segunda Mãe e por todos os carinhos.

À Lívia Maia, por todo o amor, carinho e principalmente pela paciência durante todo o mestrado.

À toda a minha família por todo o incentivo durante toda a minha vida acadêmica.

Aos meus grandes amigos, que sempre me incentivaram e me ajudaram.

Conteúdo

1	Introdução	17
1.1	Objetivos e Relevância	19
1.1.1	Objetivos	19
1.1.2	Relevância	19
2	Fundamentação Teórica e Revisão da Literatura	21
2.1	Fundamentação Teórica	21
2.1.1	Vídeo e Vídeo <i>Stream</i>	21
2.1.2	Características	22
2.1.3	Padrões de Compressão	23
2.1.4	Métricas	25
2.1.5	Compressão de Vídeo	26
2.1.6	Comunicação Sem Fio	28
2.1.7	Conceitos de Inteligência Artificial	29
2.2	Revisão de Literatura	30
2.2.1	Avaliação de formatos de compressão e de conteúdo dos vídeos .	30
2.2.2	Métricas para avaliação de qualidade de vídeos	31
2.2.3	Metodologias de avaliação de qualidade de vídeos	32
3	Sistemas Utilizados	33
3.1	Hardware	33
3.1.1	Soekris net5501	33
3.1.2	Ubiquiti eXtreme Range 2 Radio Transmissor	35
3.2	Software	36

3.2.1	Voyage Linux	36
3.2.2	FFMpeg	37
3.2.3	Iperf3	37
3.2.4	VQMT: Video Quality Measurement Tool	38
3.2.5	Tcpdump	38
3.2.6	OpenCV	39
3.2.7	Weka	39
3.2.8	R	39
4	Planejamento dos Experimentos	40
4.1	Seleção do Contexto	40
4.2	Hipóteses	40
4.2.1	Avaliação da perda de qualidade	40
4.2.2	Avaliação do conteúdo	41
4.2.3	Avaliação dos <i>codecs</i>	41
4.3	Variáveis Independentes	41
4.3.1	Variáveis Independentes Relacionadas à Rede	41
4.3.2	Variável Independente Relacionada aos Vídeos	42
4.4	Variáveis Dependentes	43
4.4.1	Variáveis Dependentes Relacionadas à Rede	43
4.4.2	Variáveis Dependentes Relacionadas aos Vídeos	44
4.5	Variável de Blocagem	44
4.6	Seleção da Amostra de Vídeos	45
4.7	<i>Design</i> dos Experimentos	45
4.8	Instrumentação	46
4.8.1	Objetos	46
4.8.2	Orientações	46
4.8.3	Medições	47
4.9	Validação	47
4.9.1	Ameaças à Conclusão	47
4.9.2	Ameaças Internas	48

4.9.3	Ameaças de Construção	48
4.9.4	Ameaças Externas	48
4.10	Diagrama de Execução de Experimentos e Análise de Resultados	49
4.11	Ambiente de Execução de Experimentos	49
4.11.1	Disposição do Ambiente	49
4.11.2	Redes Sem Fio Presentes no Ambiente	50
4.12	Classificação dos Vídeos	51
4.12.1	Vídeos	51
4.12.2	Extração de Características	52
4.12.3	Classificação Prévia	52
4.12.4	Classificador Utilizado	52
4.12.5	Categorias de Vídeos	52
4.12.6	Experimentos	52
4.12.7	Apresentação e Análise de Resultados	53
4.13	Diagrama de Classificação dos Vídeos	54
4.14	Análise da Taxa de Transmissão	55
4.14.1	Experimentos	55
4.14.2	Resultados Da Taxa de Transmissão	56
4.15	Diagrama de Extração da Porcentagem de Ocupação	58
5	Análise dos Dados Obtidos	59
5.1	Metodologia de Análise de Resultados	59
5.1.1	Análise de Normalidade	59
5.1.2	Teste de Igualdade	60
5.1.3	Teste de Alternativa Maior	61
5.2	<i>Codec</i>	61
5.2.1	<i>Codecs</i> x Atraso	61
5.2.2	<i>Codecs</i> x Perda de Pacote	62
5.2.3	<i>Codecs</i> x VIFP	63
5.2.4	<i>Codecs</i> x PSNR	65
5.2.5	<i>Codecs</i> x SSIM	66

5.2.6	<i>Codecs</i> x Quantidade de Movimento x Atraso	67
5.2.7	<i>Codecs</i> x Quantidade de Movimento x Perda de Pacote	69
5.2.8	<i>Codecs</i> x Quantidade de Movimento x VIFP	70
5.2.9	<i>Codecs</i> x Quantidade de Movimento x PSNR	71
5.2.10	<i>Codecs</i> x Quantidade de Movimento x SSIM	73
5.3	Potência do Sinal	74
5.3.1	Potência do sinal x Atraso	74
5.3.2	Potência do sinal x Perda de Pacotes	75
5.4	Porcentagem de Ocupação	77
5.4.1	Porcentagem de Ocupação x Atraso	77
5.4.2	Porcentagem de Ocupação x Perda de Pacote	78
5.5	Quantidade de Movimento	79
5.5.1	Quantidade de Movimento x VIFP	79
5.5.2	Quantidade de Movimento x PSNR	81
5.5.3	Quantidade de Movimento x SSIM	82
5.6	Compressão dos Vídeos e Quantidade de Pacotes	84
6	Considerações Finais e Sugestões para Pesquisas Futuras	86
6.1	Considerações Finais	86
6.2	Sugestões para Pesquisas Futuras	87
A	Códigos para Extração de Características dos Vídeos	91
A.1	Cálculo do Contraste de Weber	91
A.2	Cálculo da Soma de Diferenças Absolutas	92
A.3	Cálculo do Brilho	93
A.4	Cálculo do Embaçamento	95
B	Análise de Normalidade	98
C	Testes Não-Paramétricos de Igualdade	104
D	Testes Não-Paramétricos de Alternativa Maior	109

Lista de Símbolos

MD - Movimento Devagar

MM - Movimento Moderado

MR - Movimento Rápido

PSNR - *Peak Signal-to-Noise Ratio* - Pico relação sinal ruído

VIFP - *Visual Information Fidelity in Pixel Domain*

SSIM - *Structural Similarity*

MB - *Megabyte*

KB - *Kilobyte*

dBm - *Decibel-milliwatts*

Mbits/s - *Megabit* por segundo

MBytes/s - *Megabyte* por segundo

MHz - *Megahertz*

GHz - *Gigahertz*

WEP - *Wired Equivalent Privacy*

WPA - *Wi-Fi Protected Access*

WPA2 - *Wi-Fi Protected Access 2*

IEEE - *Institute of Electrical and Electronics Engineers*

DM - *Delta Modulation* - Modulação Delta

DPCM - *Differential pulse-code Modulation* - Modulação por Codificação Diferencial de Pulsos

Lista de Figuras

2.1	Lista dos canais e demonstração da sobreposição entre suas faixas de operação.	29
3.1	Soekris net5501 [21].	34
3.2	Ubiquiti eXtreme Range 2.	35
4.1	Diagrama de Execução de um Experimento e Análise de Resultados. . .	49
4.2	Planta baixa do ambiente de execução dos experimentos.	50
4.3	Lista das redes existentes no mesmo ambiente da rede de testes.	51
4.4	Resultado da clusterização hierárquica.	53
4.5	Resultado da clusterização hierárquica obtido pelos autores Khan, Sun e Ifeachor [18].	55
4.6	Diagrama de Classificação dos Vídeos.	55
4.7	Histograma da taxa de transferência obtida.	57
4.8	<i>Boxplot</i> da taxa de transferência obtida.	57
4.9	Diagrama de Extração da Porcentagem de Ocupação.	58
5.1	Gráfico de intervalos de confiança Atraso x <i>Codec</i>	62
5.2	Gráfico de intervalos de confiança Perda de Pacote x <i>Codec</i>	63
5.3	Gráfico de intervalos de confiança VIFP x <i>Codecs</i>	64
5.4	Gráfico de intervalos de confiança PSNR x <i>Codecs</i>	65
5.5	Gráfico de intervalos de confiança SSIM x <i>Codecs</i>	67
5.6	Gráfico de intervalos de confiança <i>Codecs</i> x Quantidade de Movimento x Atraso.	68

5.7	Gráfico de intervalos de confiança <i>Codecs</i> x Quantidade de Movimento x Perda de Pacote.	69
5.8	Gráfico de intervalos de confiança <i>Codecs</i> x Quantidade de Movimento x VIFP.	70
5.9	Gráfico de intervalos de confiança <i>Codecs</i> x Quantidade de Movimento x PSNR.	72
5.10	Gráfico de intervalos de confiança <i>Codecs</i> x Quantidade de Movimento x SSIM.	73
5.11	Gráfico de intervalos de confiança Atraso x Potência do sinal.	75
5.12	Gráfico de intervalos de confiança Perda de Pacote x Potência do sinal.	76
5.13	Gráfico de intervalos de confiança Atraso x Porcentagem de Ocupação.	77
5.14	Gráfico de intervalos de confiança Perda de Pacote x Porcentagem de Ocupação.	78
5.15	Gráfico de intervalos de confiança VIFP x Quantidade de Movimento. .	80
5.16	Gráfico de intervalos de confiança PSNR x Quantidade de Movimento.	81
5.17	Gráfico de intervalos de confiança SSIM x Quantidade de Movimento. .	83

Lista de Tabelas

4.1	Resultados obtidos da Rede Neural.	54
4.2	Resumo dos dados segundo a estatística descritiva.	56
5.1	Tamanho dos arquivos por <i>codec</i> utilizado.	84
5.2	Quantidade de pacotes por vídeo e <i>codec</i> utilizado.	85
B.1	Resultados dos testes de normalidade Atraso x <i>Codec</i>	98
B.2	Resultados dos testes de normalidade Atraso x Potência.	98
B.3	Resultados dos testes de normalidade Atraso x Porcentagem de Ocupação.	98
B.4	Resultados dos testes de normalidade Atraso x <i>Codec</i>	99
B.5	Resultados dos testes de normalidade Atraso x Potência.	99
B.6	Resultados dos testes de normalidade Perda de pacote x Porcentagem de Ocupação.	99
B.7	Resultados dos testes de normalidade VIFP x Quantidade de Movimento.	99
B.8	Resultados dos testes de normalidade VIFP x <i>Codec</i>	99
B.9	Resultados dos testes de normalidade PSNR x Quantidade de Movimento.	100
B.10	Resultados dos testes de normalidade PSNR x <i>Codec</i>	100
B.11	Resultados dos testes de normalidade SSIM x Quantidade de Movimento.	100
B.12	Resultados dos testes de normalidade SSIM x <i>Codec</i>	100
B.13	Resultados dos testes de normalidade Atraso x <i>Codec</i> x Quantidade de Movimento.	101
B.14	Resultados dos testes de normalidade Perda de Pacote x <i>Codec</i> x Quantidade de Movimento.	101
B.15	Resultados dos testes de normalidade VIFP x <i>Codec</i> x Quantidade de Movimento.	102

B.16 Resultados dos testes de normalidade PSNR x <i>Codec</i> x Quantidade de Movimento.	102
B.17 Resultados dos testes de normalidade SSIM x <i>Codec</i> x Quantidade de Movimento.	103
C.1 Resultados dos testes de comparação de igualdade SSIM x <i>Codec</i>	104
C.2 Resultados dos testes de comparação Atraso x Potência.	104
C.3 Resultados dos testes de comparação Atraso x Porcentagem de Ocupação.	104
C.4 Resultados dos testes de comparação Perda de Pacotes x <i>Codec</i>	105
C.5 Resultados dos testes de comparação Perda de Pacotes x Potência.	105
C.6 Resultados dos testes de comparação de igualdade Perda de Pacote x Porcentagem de Ocupação.	105
C.7 Resultados dos testes de comparação de igualdade VIFP x Quantidade de Movimento.	105
C.8 Resultados dos testes de comparação de igualdade VIFP x <i>Codec</i>	105
C.9 Resultados dos testes de comparação de igualdade PSNR x Quantidade de Movimento.	105
C.10 Resultados dos testes de comparação de igualdade PSNR x <i>Codec</i>	106
C.11 Resultados dos testes de comparação de igualdade SSIM x Quantidade de Movimento.	106
C.12 Resultados dos testes de comparação de igualdade Atraso x Quantidade de Movimento x <i>Codec</i>	106
C.13 Resultados dos testes de comparação de igualdade Perda de Pacote x Quantidade de Movimento x <i>Codec</i>	107
C.14 Resultados dos testes de comparação de igualdade SSIM x Quantidade de Movimento x <i>Codec</i>	107
C.15 Resultados dos testes de comparação de igualdade VIFP x Quantidade de Movimento x <i>Codec</i>	108
C.16 Resultados dos testes de comparação de igualdade PSNR x Quantidade de Movimento x <i>Codec</i>	108
D.1 Resultados dos testes de comparação Atraso x <i>Codec</i>	109

D.2 Resultados dos testes de comparação Perda de Pacote x Porcentagem de Ocupação.	109
D.3 Resultados dos testes de comparação VIFP x Quantidade de Movimento.	109
D.4 Resultados dos testes de comparação VIFP x <i>Codec</i>	110
D.5 Resultados dos testes de comparação PSNR x Quantidade de Movimento.	110
D.6 Resultados dos testes de comparação PSNR x <i>Codec</i>	110
D.7 Resultados dos testes de comparação de igualdade SSIM x Quantidade de Movimento.	110
D.8 Resultados dos testes de comparação SSIM x <i>Codec</i>	110

Lista de Códigos Fonte

A.1	Código C++ para Cálculo do Contraste de Weber	91
A.2	Código C++ para Cálculo da Soma de Diferenças Absolutas	92
A.3	Código C++ para Cálculo do Brilho	93
A.4	Código C++ para Cálculo do Embaçamento	95

Capítulo 1

Introdução

Neste capítulo são introduzidos os conceitos e é feita uma contextualização sobre o assunto que será abordado neste trabalho. Além disto, são especificados os objetivos e a relevância deste trabalho.

Uma rede de computadores é uma coleção de computadores autônomos conectados por uma única tecnologia. Dois computadores estão conectados se são capazes de trocar informações. A conexão pode ser feita por meio de fibra óptica, microondas, infravermelho, comunicação via satélite, além do método mais comum: fio de cobre. Redes existem em vários tamanhos, formas e modelos. Geralmente, são conectadas em conjunto para formar redes maiores, com a Internet sendo o maior e mais conhecido exemplo de uma rede de redes [1]. Através da Internet, é possível acesso a diversos conteúdos como: notícias, redes sociais, mapas, buscas em geral, lojas *online*, transmissão de áudio e vídeo em tempo real, etc.

Um vídeo é uma sequência de imagens, como visto na televisão, e é essencialmente um sinal tridimensional, em que a intensidade em qualquer ponto da imagem é uma função de três variáveis: duas variáveis espaciais e o tempo [2]. A principal aplicação do vídeo surgiu da televisão com as suas inúmeras aplicações (entretenimento, educação, engenharia, ciência, defesa, entre outras). O vídeo exibido em computadores é um vídeo digital comprimido, utilizando algum formato específico, para reduzir a quantidade de dados necessária para representá-lo. Na televisão, um vídeo é transmitido através de ondas eletromagnéticas até a antena da residência de um usuário, por um canal dedicado e por isso, mais confiável. Quando assistimos a um vídeo pela Internet, ele

está sendo transmitido (graças à tecnologia de *streaming*) por uma rede compartilhada com milhões de usuários e, por isso, se torna menos confiável.

Vídeo *stream* se refere à transmissão em tempo real de um vídeo armazenado. Implica que não é necessário fazer o *download* completo do conteúdo do vídeo, mas é reproduzido enquanto partes do conteúdo são recebidas e decodificadas [3], ou seja, o usuário pode assistir ao vídeo enquanto ele é transmitido, desde que os pacotes cheguem a tempo para serem exibidos. Quando fala-se em transmissão em tempo real, temos uma necessidade de garantia de entrega de pacotes ordenados no tempo certo, para que o vídeo possa ser exibido em sua qualidade máxima. Esta garantia é difícil de conseguir em uma rede como a Internet e mais difícil ainda de se alcançar em redes sem fio com muitos usuários.

Redes sem fio tendem a sofrer mais interferência de fontes externas pois utilizam o ar como meio de propagação, ou seja, a rede compete com quaisquer redes na mesma frequência. Mesmo eletrodomésticos que podem ser encontrados em residências (microondas, telefones sem fio, entre outros), além de sofrerem com atenuação e reflexão de sinal que ocorre naturalmente em qualquer ambiente e mais intensamente ambientes fechados e/ou com muitos obstáculos (paredes de alvenaria, armários, objetos em geral).

Cada vez mais usuários utilizam sistemas para *streaming* de vídeos como *Netflix*, *Hulu*, *YouTube*. Estes sistemas fazem o *stream* via Internet (que em sua maior parte é cabeada). É bastante comum a presença de roteadores *WiFi* nas casas dos usuários, principalmente pela facilidade e mobilidade que este tipo de rede nos oferece para conectar os mais diversos dispositivos (*notebooks*, *tablets*, *smartphones*, *etc.*). A qualidade dos vídeos pode ser alterada ao se tráfegar numa rede sem fio.

1.1 Objetivos e Relevância

1.1.1 Objetivos

Objetivo Geral

Dado que várias famílias possuem redes sem fio em suas casas e que, provavelmente, as possuem para dar conectividade e mobilidade aos seus dispositivos (*notebooks, tablets, smartphones, etc.*), o principal objetivo deste trabalho é avaliar a qualidade do vídeo recebido em um *stream*, ou seja, avaliar se houve, ou não, perda de qualidade durante a sua transmissão por uma rede sem fio.

Objetivos Específicos

Após a verificação de qualidade, serão consideradas também outras questões:

- Verificar se o conteúdo dos vídeos (cenas de ação, cenas estáticas) possui alguma influência na qualidade desses. Euisto pode ocorrer pois vídeos de ação e coloridos necessitam de uma taxa de transmissão maior para manter a qualidade e este pode ser um fator importante para a qualidade geral do vídeo. Essa necessidade de uma taxa de transmissão maior pode ser devido à mudança nos *frames*. Se um frame é diferente do outro, o sistema é obrigado a fazer um *download* de uma quantidade maior de dados do frame.
- Será avaliado também se o formato de compressão usado afetou a qualidade do vídeo recebido, ou seja, se algum formato implica uma melhor qualidade durante um *stream* de vídeo. Isto pode ocorrer devido ao fato de alguns formatos de compressão exigirem uma taxa de transferência maior, técnicas mais avançadas de codificação e decodificação, entre várias outras possibilidades para manter a qualidade do vídeo recebido.

1.1.2 Relevância

A relevância deste trabalho se encontra no fato de o *stream* de vídeo ser cada vez mais utilizado por meio da *internet* ao redor do mundo.

Por meio dos resultados obtidos, será possível saber se existe um formato de compressão (*codec*) mais adequado para transmitir um vídeo através de *streams* e, dado que exista esse formato, então pode ser indicado para uso e até melhorado para suportar situações mais extremas de congestionamento de rede.

Além disso, dado que exista perda de qualidade, é possível pensar em aplicações que controlem o fluxo de envio do vídeo de acordo com o estado da rede, ou seja, um sistema pode monitorar a rede (para descobrir possíveis congestionamentos) para que técnicas de controle de fluxo possam ser aplicadas, para manter a qualidade do vídeo recebido. Exemplos dessas técnicas são, diminuição da qualidade do vídeo de acordo com a quantidade de movimento em determinadas cenas para que o vídeo não seja interrompido, tentar fazer a reserva de largura de banda necessária de acordo com a quantidade de movimento do vídeo ou utilizar os campos de cabeçalho do protocolo IP que são responsáveis pela priorização de pacotes na internet.

Capítulo 2

Fundamentação Teórica e Revisão da Literatura

Neste capítulo, a seção 2.1 descreve os conceitos utilizados na elaboração deste trabalho e a seção 2.2 apresenta os trabalhos relacionados a este.

2.1 Fundamentação Teórica

2.1.1 Vídeo e Vídeo *Stream*

Um vídeo é uma sequência de imagens. É um sinal tridimensional, em que a intensidade em qualquer ponto da imagem é uma função de três variáveis: duas variáveis espaciais e o tempo [2]. Sua principal aplicação nasceu com a televisão. O vídeo que é exibido em computadores é digital e comprimido utilizando algum *codec*, para reduzir a quantidade de informação necessária para representá-lo.

Na televisão, o vídeo é transmitido através de ondas eletromagnéticas por um canal dedicado e, portanto, mais confiável. Quando um vídeo é assistido pela Internet, ele está sendo transmitido por uma rede compartilhada com milhões de usuários e, por isso, menos confiável com relação à entrega dos dados. A transmissão pela Internet se dá devido a criação de uma tecnologia chamada *streaming*.

Vídeo *stream* diz respeito à transmissão de um vídeo armazenado, em tempo real. Implica que não é necessário possuir todos os dados para exibir o vídeo completo, ou

seja, ou vídeo é recebido enquanto é exibido. [3]. Essa transmissão em tempo real, exige que os pacotes com os dados dos vídeos sejam entregues no tempo correto e em ordem para que os vídeos sejam exibidos em sua qualidade máxima. Esse tipo de garantia é difícil de conseguir em uma rede como a Internet.

2.1.2 Características

Um vídeo possui várias características, temporais e espaciais. Elas ajudam a classificar um vídeo de acordo com o seu conteúdo.

Como característica temporal para a pesquisa ora escrita, foi determinada a *Sum of Absolute Differences - SAD* (Soma de diferenças absolutas), por ser um indicador da quantidade de movimento no vídeo, já que calcula a diferença entre os *pixels* de um dado quadro e seu subsequente. Quanto maior a quantidade de movimento, maior a diferença entre os quadros e, portanto, maior a quantidade de informações enviadas. A soma de diferenças absolutas pode ser calculada por meio da Equação 2.1 [4].

$$SAD_{n,m} = \sum_{i=1}^n \sum_{j=1}^m |B_n(i, j) - B_m(i, j)|. \quad (2.1)$$

Em que B_n e B_m são dois *frames* de tamanho $n \times m$ e i e j , denotam as coordenadas dos *pixels*. O código fonte do sistema utilizado para cálculo desta característica é apresentado no Apêndice A.2.

As características espaciais determinadas, para a pesquisa ora escrita, foram:

- **Brilho:** O brilho de um vídeo é uma medida de qualidade de imagem visto que imagens tendem a ser difíceis de serem visualizadas. Foi escolhido como característica a ser identificada pelo fato de que quanto mais brilho uma imagem apresenta, mais dados ela precisa para ser representada, e isto pode influenciar a qualidade do vídeo recebido já que demanda mais recursos da rede pela quantidade maior de dados. Calculado por meio do módulo da diferença entre a média do brilho do *frame* anterior e o posterior. O código fonte do sistema utilizado para determinação dessa característica é apresentado no Apêndice A.3.;
- **Contraste:** O contraste de um vídeo é a diferença de aparência entre duas ou mais de suas partes. É uma medida de qualidade de imagem, visto que em ima-

gens com pouco contraste existe uma dificuldade de distinção dos objetos que a imagem contém. Pode influenciar na qualidade do vídeo recebido à medida que, dependendo do contraste exibido na imagem, ela pode ser facilmente descodificada já que alguns *codecs* se utilizam de algoritmos de inteligência artificial para inferir partes das imagens que não foram recebidas. Calculado como o contraste de Weber [5] com a Equação 2.2.

$$\text{Contraste} = \frac{I - I_b}{I_b}. \quad (2.2)$$

Em que I e I_B , representam a luminância dos objetos e do fundo da imagem, respectivamente. O código fonte do sistema utilizado para cálculo desta característica pode ser visto no Apêndice A.1;

- **Borramento:** Representa quão definidos os objetos de uma imagem estão, ou seja, quanto maior a definição maior é a quantidade de informação necessária para representar a mesma imagem e isto pode influenciar a qualidade do vídeo recebido pois é necessário mais dados para representar a mesma imagem com menos borramento. É calculado utilizando o algoritmo *Canny Edge* [6] para detecção de bordas e a transformada de Hough [7] para extração das linhas obtidas pelo *Canny*. O código fonte do sistema utilizado para cálculo desta característica é apresentado no Apêndice A.4.

2.1.3 Padrões de Compressão

MPEG-4

É um método para definir a compressão de áudio e vídeo digitais. Finalidades do MPEG-4 incluem a compressão de áudio e vídeo para a *web (streaming media)*, CDs e televisão. Absorve muitas das funcionalidades do MPEG-1, MPEG-2 e outros padrões relacionados, adicionando novas funcionalidades como suporte a VRML para renderização 3D, arquivos compostos orientados a objetos, AAC (*Advanced Audio Coding*), dentre outras funcionalidades.

É um padrão em constante evolução e está dividido em um conjunto de partes. As partes mais importantes do MPEG-4 são: Parte 2 (incluindo o *Advanced Simple*

Profile, utilizado por codificadores como DivX, Xvid, Nero Digital and Quicktime 6) e o MPEG-4 parte 10 que é o padrão de compressão conhecido como H.264 [8].

H.264

H.264 ou MPEG-4 Part 10, *Advanced Video Coding* (MPEG-4 AVC), é um padrão de compressão de vídeo que é um dos mais utilizados atualmente. H.264 é um padrão *motion-compensation-based* orientado a blocos, desenvolvido pelo ITU-T Video Coding Experts Group (VCEG) em conjunto com ISO/IEC JTC1 Moving Picture Experts Group (MPEG). A compressão realizada pelo H.264 apresenta perdas, ou seja, degrada a imagem original.

Define macroblocos de até 16×16 *pixels*. Pode utilizar múltiplos vetores de movimento por macrobloco. Predição espacial a partir das bordas dos blocos vizinhos para codificação dentro de um mesmo quadro, dentre várias outras características que o fazem ser superior ao MPEG-4 [9].

HEVC

HEVC, também conhecido como H.265 é um padrão de compressão novo desenvolvido pelo *Joint Collaborative Team on Video Coding* (JCT-VC). Foi desenvolvido com o objetivo de obter duas vezes mais compressão que o padrão predecessor (H.264/AVC). Com um mesmo nível de qualidade visual, o HEVC permite que o vídeo seja comprimido em um arquivo que tem metade do tamanho de um vídeo que foi comprimido utilizando o H.264.

Muito do poder dos padrões de compressão vem de uma técnica conhecida como *motion compensated prediction*, onde blocos de *pixels* são codificados por meio de uma referência para outra área do mesmo quadro ou de outro quadro. O H.264 define macroblocos de até 16×16 *pixels* e o HEVC define blocos de até 64×64 *pixels*.

O HEVC tem capacidade de codificar vetores de movimento com uma precisão maior, resultando em um bloco com menor erro residual. Além disso, inclui um novo método para melhorar a predição dentro de um mesmo quadro: *Adaptive Motion Vector Prediction* [10].

2.1.4 Métricas

A medição de qualidade de vídeo objetiva pode ser dividida em duas categorias. A primeira inclui a medição baseada na diferença como as métricas *Mean Squared Error (MSE)* e o *Peak Signal-to-Noise Ratio (PSNR)*; a segunda é baseada no sistema visual humano (*Human Visual System (HVS)*) que inclui métricas para avaliação da similaridade estrutural como a *Structural Similarity (SSIM)*, *Visual Information Fidelity in Pixel Domain (VIFP)*, *Structure & Hue Similarity (SHSIM)* e medições de similaridade de bordas. Os métodos baseados no sistema visual humano levam vantagem pelas características que se sabe sobre ele, e medem a qualidade de imagem estimando erros percebidos [11]. Neste trabalho, foram utilizadas métricas das duas categorias.

Como métricas de avaliação de qualidade de imagem, é possível citar:

- ***Peak Signal-to-Noise Ratio (PSNR)***: Métrica bastante utilizada pela comunidade. Define a relação entre a máxima energia possível de um sinal e o ruído que afeta a representação do sinal entre os quadros do vídeo original e degradado [12].

$$PSNR = 10 \log_{10} \frac{L^2}{\frac{1}{MNT} \sum_{m=1}^M \sum_{n=1}^N \sum_{t=1}^T [I(m, n, t) - \hat{I}(m, n, t)]^2}. \quad (2.3)$$

Na Equação 2.3, L é a faixa dinâmica dos valores de pixel, M e N são a largura e a altura das sequências de vídeo original e vídeo degradado respectivamente, T é o número de quadros que contém as sequências, $I(m, n, t)$ e $\hat{I}(m, n, t)$ representam o pixel na posição (m, n) do quadro t para a sequência original e degradada, respectivamente. A métrica varia de 0 a 100, 100 sendo o melhor valor possível[12];

- ***Structural Similarity (SSIM)***: Métrica que compara os quadros do vídeo degradado e do vídeo original para quantificar a degradação sofrida. Baseia-se na idéia que os *pixels* de uma imagem apresentam forte dependência. SSIM estima a similaridade comparando a luminosidade $l(x, y)$, o contraste $c(x, y)$ e a estrutura $s(x, y)$ do vídeo original x e do vídeo degradado y . Varia de 0 a 1, sendo 1 o melhor valor possível[12];

- **Visual Information Fidelity in Pixel Domain (VIFP):** Explora um critério de informação teórico para fidelidade de imagem usando *Natural Scene Statistics (NSS)*. Uma revisão de modelos NSS pode ser encontrada em [13]. Imagens e vídeos gerados a partir do ambiente visual vem de uma classe em comum: cenas naturais. Cenas naturais vem de um subespaço de todos os sinais possíveis, e os pesquisadores desenvolveram modelos sofisticados para caracterizar estas estatísticas. No mundo real, distorções são aplicadas às imagens (no caso deste trabalho a transmissão por uma rede sem fio.) que distorcem estas estatísticas e tornam os sinais das imagens não naturais. VIFP é uma métrica baseada na quantificação do conteúdo da imagem não distorcida para avaliar a informação perdida relativa por meio de um critério de fidelidade de informação definido a partir do NSS, distorção e *Human Visual System (HVS)*[14].

2.1.5 Compressão de Vídeo

O termo *compressão de dados* refere-se ao processo de redução do montante de dados exigidos para representar uma dada quantidade de informação, em que dados são os meios pelos quais uma informação é transmitida. Em uma conversa, por exemplo, uma pessoa pode contar uma história utilizando mais ou menos palavras. Neste caso, a informação que interessa é a história e as palavras são os dados utilizados para relatar as informações. Se duas pessoas utilizam quantidades diferentes de palavras, tem-se duas versões e pelo menos uma delas inclui dados não essenciais que podem não oferecer informações relevantes ou, simplesmente, reafirmar o que já é sabido, o que se chama de "redundância de dados". A redundância é um aspecto essencial na compressão de imagens digitais e três tipos de redundância básicas podem ser identificados: de codificação, interpixel e psicovisual [15].

Um sistema de compressão é composto de 4 sub-blocos [15]:

- Codificador de Fonte: remove as redundâncias percebidas;
- Codificador de Canal: aumenta a imunidade a ruídos do sinal produzido à saída do codificador de fonte através de inserção de uma forma controlada de redundância;

- Decodificador de Canal: verifica a paridade da palavracódigo recebida, para saber se ela é coerente com a convenção de paridade previamente estabelecida;
- Decodificador de Fonte: realiza, em sequência oposta, as operações inversas dos blocos mapeadores e codificadores de símbolo do codificador de fonte;

Compressão sem Perdas

Compressão onde não existe a perda de dados, também chamada de compactação.

Algumas técnicas podem ser citadas para este tipo de compressão:

- Código de Huffman - o primeiro passo no algoritmo de Huffman consiste na criação de uma série de reduções na fonte original, através da ordenação das probabilidades de ocorrência dos símbolos sob consideração, combinando os (dois) símbolos de menor probabilidade em um único símbolo que irá substituí-los na próxima etapa de redução da fonte. O segundo passo no algoritmo de Huffman consiste em codificar cada fonte reduzida, iniciando pela menor fonte e caminhando em direção à fonte original [15];
- Codificação Aritmética - é uma técnica orientada a bloco, na qual uma palavracódigo aritmética é atribuída a uma seqüência de símbolos de entrada. A palavra código em si define um intervalo de números reais entre 0 e 1. À medida que o número de símbolos na mensagem aumenta, o intervalo usado para representá-la se torna menor e o número de bits utilizados para representá-lo se torna maior [15];
- Codificação LZW - baseia-se na construção progressiva de uma tabela (dicionário) contendo as strings de símbolos de vários comprimentos encontradas na seqüência de símbolos da fonte. Como este dicionário é criado a partir da seqüência a ser codificada, ele reflete com precisão as estatísticas daquela fonte de informação sendo codificada [15];
- Codificação *bit-plane* - é baseado no conceito de decomposição de uma imagem de múltiplos tons de cinza em uma série de imagens binárias, comprimindo a seguir

cada uma delas utilizando um dos inúmeros métodos de compressão de imagens binárias [15].;

Compressão com Perdas

Baseia-se no conceito de comprometimento da precisão da imagem reconstituída em troca de uma maior compressão [15], ou seja, pode distorcer a imagem para que o tamanho final do arquivo seja menor.

Este tipo de compressão pode ser classificado em:

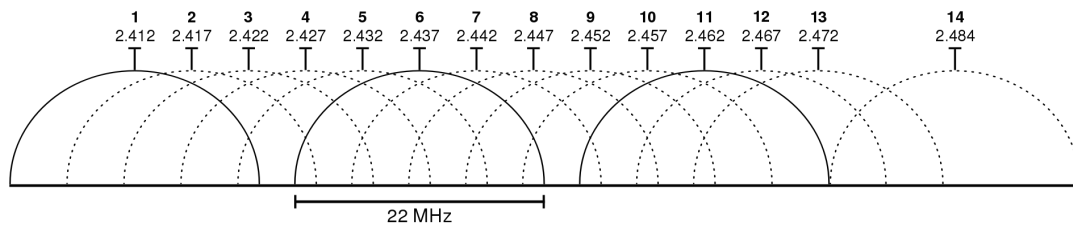
- Codificação preditiva com perdas - duas das técnicas de compressão preditiva com perdas mais conhecidas são a modulação delta (DM) e a Modulação por Codificação Diferencial de Pulsos (DPCM). Operam diretamente nos pixels de uma imagem [15];
- Codificação por transformadas - utiliza-se uma operação matemática linear reversível para mapear a imagem dentro de um conjunto de coeficientes, os quais em seguida são quantizados e codificados [15];

2.1.6 Comunicação Sem Fio

O padrão IEEE 802.11 define várias faixas de frequência para utilização de redes sem fio (2,4 GHz, 3,6 GHz, 4,9 GHz, 5 GHz, e 5,9GHz). A faixa utilizada foi a de 2,4GHz, no padrão IEEE 802.11g, que apresenta 14 canais (mas apenas 11 podem ser utilizados no Brasil). Cada canal possui uma largura de banda de 22 MHz para operar.

Uma rede sem fio é uma rede de computador que utiliza o ar como meio de propagação dos dados enviados e recebidos. O fato de usar o ar para transmitir seus dados faz com que a rede possa sofrer com algumas fontes de interferências que a rede cabeada não sofre. Uma fonte de interferência que pode afetar a rede é, justamente, a existência de outras redes no ambiente. Além da existência de outras redes sem fio, a mesma sofre também com a degradação do sinal à medida que a distância aumenta, equipamentos que utilizem a mesma frequência para transmitir e ainda absorção e refração de sinal em objetos presentes no ambiente.

Figura 2.1: Lista dos canais e demonstração da sobreposição entre suas faixas de operação.



A interferência que uma rede pode causar na outra acontece quando existe sobreposição entre as frequências de dois canais distintos, ou seja, um canal utiliza parte da banda de operação de outro. Apesar de a maioria dos canais se sobreporem pelo menos um outro canal, existem 3 que não se sobrepõem entre si. Os canais 1, 6 e 11 são livres de interferência de outras redes desde que existam apenas 3 redes no mesmo ambiente e cada uma utilize um dos três canais. As faixas de cada canal e suas sobreposições são mostradas na Figura 2.1.

Uma situação bastante comum em ambientes residenciais é a coexistência de várias redes sem fio. Isto ocorre pois, cada residência possui sua rede sem fio que pode interferir no sinal de outra, dependendo da escolha do canal de cada uma.

2.1.7 Conceitos de Inteligência Artificial

Redes Neurais

Uma rede neural é composta por um conjunto de nodos (unidades), conectados por *links*. Cada *links* tem um peso associado. Os pesos são os meios primários de armazenamento em redes neurais e a aprendizagem se dá pela atualização destes pesos. Alguns dos nodos podem estar conectados a um ambiente externo e podem ser designados como entrada ou saída da rede. Os pesos são atualizados na tentativa de fazer com que a relação entrada/saída esteja de acordo com as entradas providas pelo ambiente [16].

Existem diferentes tipos de redes neurais e uma delas é a *Multi-Layer Perceptron*. É um modelo de rede neural de alimentação *feedforward*, que mapeia conjuntos de dados de entrada para um conjunto de saídas apropriadas. Composto por várias camadas

de nodos em um grafo direcionado onde cada camada é totalmente conectada com a próxima. Já se encontra implementado no Weka [17] e foi utilizado com os parâmetros padrão do mesmo.

Clusterização

É uma técnica de aprendizagem não supervisionada. A clusterização começa com os dados. Após a entrada dos dados, o algoritmo tenta entender qual a distribuição de probabilidade deu origem a estes dados, para então separá-los em categorias ou *clusters* [16].

Uma forma comum de utilizar a clusterização é por meio da distância euclidiana, que é a distância entre dois pontos. A partir das distâncias entre os pontos, o algoritmo de clusterização separa os pontos de dados em categorias distintas.

A clusterização também já se encontra implementada no Weka [17] e foi utilizada com os parâmetros padrão do mesmo.

2.2 Revisão de Literatura

A revisão de literatura se encontra separada por tópico que se relaciona a este trabalho.

2.2.1 Avaliação de formatos de compressão e de conteúdo dos vídeos

É possível encontrar, na literatura, alguns trabalhos que se propõem a realizar a avaliação da qualidade de vídeos transmitidos via *streaming*. No trabalho desenvolvido por Chan, Zen e Mohapatra [4], é feita a avaliação de qualidade de vídeos no formato MPEG-4 em uma rede padrão IEEE 802.11e. Nesse trabalho foi utilizado o simulador NS-2 com uma abordagem experimental, com discussão em detalhes de como a satisfação humana sobre a qualidade do vídeo é afetada pelos parâmetros de qualidade de serviço introduzidos pelo padrão 802.11e.

Em Khan, Sun e Ifeachor [18] foi feita uma avaliação do conteúdo dos vídeos durante um *stream*. Para tanto, utilizaram técnicas de clusterização para separar os vídeos de

acordo com a extração de características temporais e espaciais do vídeo. Nesse trabalho, foi utilizado o formato de compressão MPEG-4. Como métricas de avaliação da qualidade dos vídeos foram utilizados *Peak-Signal-to-Noise-Ratio* (PSNR) e *Decodable frame rate* (Q). Além da avaliação de qualidade do vídeo, também foram investigados: o impacto da perda de pacotes no conteúdo dos vídeos para então, a determinação dos limiares (superior, inferior e médio), em que a percepção de qualidade de serviço do usuário é aceitável. É estudada também a taxa de transmissão mínima para atingir os requisitos de qualidade de serviço. No trabalho, foram utilizados 12 vídeos (também utilizados no presente trabalho.)

2.2.2 Métricas para avaliação de qualidade de vídeos

Na literatura, é possível encontrar trabalhos que têm o objetivo de estudar e propor novas métricas para avaliação de qualidade de vídeos. Em Gao, Cai e Bao [19] foram avaliadas as métricas utilizadas (objetivas e subjetivas). Além disso, novas métricas foram propostas. Estas métricas utilizam dados sobre a rede em conjunto com as métricas já citadas de avaliação de qualidade de vídeo.

Coaquira Begazo, Zegarra Rodriguez e Arjona Ramirez [12] avaliaram a perda na qualidade do serviço utilizando as métricas coletadas sobre a rede (atraso, perda de pacotes e variação do atraso). Nesse trabalho, foi montado um cenário isolado de emulação de rede com parametrização de diferentes condições da rede. Várias métricas objetivas, de qualidade de vídeo, são descritas tais como: (*Structural Similarity* ou SSIM, *Peak Signal-to-Noise Ratio* ou PSNR e *Video Quality Measure* ou (VQM). Os autores chegaram à conclusão de que a qualidade do vídeo é degradada ao transmiti-lo por uma rede IP e que esta degradação acontece, principalmente, pelo atraso de entrega dos pacotes.

Ao observar as métricas objetivas existentes, é possível verificar que todas levam em consideração a imagem em preto e branco (*grayscale*), que, normalmente, simplifica a detecção dos objetos, mas ignora as cores, que são descritores e influenciam o julgamento dos humanos. Levando isto em consideração, Niveditta Thakur e Swapna Devi [11], propõem uma nova métrica objetiva para aferição de qualidade de imagens que é a combinação de quatro fatores: similaridade de luminância, correlação de es-

trutura, similaridade de borda e similaridade de cor. Os(as) autores(as), fazem um *survey* sobre as métricas existentes classificando-as em duas categorias: baseadas na diferença (*Mean Squared Error* e *Peak Signal-to-Noise Ratio*) e baseadas no sistema visual humano (*Structural Similarity* e *Visual Information Fidelity in Pixel Domain*).

2.2.3 Metodologias de avaliação de qualidade de vídeos

Sobre metodologias de avaliação contínua da qualidade dos *streams*, é possível citar Kwagjin Choi *et al* [20], expondo que o serviço de vídeo *streaming* é muito utilizado sobre redes que utilizam o protocolo IP (*Internet Protocol*) sendo necessário o estabelecimento de metodologias para medição da qualidade. Os autores separam as metodologias em duas categorias: subjetiva e objetiva. As metodologias objetivas são subdivididas em: baseada na existência de vídeo referência e *measurement location*. Não existe menção sobre metodologias subjetivas de avaliação contínua.

Capítulo 3

Sistemas Utilizados

Neste capítulo são especificados os sistemas (*softwares*) e *hardwares*, utilizados no desenvolvimento do presente trabalho.

3.1 Hardware

Com o objetivo de criar um ambiente controlado e customizável, foi utilizada um *Single Board Computer* com uma placa Mini-PCI *Wireless* para exercer a função de roteador nos experimentos. Por meio desta configuração, é possível controlar a potência que está sendo utilizada para transmissão dos dados para simular o aumento ou diminuição de ruído no canal.

3.1.1 Soekris net5501

O Soekris net5501 [21] é um *Single Board Computer*, de baixo custo e baixo consumo de energia. Dotado de um processador de 500Mhz e 512MByte DDR de memória RAM. Pode utilizar um cartão de memória tipo *flash* para armazenamento do sistema operacional instalado e de todos os arquivos que devem ser armazenados. O Soekris net5501 pode ser vista na Figura 3.1.

A seguir, temos as especificações mais detalhadas da Soekris net5501[21]:

- Processador de chip único AMD Geode LX 500Mhz;
- 512 MByte DDR-SDRAM;

Figura 3.1: Soekris net5501 [21].



- 4 Mbit BIOS/BOOT Flash;
- Soquete CompactFLASH Type I/II;
- Interface UltraDMA-100 com conector de 44 pinos para 2.5" Hard Drive;
- Serial ATA 1.0 interface for Hard Drive;
- 4 portas Ethernet VIA VT6105M 10/100 Mbit, RJ-45;
- 2 portas serial;
- USB 2.0 interface, uma porta interna e outra externa;
- Socket Mini-PCI tipo III;
- PCI Slot;
- Monitor de temperatura e tensão;
- *Hardware watchdog*;
- Alimentação de energia externa de 6-25V DC, máximo de 20 Watt, protegido com TVS;
- Opção de suprimento de 5V utilizando conector interno;
- Temperatura de Operação: 0-60°C.

3.1.2 Ubiquiti eXtreme Range 2 Radio Transmissor

Para controlar a potência utilizada para transmissão de dados de um computador para outro, foi utilizada uma placa Mini-PCI Ubiquiti XR2[22] que possui uma potência de operação de até 28 dBm ou 600 mW, possibilitando a escolha da potência de transmissão, aumentando, assim, a possibilidade de variação nos experimentos realizados. A Ubiquiti eXtreme Range 2 pode ser vista na Figura 3.2.

Figura 3.2: Ubiquiti eXtreme Range 2.



A seguir, temos as especificações mais detalhadas da Ubiquiti eXtreme Range 2 Radio Transmissor[22]:

- Chipset: Atheros, 6th Generation, AR5414;
- Frequência de Operação: IEEE 802.11b/g, 2.4GHz;
- Interface: 32-bit mini-PCI Tipo IIIA;
- Voltagem de Operação: 3.3VDC;
- Antena: Única MMCX;
- Temperatura de Operação: -45C to +90C;
- Segurança: WPA, WPA2, AES-CCM & TKIP Encryption, 802.1x, 64/128/152bit WEP;
- Taxas de transmissão: 6 Mbits/s, 9 Mbits/s, 12 Mbits/s, 24 Mbits/s, 36 Mbits/s, 48 Mbits/s, 5Mbits/s;

- Larguras de banda suportadas: 5 MHz / 10 MHz / 20 MHz / 40 MHz.

3.2 Software

3.2.1 Voyage Linux

O Voyage Linux[23], é uma distribuição baseada no Debian que é desenvolvida para ser executada em plataformas embarcadas x86 como ALIX/WRAP e Soekris.

É possível executá-lo também em plataformas PC x86 *low-end*. Uma instalação do Voyage ocupa, normalmente, 256 MB de espaço de disco deixando o restante do espaço disponível para instalação de pacotes necessários de acordo com a utilidade escolhida pelo usuário.

Algumas das funcionalidades e características do Voyage são:

- Baseado no Debian;
- Muitas possibilidades de extensões já que utiliza um sistema de gerenciamento de pacotes;
- Linux 3.0/2.6 Kernel;
- Suporte WPA/WPA2 via hostapd e wpa_supplicant;
- *Drivers* de placas de rede sem fio pré-instalados.

Os usos mais comuns para o Voyage Linux são[23]:

- Ponto de acesso sem fio;
- Servidor VoIP;
- Rede sem fio *mesh*;
- *Music Player Daemon*.

A versão utilizada foi a 0.9.1

3.2.2 FFMpeg

FFMpeg[24] é o *framework* mais utilizado para decodificar, codificar, transcodificar, multiplexar, demultiplexar, *stream*, filtrar e reproduzir vários formatos de áudio e vídeo. Suporta dos mais antigos aos mais novos formatos de áudio de vídeo. É um *software* livre, mantido por uma comunidade que tenta prover as melhores soluções para desenvolvedores de aplicações e usuários finais. Para atingir este objetivo, utiliza um conjunto de bibliotecas gratuitas para realizar as operações sobre áudio e vídeo. Algumas delas são libavcodec, libavutil, libavfilter.

A versão utilizada na execução dos experimentos foi a 2.6 que é composta das seguintes versões de bibliotecas utilizadas:

- libavutil 54. 20.100;
- libavcodec 56. 26.100;
- libavformat 56. 25.101;
- libavdevice 56. 4.100;
- libavfilter 5. 11.102;
- libswscale 3. 1.101;
- libswresample 1. 1.100;
- libpostproc 53. 3.100.

3.2.3 Iperf3

Iperf[25], é uma ferramenta para medição da capacidade de transmissão em redes IP. Suporta o ajuste de vários parâmetros relacionado ao tempo, protocolo e *buffers*. É uma versão desenvolvida pelo NLANR/DAST há bastante tempo.

O Iperf3[26], como é chamado normalmente, é um *redesign* da versão original. Implementado desde o início com o objetivo de uma base de código menor e mais simples além de uma versão das funcionalidades em formato de biblioteca para que seja

usado em outros programas. Incorpora um certo número de funcionalidades existentes em outras ferramentas similares mas que não existiam no iperf original.

A versão utilizada foi a 3.0.11.

3.2.4 VQMT: Video Quality Measurement Tool

O VQMT [27] é uma ferramenta que provê o cálculo rápido de métricas objetivas relacionadas à qualidade de imagem de vídeos no formato YUV.

Nessa ferramenta, as métricas estão implementadas em C++, utilizando a biblioteca OpenCV[28], baseadas nas implementações originais em Matlab disponibilizadas pelos seus desenvolvedores. O código pode ser compilado em qualquer plataforma, desde que a biblioteca OpenCV esteja instalada.

As seguintes métricas estão disponíveis no VQMT:

- PSNR: Peak Signal-to-Noise Ratio;
- SSIM: Structural Similarity;
- MS-SSIM: Multi-Scale Structural Similarity;
- VIFp: Visual Information Fidelity, pixel domain version;
- PSNR-HVS: Peak Signal-to-Noise Ratio levando em consideração a *Contrast Sensitivity Function* (CSF);
- PSNR-HVS-M: Peak Signal-to-Noise Ratio levando em consideração a *Contrast Sensitivity Function* (CSF) e coeficiente de contraste de mascaramento de funções de base DCT.

3.2.5 Tcpdump

O Tcpdump[29] é uma ferramenta de análise de pacotes no nível de linha de comando. É responsável por capturar os pacotes que chegam ou saem de um determinado computador ou interface. É possível especificar *host* destino e porta por meio de uma expressão booleana.

Foi utilizado para capturar os pacotes, contendo dados de vídeos, enviados e recebidos para cálculo de atraso e perda de pacotes. A versão utilizada foi a 4.3.0.

3.2.6 OpenCV

A OpenCV[28] é uma biblioteca *open-source* lançada sobre a licença BSD. Possui algoritmos relacionados à computação visual. É escrita utilizando C/C++ otimizado. É mantida por uma comunidade de mais de 47 mil pessoas.

A versão utilizada foi a 2.4.

3.2.7 Weka

A Weka[17], é uma ferramenta desenvolvida e mantida pela Universidade de Waikato. Trata-se de uma coleção de algoritmos de aprendizagem de máquina e de tarefas de mineração de dados. Pode ser utilizada com interface própria ou como biblioteca dentro de códigos Java.

3.2.8 R

O R é tanto uma linguagem, quanto um ambiente desenvolvido para análise estatística e gráfica. É similar a linguagem e ambiente S desenvolvido pelos laboratórios Bell.

Provê uma variedade de técnicas estatísticas (modelagem linear e não-linear, testes estatísticos clássicos, etc.) e gráficas, além de ser extensível. O R tem a capacidade de geração de gráficos e plotagens desenhados, incluindo símbolos matemáticos e fórmulas.

Capítulo 4

Planejamento dos Experimentos

Neste capítulo, são expostos todos os aspectos dos experimentos executados. Além disto, são mostrados: o ambiente de execução e experimentos preliminares para definição de variáveis independentes.

4.1 Seleção do Contexto

O contexto a ser analisado pelos experimentos que estão descritos nas sub-seções seguintes é: transmissão de vídeos via *stream*, de um computador para outro, em ambiente residencial, para avaliar a qualidade do vídeo recebido e sua relação com a rede utilizada para transmissão.

4.2 Hipóteses

Baseado nos objetivos definidos na seção 1.1, foram formuladas as seguintes hipóteses que irão guiar o planejamento dos experimentos.

4.2.1 Avaliação da perda de qualidade

Para avaliar a perda de qualidade, foram formuladas as seguintes hipóteses:

Hipótese 1 *Os vídeos recebidos possuem a mesma qualidade dos enviados;*

Hipótese 2 *Os vídeos recebidos não possuem a mesma qualidade dos enviados.*

4.2.2 Avaliação do conteúdo

Para avaliar o conteúdo, foram formuladas as seguintes hipóteses:

Hipótese 3 *O conteúdo dos vídeos não influenciaram na qualidade do vídeo recebido;*

Hipótese 4 *O conteúdo dos vídeos influenciaram na qualidade do vídeo recebido.*

4.2.3 Avaliação dos *codecs*

Para avaliar os *codecs*, foram formuladas as seguintes hipóteses:

Hipótese 5 *Os diferentes codecs possuem a mesma qualidade de imagem ao enviar via stream;*

Hipótese 6 *Os diferentes codecs não possuem a mesma qualidade de imagem ao enviar via stream;*

4.3 Variáveis Independentes

Variáveis independentes são aquelas que podem ser controladas e que possuem algum efeito (positivo ou negativo) nas variáveis dependentes[30]. Ou seja, são fatores do ambiente que desejamos controlar para verificar seus efeitos sobre as variáveis dependentes. A escolha destas inclui a definição de escalas, faixas de valores utilizados e os níveis nos quais os testes são realizados.

Como o objetivo deste trabalho é estudar o comportamento da qualidade de vídeos recebidos através de uma rede sem fio, faz-se necessário o controle dos parâmetros da rede, por exemplo: potência do sinal, canal de transmissão e taxa de utilização da banda. Além disso, como, este trabalho, lida com vídeos temos também, que controlar o formato de compressão (*codec*), utilizado na codificação dos vídeos para a transmissão.

4.3.1 Variáveis Independentes Relacionadas à Rede

Com tudo o que foi dito, chegamos as seguintes variáveis independentes, relacionadas à rede, para os experimentos:

- **Potência do sinal:** É importante controlar a potência do sinal da rede sem fio pois ele pode interferir diretamente na qualidade do sinal recebido pelos computadores. Utilizamos a notação em *Decibel-Milliwatts* (dBm) para definição das potências;
Níveis: 1 dBm, 7 dBm e 14 dBm;
- **Canal de transmissão:** O canal pode ser importante se o mesmo já se encontrar bastante saturado (ou não) no ambiente em que as transmissões serão realizadas. Utilizamos apenas um canal no qual já existiam algumas redes, para simular um cenário bastante comum;
Nível: Canal 1 (2.401MHz a 2.423MHz);
- **Taxa de utilização da banda:** Como estamos lidando com experimentos baseados no ambiente residencial, é possível que durante a transmissão de um *stream* de vídeo, outras pessoas da residência também estejam utilizando a mesma rede (adicionando uma carga a ela) e, por isso, torna-se necessário verificar a qualidade do vídeo e da rede durante transmissões simultâneas com cargas diferentes. Para essa variável, testamos 4 níveis, sendo o nível de 0% o *baseline*, com a rede livre para transmissão dos vídeos.
Níveis: 0%, 30%, 60% e 90%.

4.3.2 Variável Independente Relacionada aos Vídeos

Definidas as variáveis relacionadas à rede, é necessário definir as variáveis independentes relacionadas aos vídeos transmitidos. Como um dos objetivos deste trabalho é avaliar a qualidade dos vídeos recebidos com diferentes *codecs*, foram utilizados 3 padrões de compressão para avaliação: H.264, MPEG-4 e HEVC (também conhecido como H.265). Os padrões de compressão podem afetar a transmissão de acordo com a necessidade de uma *bitrate* maior ou menor para exibição de um vídeo. Exemplo: Se um vídeo precisa de um *bitrate* de 400 *Kbit/s* mas a rede só oferece 200 *Kbit/s* de taxa de transmissão, este vídeo provavelmente terá sua qualidade deteriorada no computador que o receber.

Outro fator relacionado ao *codec* é a compressão realizada por ele, que modifica o tamanho do arquivo e assim requer mais ou menos pacotes para sua total recepção

no computador cliente. Quanto mais pacotes forem necessários, maiores as chances de aumentar o atraso e o pacote acabar sendo descartado, ou maiores as chances de que o pacote nem chegue ao destinatário por quaisquer razões inerentes à rede.

4.4 Variáveis Dependentes

São variáveis consideradas como respostas dos diferentes tratamentos obtidos com as variáveis independentes [30]. São estas que trazem as respostas para as nossas hipóteses.

Mais uma vez, devemos subdividir as nossas variáveis dependentes para facilitar o entendimento.

4.4.1 Variáveis Dependentes Relacionadas à Rede

Como estamos avaliando uma transmissão sobre uma rede de computadores, é interessante analisar os resultados obtidos através de variáveis relacionadas à rede em estudo. Assim, definimos as seguintes variáveis:

- **Perda de Pacote:** A perda pode ocorrer por diversos motivos: congestionamento, falta de sinal, entre outros. Impacta diretamente na qualidade do vídeo recebido, pois cada pacote contém uma parte do vídeo a ser exibido. Se esta parte é perdida, o formato de compressão, provavelmente, terá que se ajustar este cenário, cortando o vídeo ou exibindo quadros sem algumas partes. A perda de pacote é calculada a partir da Equação 4.1;

$$Perda = \frac{T_r - T_e}{T_e}. \quad (4.1)$$

Em que, T_r é o total de pacotes recebidos e T_e é o total de pacotes enviados.

- **Atraso:** O atraso na entrega de um pacote ocorre, principalmente, por congestionamento na rede. Impacta na qualidade do vídeo, pois se o pacote contendo certa parte do vídeo não chega a tempo de ser exibido, o formato de compressão, durante a decodificação, terá que se ajustar a este cenário cortando o vídeo ou exibindo quadros sem algumas partes. Medir o atraso de pacotes é uma tarefa que exige a sincronização dos relógios dos dois computadores. Para sincronizar os

relógios foi utilizado o protocolo NTP[31]. Para calcular o atraso da transferência de um único vídeo foi utilizada a Equação 4.2.

$$Atraso = \frac{\sum_{n=1}^N PE_n - PR_n}{T_e}. \quad (4.2)$$

Em que, PE_n é a *timestamp* de envio do pacote n , PR_n é a *timestamp* de recebimento do pacote n e T_e é o total de pacotes enviados.

4.4.2 Variáveis Dependentes Relacionadas aos Vídeos

Para avaliar a qualidade do vídeo recebido, em relação ao transmitido, utilizaremos métricas objetivas que são calculadas a partir dos dois vídeos, verificando se houve degradação da qualidade da imagem durante a transmissão.

Para avaliar os vídeos recebidos nas duas categorias citadas, escolhemos as seguintes métricas:

- **Peak Signal-to-Noise Ratio (PSNR):** Métrica bastante utilizada pela comunidade. Encontra-se definida na subseção 2.1.4.
- **Structural Similarity (SSIM):** Métrica que compara os quadros do vídeo degradado e do vídeo original para quantificar a degradação sofrida. Encontra-se descrita na subseção 2.1.4.
- **Visual Information Fidelity in Pixel Domain (VIFP):** Explora um critério de informação teórico para fidelidade de imagem usando *Natural Scene Statistics (NSS)*. Encontra-se detalhada na subseção 2.1.4.

4.5 Variável de Blocagem

São variáveis que apresentam ruídos nos dados gerais dos experimentos e, portanto, nos dados obtidos, fazendo com que os resultados não sejam verdadeiros [32].

No objeto de estudo deste trabalho pode-se controlar todas as variáveis menos o conteúdo do vídeo. É possível que o conteúdo do vídeo influencie os resultados, pois vídeos com mais movimento tendem a necessitar de uma maior *bitrate* para a sua

exibição com qualidade total. Assim, separamos os vídeos em três categorias, cuja classificação foi discutida na Seção 4.12, para que seja possível controlar a influência deste fator nos experimentos.

Os vídeos foram classificados em 3 categorias (níveis):

- Movimento Devagar (MD);
- Movimento Moderado (MM);
- Movimento Rápido (MR).

4.6 Seleção da Amostra de Vídeos

Foram utilizados 29 vídeos no formato mais comum de testes: 4:2:0 YUV. Os vídeos podem ser encontrados no site Video Coders[33] e em um site da Arizona State University[34].

Como já foi dito na seção anterior, os vídeos foram classificados em 3 categorias. A partir da classificação gerada foram escolhidos, aleatoriamente, 3 vídeos para serem utilizados em cada categoria. Os vídeos escolhidos, separados por categoria, foram:

- Movimento Devagar (MD): *Akiyo*, *Mother-daughter* e *Silent*;
- Movimento Moderado (MM): *Hall*, *News* e *Tempete* ;
- Movimento Rápido (MR): *Bus*, *Football* e *Stefan*;

4.7 *Design* dos Experimentos

Observando as hipóteses descritas no início deste capítulo, verificamos que precisamos de um *design* que nos proporcione a maior quantidade de dados possível com a menor quantidade de ruído. Também é necessário garantir que o *design* de experimento utilizado, seja replicável com facilidade.

Para tanto, utilizamos o *design* fatorial completo com blocagem (explicada em seção anterior). Além disso, o *design* proposto possui a característica de aleatoriedade, visto

que os experimentos são executados em ordem aleatória e que os vídeos foram escolhidos de forma aleatória, resultando em uma amostra independente. Outra característica importante, no *design* proposto, é o balanceamento, ou seja, todos os tratamentos possuem o mesmo número de resultados para avaliação.

Por meio da decisão sobre o *design* fatorial completo com blocagem, é possível chegar à quantidade total de experimentos realizados. Considerando-se que temos: 3 níveis para a variável potência do sinal, 4 níveis para a variável de taxa de utilização, 1 nível para a variável canal, 3 níveis para a variável *codec*, mais 3 níveis para a variável de blocagem quantidade de movimento, 3 vídeos em cada categoria da variável de blocagem e que cada tratamento foi executado 20 vezes, temos um total de 6.480 replicações dos experimentos.

4.8 Instrumentação

A instrumentação de um experimento não deve afetar os seus resultados e podem ser classificados em três tipos: objetos, orientações e medição[30].

4.8.1 Objetos

É importante garantir que os objetos utilizados no experimentos não contém falhas e que são apropriados para todos os experimentos. O fato de os vídeos serem utilizados por outros pesquisadores, como os da Universidade do Arizona[34] e do site Video Coders[33], torna os vídeos usados neste trabalho, apropriados para os experimentos.

Para verificar possíveis falhas nos mesmos, utilizamos a ferramenta FFMpeg[24].

4.8.2 Orientações

Consistem em orientações para execução dos experimentos ou treinamentos em métodos utilizados pelos participantes (caso envolva humanos). Esse caso, não envolve a participação de humanos, então tem-se apenas as orientações para execução dos mesmos.

Os experimentos foram executados sempre em um período de menor utilização de

outras redes (exemplo: madrugada), para minimizar os efeitos de interferência que outras redes possam ter na nossa. Sempre foi verificado se os experimentos haviam realmente acabado e se todos os scripts e sistemas utilizados na medição foram encerrados corretamente.

4.8.3 Medições

As medições em um experimento são feitas por meio de coleta de dados. No objeto de estudo deste trabalho, as medições consistem no monitoramento das redes para coleta de dados de atraso e perda de pacote, além da medição da qualidade dos vídeos recebidos de acordo com as métricas (variáveis dependentes) definidas na seção 4.4.2.

4.9 Validação

É importante considerar a validade ainda no planejamento dos experimentos para que ela seja garantida, e assim, que os resultados sejam válidos. A validade se refere a que os resultados devem ser válidos para a população de interesse. Primeiramente deve-se garantir a validade para a população estudada e depois tentar generalizar estes resultados para uma população maior.

Segundo [30], as ameaças à validade de um *design* de experimentos podem ser estudadas nas seguintes categorias: de conclusão, interna, de construção e externa. Nas sub-seções seguintes descrevemos as ameaças a este trabalho.

4.9.1 Ameaças à Conclusão

As ameaças à conclusão estão ligadas ao relacionamento entre o tratamento e o resultado. É importante garantir que exista uma relação estatística dada uma certa significância.

Para garantir a validade de conclusão devemos, então, utilizar os testes estatísticos adequados na análise de resultados, garantir que o tamanho da amostra é adequado e que os resultados sejam confiáveis, além de tomar todos os cuidados possíveis durante a implementação e a medição dos resultados de um experimento.

4.9.2 Ameaças Internas

Fatores de ameaças internas são aqueles relacionados à escolha de objetos de testes e como eles são separados em diferentes classes.

Para tanto, foram utilizados vídeos também usados por outros pesquisadores [33], [34] e foi utilizado um método de classificação (clusterização hierárquica) já utilizado em outro trabalho que foi citado na revisão de literatura do presente trabalho[18].

4.9.3 Ameaças de Construção

A validade de construção diz respeito à generalização dos resultados dos experimentos ao conceito ou teoria por trás dos experimentos.

Para evitar estas ameaças, todo o experimento foi explicado detalhadamente, incluindo como considerar um resultado melhor que o outro, sem deixar margem para dúvidas. Foram utilizados diferentes variáveis independentes e vários objetos de estudo foram incluídos para que não exista um fator que seja beneficiado com algum cenário específico e, assim, obtenha um resultado diferente do real.

4.9.4 Ameaças Externas

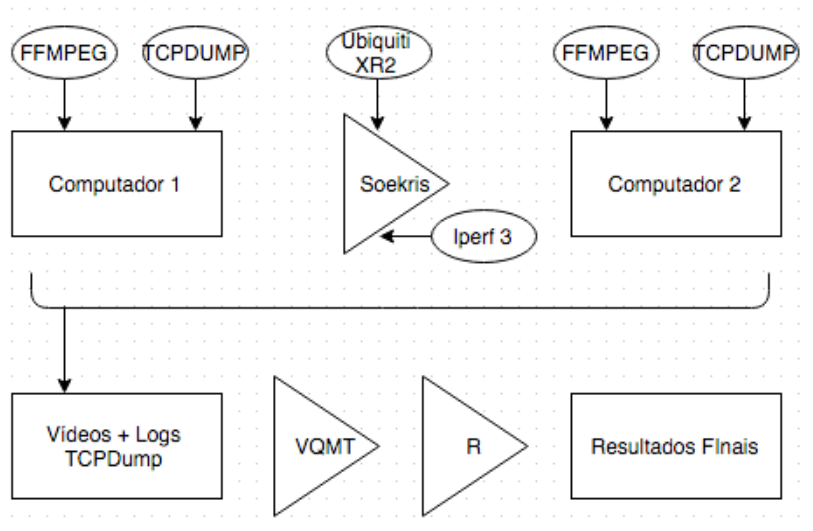
Existem 3 fatores que podem ser consideradas ameaças externas: escolha dos objetos de estudo de forma errada, conduzir os experimentos em um ambiente errado, realização os experimentos de modo que o tempo afete o resultados.

Para tanto, foram utilizados vídeos usados por outros pesquisadores [33], [34], já que os nossos experimentos são baseados em vídeos. O ambiente utilizado foi uma casa de construção bastante comum no Brasil, visto que foi este o ambiente de estudo definido. Todos os experimentos foram executados durante a madrugada para que a utilização das redes presentes no ambiente não afetasse os resultados obtidos por este trabalho.

4.10 Diagrama de Execução de Experimentos e Análise de Resultados

Para ilustrar a execução de um experimento, bem como apresentar a participação de cada sistema exposto no Capítulo 3, é apresentada a Figura 4.1.

Figura 4.1: Diagrama de Execução de um Experimento e Análise de Resultados.



Como é possível observar na Figura 4.1, o Computador 1 envia os vídeos utilizando o FFmpeg para a Soekris que deve entregar ao destinatário, neste caso, Computador 2. Toda essa operação é monitorada pelo TCPDump para capturar dados da rede em questão. Após a execução de todos os experimentos, os vídeos recebidos e os dados do TCPDump são analisados. Os vídeos tem suas métricas extraídas através do VQMT.

Após a extração das métricas, os dados são enviados ao R para a análise estatística e extração dos resultados finais.

4.11 Ambiente de Execução de Experimentos

4.11.1 Disposição do Ambiente

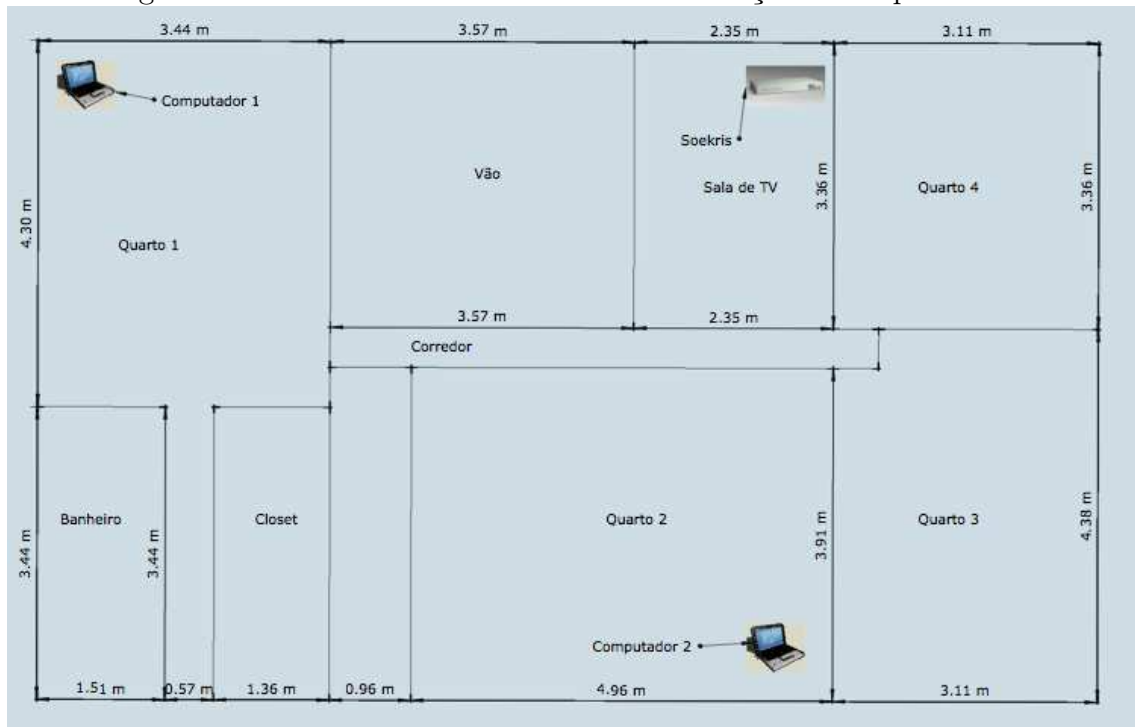
Como o objetivo é avaliar a qualidade dos vídeos recebidos em ambientes residenciais, foi usada uma casa como ambiente de testes.

A casa utilizada foi construída em alvenaria. Possui primeiro andar e cerca de

12 cômodos. A casa em alvenaria foi escolhida pois no Brasil, a maioria de casas e construções são feitas em alvenaria. Além disso, a alvenaria representa um obstáculo para propagação ondas eletromagnéticas.

Na Figura 4.2, é mostrada a planta baixa do primeiro andar da casa, local onde foram executados os experimentos.

Figura 4.2: Planta baixa do ambiente de execução dos experimentos.



Foram feitas transmissões entre dois *notebooks*: o primeiro localizado no quarto 1 e o segundo localizado no quarto 2, como dispostos na Figura 4.2. Os dois computadores estavam ligados à mesma rede sem fio (LATEC_EXP), configurada para usar o WPA2 como nível de segurança para garantir que nenhum outro computador utilizasse a rede.

4.11.2 Redes Sem Fio Presentes no Ambiente

Para ilustrar as redes presentes no ambiente de testes escolhido, tem-se a Figura 4.3. Nesta figura são mostrados os nomes das redes, canais que ocupam e qualidade do sinal (nível de ruído e nível de sinal) obtido no momento da coleta da informação.

Como é possível observar na Figura 4.3, temos algumas redes no mesmo canal. O canal 1 foi escolhido para simular uma situação bastante comum nos ambientes

Figura 4.3: Lista das redes existentes no mesmo ambiente da rede de testes.

Network Name	BSSID	Security	Protocol	RSSI	Noise	Channel	Band
Casa BALAO	80:3f:5d:87:dc:9c	WPA Personal	802.11b/g/n	-82	-87	7	2.4GHz
DIRECT-U8SCX-3400 Series ef728c	32:cd:a7:5d:30:42	WPA2 Personal	802.11g	-77	-92	11	2.4GHz
GVT-063C	d8:97:ba:da:8a:8c	WPA/WPA2 Personal	802.11n	-82	-92	11	2.4GHz
LATEC_EXP	d8:fe:e3:6a:06:3c	WPA/WPA2 Personal	802.11b/g/n	-40	-88	1	2.4GHz
Net Virtua 1043	00:15:6d:64:e6:db	WPA2 Personal	802.11b/g	-22	-88	1	2.4GHz
netvirtua casa1067	8c:04:ff:80:1a:5d	WPA/WPA2 Personal	802.11b/g	-86	-90	8	2.4GHz
NETWIFI	d8:97:ba:3b:a4:88	WPA/WPA2 Personal	802.11n	-87	-90	1	2.4GHz
	c8:3a:35:52:cd:fc	WPA Personal	802.11n	-82	-92	11	2.4GHz

residenciais: a sobreposição de redes no mesmo canal.

4.12 Classificação dos Vídeos

Diversas características podem ser destacadas quando se fala de vídeos. Resolução, *bit rate*, contraste e brilho são algumas delas, e fazem com que cada um deles demandem uma quantidade diferente de recursos da rede, e sofram de maneira diferente com as interferências que podem acontecer.

Levando-se em consideração essas características, podemos identificar grupos de vídeos, e através deles podemos tentar controlar a variável (como visto na seção 4), conteúdo dos vídeos.

Para chegar à classificação dos vídeos em categorias, foi feita, inicialmente, uma identificação de características temporais e espaciais dos vídeos. Com estes dados em mãos, foi utilizada uma técnica não-supervisionada para gerar uma classificação inicial e, em seguida, uma técnica supervisionada para verificar se os resultados obtidos eram aceitáveis. A classificação inicial (feita por meio da técnica de aprendizagem não-supervisionada) foi feita para servir como entrada para a classificação através da técnica de aprendizagem supervisionada.

4.12.1 Vídeos

Foram utilizados 29 vídeos no formato mais comum de testes: 4:2:0 YUV. Os vídeos podem ser encontrados no site Video Coders [33] e em um site da Arizona State University [34].

4.12.2 Extração de Características

Para fazer a classificação dos vídeos, precisamos identificar as suas características. Para realizar essa tarefa, foi feito uso da biblioteca OpenCV [28]. Foram identificadas características temporais e espaciais de cada vídeo.

As características escolhidas foram apresentadas e descritas descritas na subseção 2.1.2.

4.12.3 Classificação Prévia

Como não existia uma classificação prévia dos vídeos, fez-se necessária a separação dos vídeos em categorias para que os algoritmos supervisionados entrassem em ação e pudessem classificar os vídeos, validando a classificação original. Para separar os vídeos em categorias prévias, utilizamos uma clusterização hierárquica, por meio da distância euclidiana, presente no sistema Weka[17].

4.12.4 Classificador Utilizado

Para validar a classificação realizada pela clusterização hierárquica, foi utilizada uma técnica de aprendizagem supervisionada, tomando como entrada a saída da clusterização. Como técnica supervisionada, foi utilizada a rede neural *Multi Layer Perceptron* (MLP), também encontrada no Weka [17].

4.12.5 Categorias de Vídeos

Como categorias para classificação dos vídeos, escolhemos as seguintes:

- Movimento Devagar (MD);
- Movimento Moderado (MM);
- Movimento Rápido (MR).

4.12.6 Experimentos

Os dados obtidos através da extração das características dos vídeos foram submetidos

a uma clusterização hierárquica no Weka[17], para obter a classificação inicial.

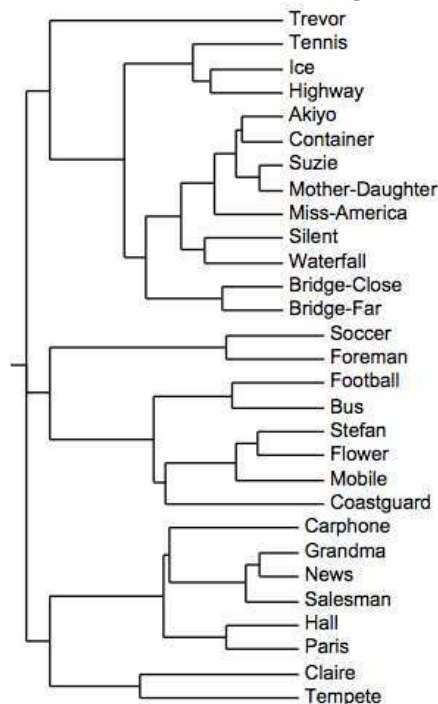
A rede neural foi executada em 10 *fold cross-validation* através do Weka[17]. Os dados foram separados em 66% para treino e 34% para teste. Com a validação cruzada, os dados são separados em conjunto de treinamento e de teste a cada nova validação realizada, não sendo utilizados os mesmos conjuntos sempre.

4.12.7 Apresentação e Análise de Resultados

No diagrama da Figura 4.4, é possível observar o resultado da clusterização hierárquica realizada nos dados utilizando a distância euclidiana.

É possível observar que os vídeos podem ser claramente separados em 3 grupos distintos (citados acima). No grupo intitulado Movimento Devagar, temos os vídeos: *Trevor*, *Tennis*, *Ice*, *Highway*, *Akiyo*, *Container*, *Suzie*, *Mother-Daughter*, *Miss-America*, *Silent*, *Waterfall*, *Bridge-Close* e *Bridge-Far*. No grupo de vídeos com Movimento Moderado, temos: *Carphone*, *Grandma*, *News*, *Salesman*, *Hall*, *Paris*, *Claire* e *Tempete*. Por fim, temos o grupo de Movimento Rápido com os vídeos: *Soccer*, *Foreman*, *Football*, *Bus*, *Stefan*, *Flower*, *Mobile*, *Coastguard*.

Figura 4.4: Resultado da clusterização hierárquica.



Ao utilizar os dados de saída da clusterização como entrada da rede neural, foram obtidos os resultados que se encontram na Tabela 4.1.

Tabela 4.1: Resultados obtidos da Rede Neural.

Instâncias Corretas	25 (86,2%)
Instâncias Erradas	4 (13,80%)
Estatística Kappa	0,78
Erro Médio Absoluto	0,11
Erro Médio Relativo	25,60%
Positivo Verdadeiro	0,86
Falso Positivo	0,08
Precisão	86,20%

A Tabela 4.1 mostra que, para os dados da rede neural, cerca de 86% dos vídeos foram corretamente classificados, ou seja, 86 vídeos em 100 teriam a sua classificação corretamente determinada a partir das características dos vídeos. A estatística Kappa mostra que há grande concordância entre o que se esperava da classificação dos vídeos e o que foi classificado corretamente. A estatística Erro Médio Relativo foi de aproximadamente 25%, mostrando que há conformidade entre o que foi classificado pelo modelo e o que realmente foi observado na análise. Fica constatada uma taxa adequada de falso positivo e de positivo verdadeiro e ainda uma precisão de 86,3%. Todos estes dados são indicadores de um modelo de classificação adequado.

Os resultados dessa classificação se assemelham aos obtidos por Khan, Sun e Ifeakor [18], que podem ser observados na Figura 4.5. É possível observar que os vídeos *Suzie*, *Akiyo* e *Grandma* são classificados como Pouco Movimento, os vídeos *Carphone*, *Foreman*, *Table-tennis* and *Rubgy* são classificados como Movimento Moderado e que os vídeos *Stefan* e *Football* são classificados como Muito Movimento.

4.13 Diagrama de Classificação dos Vídeos

Para ilustrar a classificação dos vídeos, bem como apresentar a participação de cada sistema exposto no Capítulo 3, é apresentada a Figura 4.6.

Figura 4.5: Resultado da clusterização hierárquica obtido pelos autores Khan, Sun e Ifeachor [18].

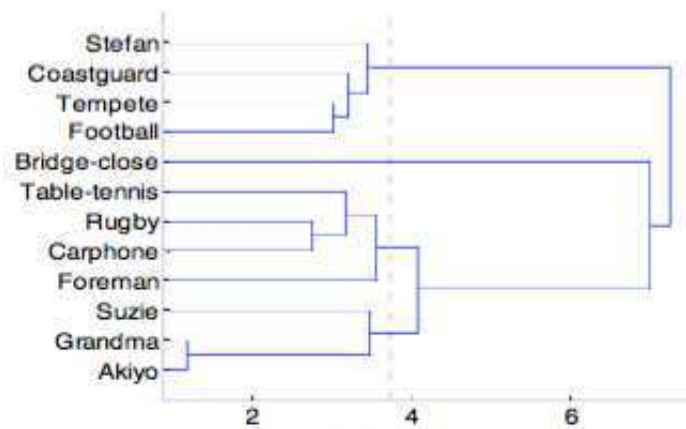
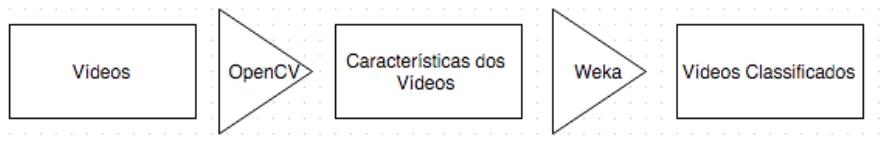


Figura 4.6: Diagrama de Classificação dos Vídeos.



Como é possível observar na Figura 4.6, os vídeos são enviados ao OpenCV para extração das características dos mesmos. A partir dessas características, é utilizado o Weka para fazer a classificação em 3 categorias diferentes

4.14 Análise da Taxa de Transmissão

Para chegarmos aos valores descritos na Seção 4, foi preciso fazer um conjunto de experimentos para determinar a taxa média de transferência da rede utilizada nos experimentos de transferência de vídeos. O ambiente de execução destes experimentos é o mesmo utilizado para a transferência dos vídeos e está descrito na Seção 4.11. Os experimentos para determinar a taxa média estão descritos nesta seção.

4.14.1 Experimentos

Os experimentos executados para determinar a taxa de transferência média são simples. Consistem na transferência do maior número de pacotes possível, de um computador para o outro, utilizando o protocolo TCP, através do Iperf3 e[26].

Foram executadas 100 repetições de tentativas de transferência na capacidade média. Os dados foram obtidos através da análise dos *logs* gerados pelo Iperf3.

4.14.2 Resultados Da Taxa de Transmissão

Através da análise dos *logs* gerados pelo Iperf3, foi possível obter os dados de taxa de transferência de cada repetição. Os dados foram analisados utilizando a ferramenta de análise estatística R. Os resultados foram medidos em *Mbits/s*.

A Tabela 4.2 resume os dados obtidos segundo a estatística descritiva.

Tabela 4.2: Resumo dos dados segundo a estatística descritiva.

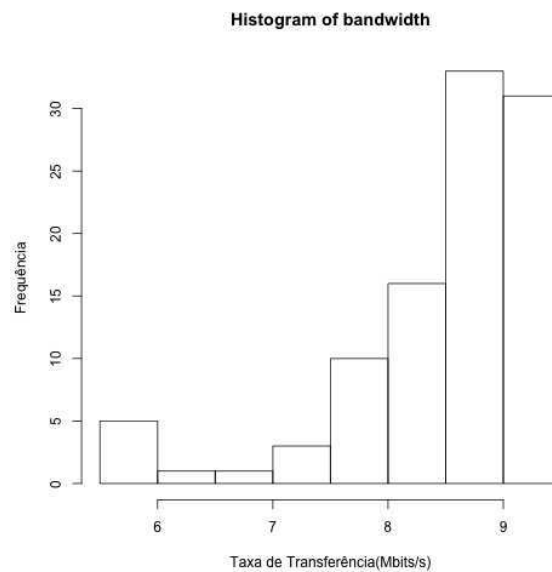
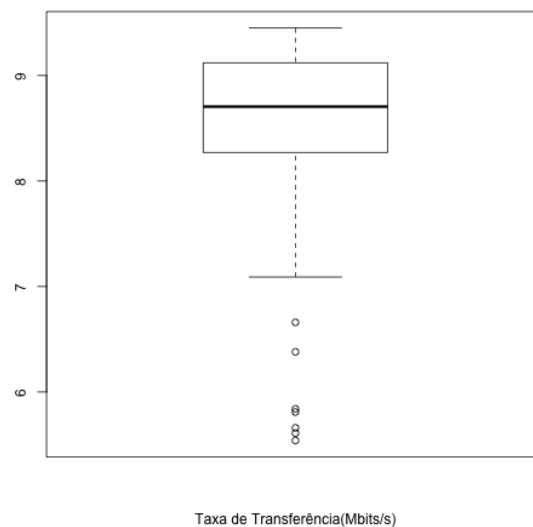
Média	8,49 <i>Mbits/s</i>
Mediana	8,70 <i>Mbits/s</i>
Mínimo	5,54 <i>Mbits/s</i>
Máximo	9,45 <i>Mbits/s</i>
Quartil 1	8,27 <i>Mbits/s</i>
Quartil 3	9,12 <i>Mbits/s</i>
Percentil 05	6,35 <i>Mbits/s</i>
Percentil 95	9,30 <i>Mbits/s</i>
Desvio Padrão	0,88 <i>Mbits/s</i>
Intervalo Inter-quartil	0,84 <i>Mbits/s</i>

Para que a média ou a mediana seja utilizada para calcular as porcentagens de utilização da taxa de transferência em cada experimento de envio de vídeos, devemos saber se os dados estão distribuídos de forma simétrica ou não. Para tanto, devemos observar o histograma e o *boxplot* gerados a partir dos dados das 100 repetições de experimento.

Na Figura 4.7 é apresentado o histograma da taxa de transferência atingida nos experimentos executados. Observa-se, neste gráfico, que não é possível verificar indícios de simetria nos resultados obtidos.

Na Figura 4.8 é mostrado o *boxplot* da taxa de transferência atingida nos experimentos executados. A partir deste gráfico, é possível observar que não existem indícios de simetria nas taxas de transferência coletadas por meio dos experimentos.

Figura 4.7: Histograma da taxa de transferência obtida.

Figura 4.8: *Boxplot* da taxa de transferência obtida.

Como os dados não mostram indícios de simetria, deve-se utilizar a mediana (8.705 Mbit/s, conforme a Tabela 4.2) como índice de tendência central e assim chegamos aos seguintes valores para as porcentagens de ocupação do canal de transmissão:

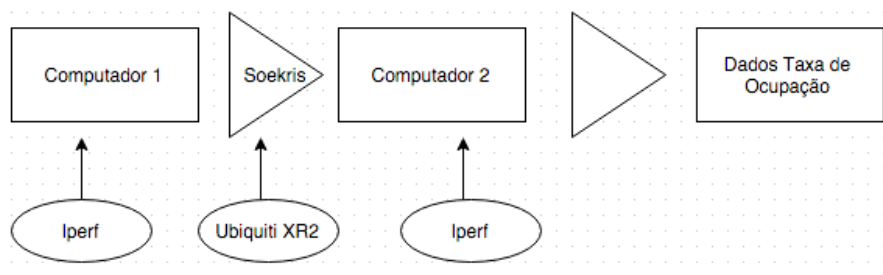
- $0\% = 8,705 \text{ Mbits/s} = 1,088125 \text{ MBytes/s}$
- $30\% = 8,705 \times 0,3 = 2,6115 \text{ Mbits/s} = 0,3264375 \text{ MBytes/s}$
- $60\% = 8,705 \times 0,6 = 5,223 \text{ Mbits/s} = 0,652875 \text{ MBytes/s}$

- $90\% = 8,705 \times 0,9 = 7,8345 \text{ Mbits/s} = 0,9793125 \text{ MBytes/s}$

4.15 Diagrama de Extração da Porcentagem de Ocupação

Para ilustrar a extração da porcentagem de ocupação, bem como apresentar a participação de cada sistema exposto no Capítulo 3, é apresentada a Figura 4.9.

Figura 4.9: Diagrama de Extração da Porcentagem de Ocupação.



Como é possível observar na Figura 4.9, os pacotes são gerados no Computador 1 pelo Iperf, são enviados para a Soekris que deve entregar ao destinatário, neste caso, Computador 2. Após a entrega, o Iperf gera um *log* de informações com quantidade de pacotes perdidos e enviados, e a partir disso, podemos calcular a capacidade da rede em estudo.

Capítulo 5

Análise dos Dados Obtidos

5.1 Metodologia de Análise de Resultados

A metodologia de análise de resultados envolve a execução de 2 ou 3 passos, dependendo da necessidade. São eles:

- Análise de normalidade;
- Teste de Igualdade;
- Teste de Alternativa Maior;

Os passos estão apresentados e detalhados nas seções subsequentes.

5.1.1 Análise de Normalidade

Para analisar os dados utilizando os métodos estatísticos da forma correta, precisamos saber que distribuição de probabilidades os dados seguem. Assim, poderemos escolher se utilizaremos os testes paramétricos ou os não-paramétricos. Desta forma, a primeira seção deste capítulo trata da análise de normalidade dos dados, enquanto que nas seguintes, as análises completas separadas por variável dependente escolhida.

Para analisar a normalidade dos dados, pode-se utilizar dois testes clássicos: *Anderson-Darling* e/ou o *Shapiro-Wilk*. Foi utilizado o teste de *Shapiro-Wilk*.

Para realizar os testes escolhidos, são necessárias duas hipóteses:

Hipótese 7 *Os dados seguem uma distribuição normal;*

Hipótese 8 *Os dados não seguem uma distribuição normal;*

A hipótese 7 afirma que os dados obtidos seguem a distribuição normal (pode-se utilizar os testes paramétricos) e deve ser aceita se o p-valor do teste for maior que o valor do α escolhido ($\alpha = 0,05$). A hipótese 8, afirma que os dados não seguem uma distribuição normal (deve-se utilizar os testes não-paramétricos) e deve ser aceita se o p-valor dos testes for menor que o valor do α .

As Hipóteses 7 e 8 serão consideradas antes de cada teste para decisão sobre o tipo de teste utilizado nas seções subsequentes. Os resultados dos testes de normalidade se encontram na seção B.

5.1.2 Teste de Igualdade

A partir da definição da normalidade dos dados, deve-se escolher um teste paramétrico ou não-paramétrico. Todos os dados desse trabalho foram considerados não normais e portanto deve-se utilizar testes não-paramétricos para avaliar as hipóteses.

O teste de igualdade escolhido foi o *Pairwise-Wilcoxon*, encontrado no R. Esse teste é baseado em duas hipóteses que verificam se os dados vem de uma população com mesma média. As hipóteses são as que seguem:

Hipótese 9 *O nível x da variável y tem um valor da métrica w médio igual ao nível z da variável y*

Hipótese 10 *O nível x da variável y tem um valor da métrica w médio diferente ao nível z da variável y*

Em que, x é um nível da variável independente y , z é outro nível da variável dependente y e w é uma variável dependente. A hipótese 9 afirma que os dados comparados possuem a mesma média, estatisticamente, e deve ser aceita se o p-valor do teste for maior que o valor do α escolhido ($\alpha = 0,05$), enquanto que a hipótese 10 contraria esta afirmação e deve ser aceita se o p-valor do teste for menor que o valor do α .

Os resultados dos testes de normalidade se encontram na seção C.

5.1.3 Teste de Alternativa Maior

O teste de alternativa maior só será necessário quando o teste de igualdade apontar diferenças entre os dados, ou seja, quando aceitarmos a hipótese alternativa ao executar o mesmo para que seja possível verificar qual nível obteve o melhor resultado.

O teste de alternativa maior escolhido foi o *Pairwise-Wilcoxon*, encontrado no R. Esse teste é baseado em duas hipóteses que verificam se os dados vem de uma população com média maior. As hipóteses são as que seguem:

Hipótese 11 *O nível x da variável y tem um valor da métrica w médio menor que o nível z da variável y*

Hipótese 12 *O nível x da variável y tem um valor da métrica w médio maior que o nível z da variável y*

Em que, x é um nível da variável independente y , z é outro nível da variável dependente y e w é uma variável dependente. A hipótese 11 afirma que o nível x tem uma média menor que o nível z , estatisticamente, e deve ser aceita se o p-valor do teste for maior que o valor do α escolhido ($\alpha = 0,05$), enquanto que a hipótese 12 contraria esta afirmação e deve ser aceita se o p-valor do teste for menor que o valor do α .

Os resultados dos testes de normalidade se encontram na seção D.

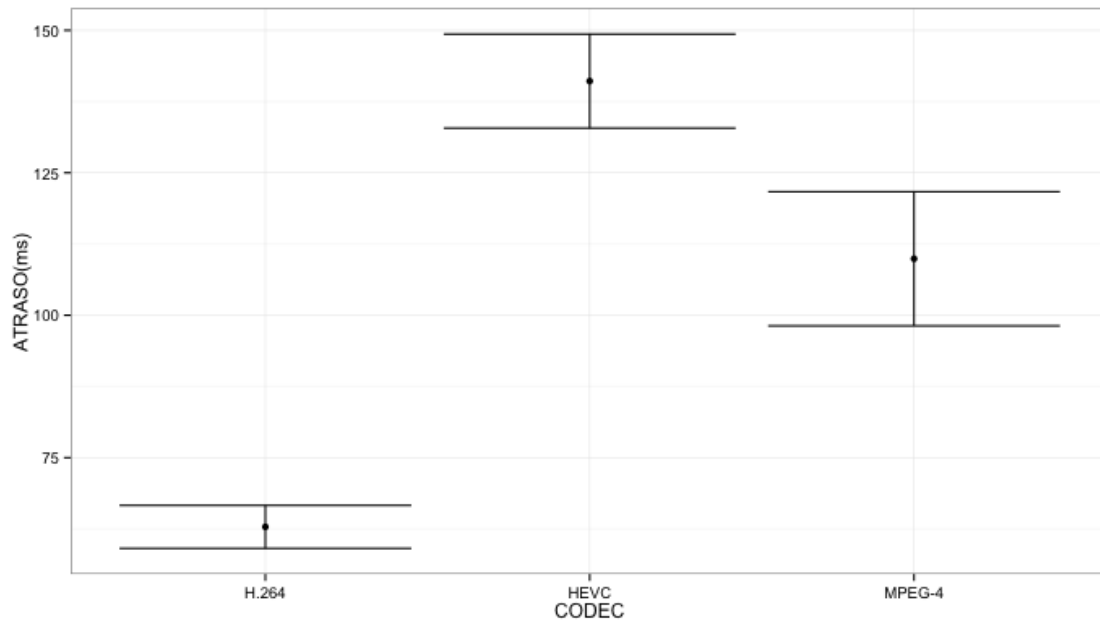
5.2 Codec

5.2.1 Codecs x Atraso

Analisar os dados de atraso relativos aos formatos de compressão utilizados se torna útil para observar se algum formato de compressão influencia no atraso da entrega dos pacotes. Para tanto, foram gerados os intervalos de confiança do atraso separados por formato de compressão.

No gráfico da Figura 5.1, pode-se observar que os intervalos para os três *codecs* estão bem separados, ou seja, é possível concluir que o HEVC apresenta o maior atraso, com o MPEG-4 logo após e o H.264 com o melhor resultado (menor atraso) dentre os três formatos de compressão.

Figura 5.1: Gráfico de intervalos de confiança Atraso x Codec.



Os resultados dos testes de normalidade são mostrados na Tabela B.1, a partir dos quais é possível dizer que os dados não seguem uma distribuição normal.

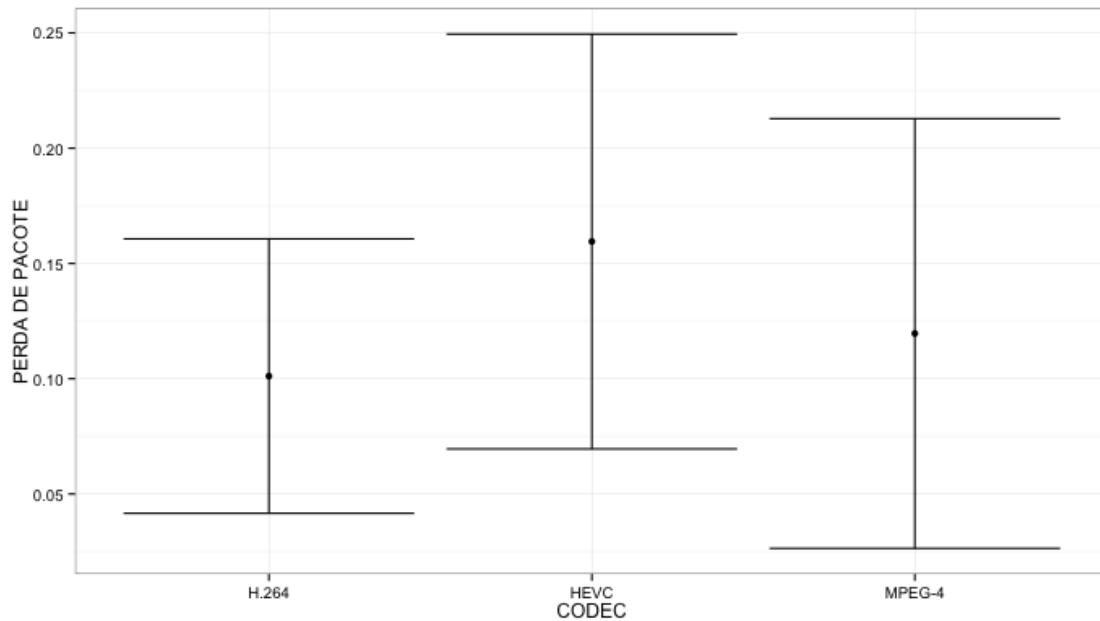
Na Tabela D.1, é mostrado o resultado da execução dos testes no R. É possível observar que o formato de compressão que obteve o maior atraso foi o HEVC, com p-valor $< 2^{-16}$ quando comparado ao H.264 e com p-valor 1,00 quando comparado ao MPEG-4, confirmando os resultados obtidos através dos intervalos de confiança.

5.2.2 Codecs x Perda de Pacote

Analisar os dados de atraso relativos aos formatos de compressão utilizados se torna útil para observar se algum formato de compressão causa uma maior perda de pacotes ou não. Para tanto, foram gerados os intervalos de confiança da perda separados por formato de compressão.

No gráfico da Figura 5.2, vemos que os intervalos dos 3 formatos de compressão estão praticamente sobrepostos o que pode ser um indício de igualdade na perda de pacotes para os três formatos. Para uma análise mais profunda, vamos partir para outros testes.

Os resultados dos testes de normalidade são mostrados na Tabela B.4, a partir dos quais é possível dizer que os dados não seguem uma distribuição normal.

Figura 5.2: Gráfico de intervalos de confiança Perda de Pacote x *Codec*.

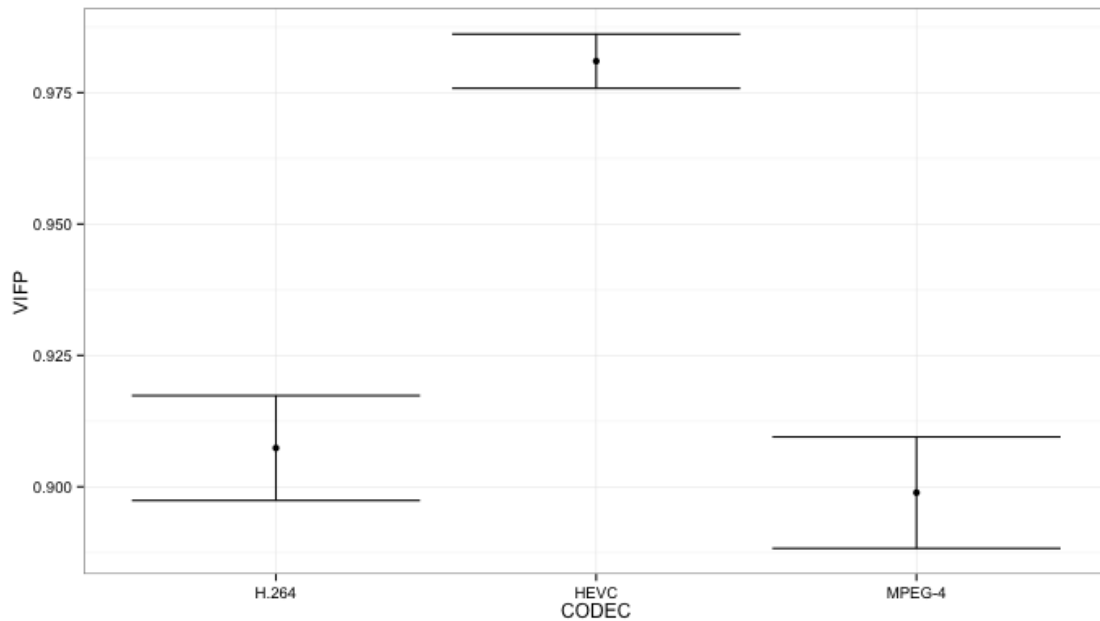
Na Tabela C.4, é mostrado o resultado da execução dos testes no R. Considerando as hipóteses acima e os resultados da tabela em questão, vemos que nenhum valor está abaixo do $\alpha = 0,05$, ou seja, devemos rejeitar a Hipótese alternativa 10 e aceitar a Hipótese nula 9, assim todos os formatos de compressão devem ser considerados com perdas iguais, estatisticamente. Por meio deste resultado, é possível entender que os *codecs* não influenciam na perda de pacotes. Isso ocorre, pelo fato de nenhum *codec* ter o pacote com tratamento ou formato diferente do outro.

5.2.3 *Codecs* x VIFP

Continuando com nosso objetivos, devemos avaliar a variável dependente VIFP agrupada pelos *codecs* utilizados para comprimir os vídeos e assim descobrir se os vídeos podem perder qualidade com a transmissão e qual o melhor formato de compressão para transmissão via *stream*. Com este objetivo, foram gerados os intervalos de confiança.

No gráfico da Figura 5.3, podemos observar que o HEVC se destaca um pouco acima dos outros dois formatos. Com relação à comparação entre o H.264 e o MPEG-4, não é possível concluir nada, visto que os intervalos englobam os pontos centrais. Assim, devemos fazer uma análise mais profunda através de outros testes.

Os resultados dos testes de normalidade são mostrados na Tabela B.8, a partir dos

Figura 5.3: Gráfico de intervalos de confiança VIFP x *Codecs*.

quais é possível dizer que os dados não seguem uma distribuição normal.

Na Tabela C.8, é mostrado o resultado da execução dos testes no R, que indicam que os três *codecs* apresentam resultados estatisticamente diferentes. Os resultados mostram que: com um p-valor de 2^{-16} quando comparado ao H.264 e 2^{-16} quando comparado ao MPEG-4 que existem diferenças entre o HEVC e esses dois formatos. Já com relação ao H.264 e MPEG-4, o teste mostra que com um p-valor de 0,03 ($< \alpha = 0,05$), as médias das duas amostras também não são iguais, estatisticamente, ou seja, existe uma diferença significativa entre a perda de qualidade que o H.264 sofre e que o MPEG-4 sofre.

Para saber qual o melhor formato de compressão, precisamos realizar mais um teste que considera as seguintes Hipóteses:

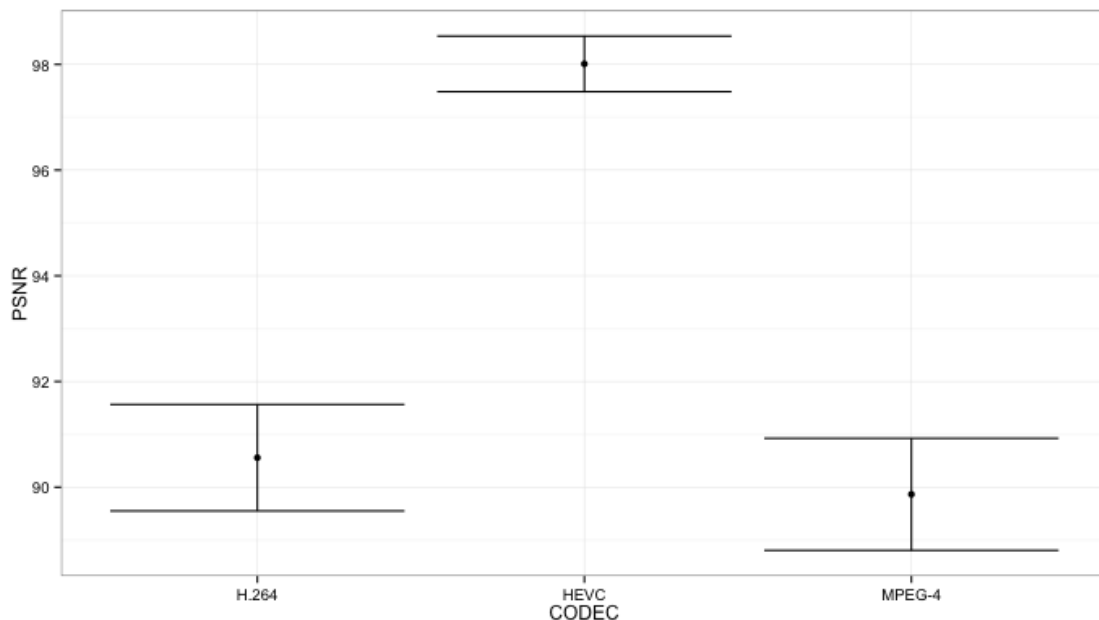
Na Tabela D.4, é mostrado o resultado da execução dos testes no R, que indicam que o HEVC obteve a maior média entre os *codecs*, seguido pelo H.264 e tendo o MPEG-4 com a menor média entre os *codecs* avaliados. Pode-se notar que ao compararmos o HEVC com o H.264 obtém-se um p-valor de 2^{-16} que é inferior ao $\alpha = 0,05$ e assim devemos rejeitar a Hipótese nula 11 e aceitar a Hipótese alternativa 12, ou seja o HEVC obteve uma média mais alta na métrica VIFP com relação ao H.264. Quando comparamos o HEVC com o MPEG-4, tem-se um p-valor de 1,000 que diz que ele obteve

uma média maior que o MPEG-4 na variável dependente VIFP. O mesmo acontece na comparação do MPEG-4 com o H.264, com o H.264 obtendo uma média mais alta na variável VIFP.

5.2.4 Codecs x PSNR

Continuando com nossos objetivos, devemos avaliar a variável dependente PSNR agrupada pelos *codecs* utilizados para comprimir os vídeos e assim descobrir se os vídeos podem perder qualidade com a transmissão e qual o melhor formato de compressão para transmissão via *stream*. Com este objetivo, foram gerados os intervalos de confiança.

Figura 5.4: Gráfico de intervalos de confiança PSNR x *Codecs*.



No gráfico da Figura 5.4, podemos observar que o HEVC se destaca um pouco acima dos outros dois formatos. Com relação à comparação entre o H.264 e o MPEG-4, não é possível concluir nada com certeza, visto que os intervalos englobam os pontos centrais. Assim, devemos fazer uma análise mais profunda através de outros testes.

Os resultados dos testes de normalidade são mostrados na Tabela B.10, a partir dos quais é possível dizer que os dados não seguem uma distribuição normal.

Na Tabela C.10, é mostrado o resultado da execução dos testes no R. Os resultados mostram que: com um p-valor de 2^{-16} quando comparado ao H.264 e 2^{-16} quando

comparado ao MPEG-4 que existem diferenças entre o HEVC e esses dois formatos. Já com relação ao H.264 e MPEG-4, o teste mostra que com um p-valor de 0,061 ($> \alpha = 0,05$), as médias das duas amostras são iguais, estatisticamente, ou seja, não existe uma diferença significativa entre a perda de qualidade que o H.264 sofre e que o MPEG-4 sofre.

Para saber qual o melhor formato de compressão, precisamos realizar mais um teste que considera as seguintes Hipóteses:

Na Tabela D.6, é mostrado o resultado da execução dos testes no R. Pode-se notar que ao compararmos o HEVC com o H.264 obtém-se um p-valor de 2^{-16} que é inferior ao $\alpha = 0,05$ e assim devemos rejeitar a Hipótese alternativa 12 e aceitar a Hipótese nula 11, ou seja o HEVC obteve uma média mais alta na métrica PSNR com relação ao H.264. Quando comparamos o HEVC com o MPEG-4, tem-se um p-valor de 1,000 que diz que ele obteve uma média maior que o MPEG-4 na variável dependente PSNR.

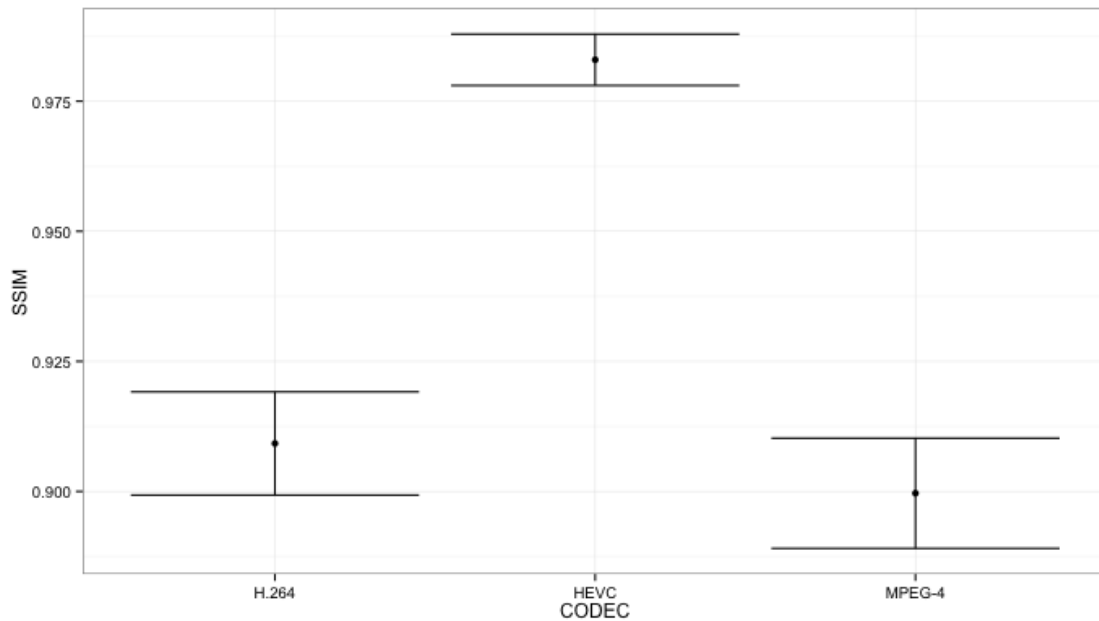
5.2.5 Codecs x SSIM

Continuando com nossos objetivos, devemos avaliar a variável dependente SSIM agrupada pelos *codecs* utilizados para comprimir os vídeos e assim descobrir se os vídeos podem perder qualidade com a transmissão e qual o melhor formato de compressão para transmissão via *stream*. Com este objetivo, foram gerados os intervalos de confiança.

No gráfico da Figura 5.5, podemos observar que o HEVC se destaca um pouco acima dos outros dois formatos, mas ainda não podemos chegar a nenhuma conclusão, visto que os intervalos se sobrepõem, mesmo que por uma fração pequena. Com relação à comparação entre o H.264 e o MPEG-4, não é possível concluir nada com certeza, visto que os intervalos englobam os pontos centrais. Assim, devemos fazer uma análise mais profunda através de outros testes.

Os resultados dos testes de normalidade são mostrados na Tabela B.12, a partir dos quais é possível dizer que os dados não seguem uma distribuição normal.

Na Tabela C.1, é mostrado o resultado da execução dos testes no R. Os resultados mostram que: com um p-valor de 2^{-16} quando comparado ao H.264 e 2^{-16} quando comparado ao MPEG-4 que existem diferenças entre o HEVC e esses dois formatos. Já com relação ao H.264 e MPEG-4, o teste mostra que com um p-valor de 0,016

Figura 5.5: Gráfico de intervalos de confiança SSIM x *Codecs*.

($> \alpha = 0,05$), as médias das duas amostras não são iguais, estatisticamente, ou seja, existe uma diferença significativa entre a perda de qualidade que o H.264 sofre e que o MPEG-4 sofre.

Para saber qual o melhor formato de compressão, precisamos realizar mais um teste que considera as seguintes Hipóteses:

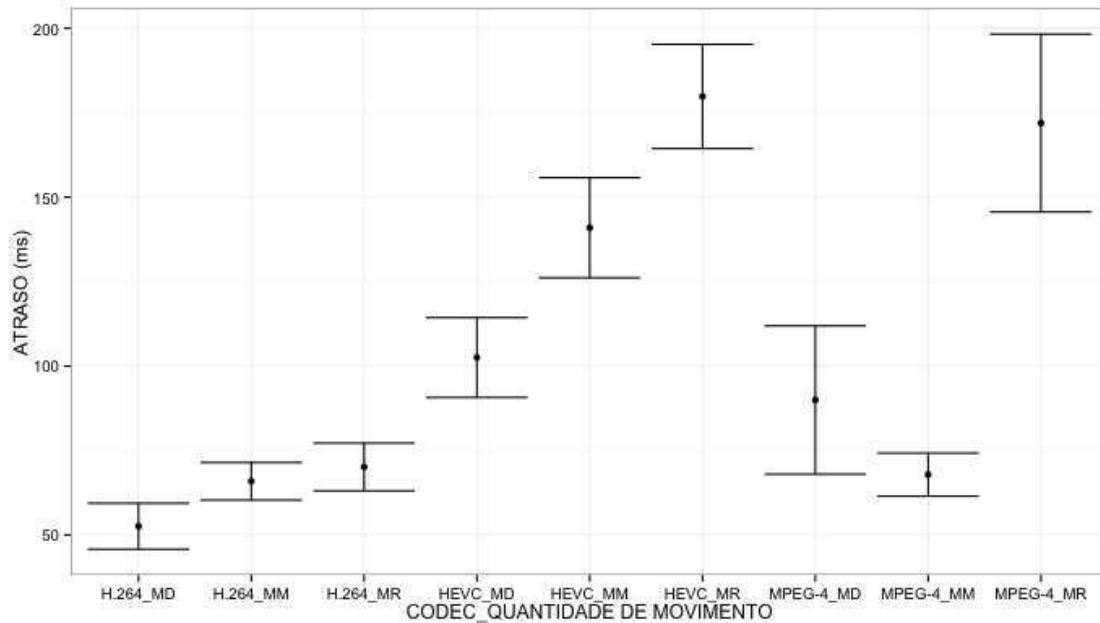
Na Tabela D.8, é mostrado o resultado da execução dos testes no R. Pode-se notar que ao compararmos o HEVC com o H.264 obtém-se um p-valor de 2^{-16} que é inferior ao $\alpha = 0,05$ e assim devemos rejeitar a Hipótese alternativa 12 e aceitar a Hipótese nula 11, ou seja o HEVC obteve uma média mais alta na métrica SSIM com relação ao H.264. Quando comparamos o HEVC com o MPEG-4, tem-se um p-valor de 1,000 que diz que ele obteve uma média maior que o MPEG-4 na variável dependente SSIM. Na comparação entre MPEG-4 e H.264, obtivemos um p-valor de 1,000 ($> \alpha = 0,05$), ou seja, devemos rejeitar a Hipótese alternativa 12 e aceitar a Hipótese nula 11, com o H.264 obtendo uma média mais alta na métrica SSIM.

5.2.6 *Codecs* x Quantidade de Movimento x Atraso

Para que possamos aprofundar ainda mais os resultados obtidos, faz-se necessária a avaliação do atraso agrupado pela quantidade de movimento em cada *codec*. Assim,

nessa seção estes resultados serão discutidos.

Figura 5.6: Gráfico de intervalos de confiança *Codecs* x Quantidade de Movimento x Atraso.



No gráfico da Figura 5.6, são mostrados os resultados obtidos do atraso agrupados pela quantidade de movimento em cada *codec*. Através desse gráfico é possível observar que, aparentemente, existe um atraso crescente começando do H.264 MD até o HEVC MR, ou seja, os pacotes estão demorando cada vez mais tempo para chegar. Quando observamos os atrasos relacionados ao MPEG-4, notamos uma sobreposição entre os intervalos do MPEG-4 MD e MPEG-4 MM, com o atraso do MPEG-4 MR mais alto e, aparentemente, igual ao do HEVC MR. Para chegar a conclusões mais precisas, vamos realizar os testes estatísticos necessários.

Os resultados dos testes de normalidade são mostrados na Tabela B.13, a partir dos quais é possível dizer que os dados não seguem uma distribuição normal.

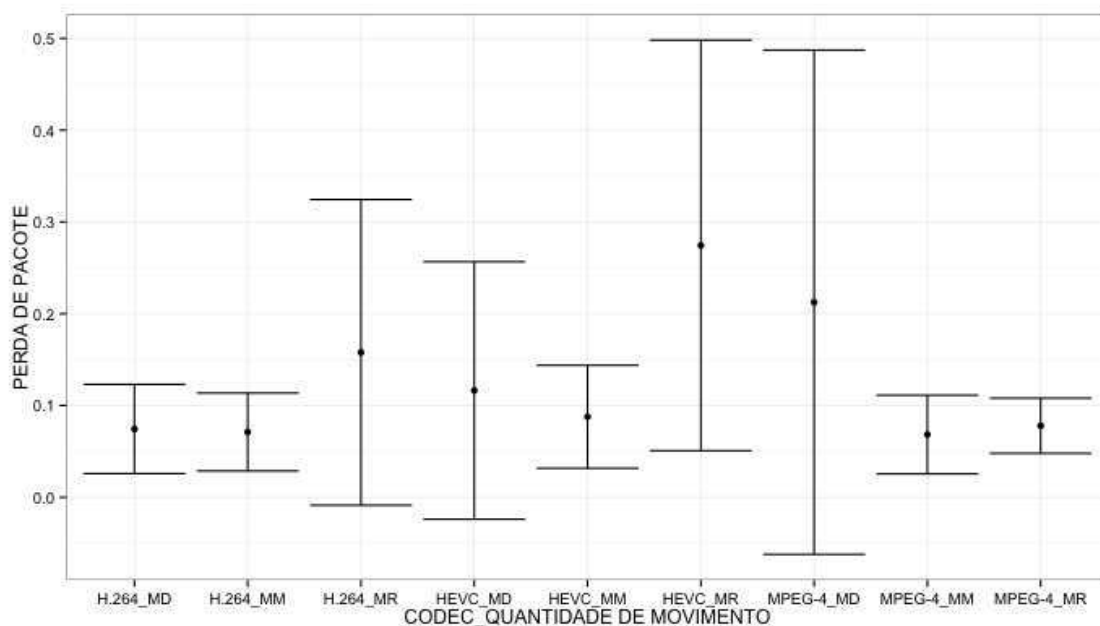
O resultado do teste de igualdade para a perda de pacotes agrupada pela quantidade de movimento e pelos *codecs* se encontra na Tabela C.12. Nessa Tabela, pode-se observar que os únicos valores acima do $\alpha = 0,05$ resultam da comparação entre H.264 MR e H.264 MM, MPEG-4 MM e H.264 MM, MPEG-4 MM e H.264 MR, HEVC MR e MPEG-4 MR, ou seja, estas são as comparações que produziram resultados iguais, estatisticamente. As outras comparações são consideradas diferentes e podem ser in-

duzidas a partir do gráfico da Figura 5.6.

5.2.7 Codecs x Quantidade de Movimento x Perda de Pacote

Para que possamos aprofundar ainda mais os resultados obtidos, faz-se necessária a avaliação da perda de pacote agrupada pela quantidade de movimento em cada *codec*. Assim, nessa seção estes resultados serão discutidos.

Figura 5.7: Gráfico de intervalos de confiança *Codecs* x Quantidade de Movimento x Perda de Pacote.



No gráfico da Figura 5.7, são mostrados os resultados obtidos da perda de pacote agrupados pela quantidade de movimento em cada *codec*. A partir desse gráfico é possível observar que todos os intervalos estão sobrepostos e portanto, não é possível tirar conclusões apenas com os intervalos de confiança. Faz-se necessária a execução de testes estatísticos.

Os resultados dos testes de normalidade são mostrados na Tabela B.14, a partir dos quais é possível dizer que os dados não seguem uma distribuição normal.

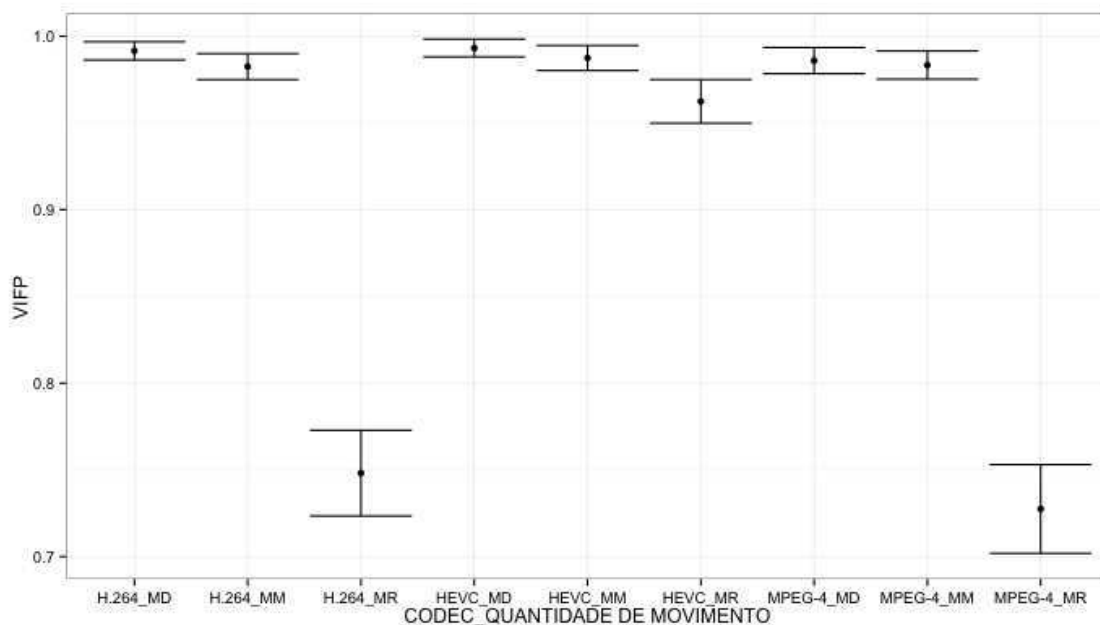
O resultado do teste de igualdade para a perda de pacotes agrupada pela quantidade de movimento e pelos *codecs* se encontra na Tabela C.13. Nessa Tabela, pode-se observar que não existe nenhum valor abaixo do $\alpha = 0,05$, e portanto deve-se rejeitar

a Hipótese 10 e aceitar a Hipótese 9, ou seja, todas as perdas de pacotes apresentaram a mesma média, estatisticamente. Este resultado pode ser explicado pelo fato de nenhum *codec* ou quantidade de movimento receber nenhum tratamento especial no encaminhamento de pacotes e assim o descarte dos pacotes acontece por congestionamento da rede.

5.2.8 *Codecs* x Quantidade de Movimento x VIFP

Para que possamos aprofundar ainda mais os resultados obtidos, faz-se necessária a avaliação do VIFP agrupado pela quantidade de movimento em cada *codec*. Assim, nessa seção estes resultados serão discutidos.

Figura 5.8: Gráfico de intervalos de confiança *Codecs* x Quantidade de Movimento x VIFP.



No gráfico da Figura 5.8, são mostrados os resultados obtidos da perda de pacote agrupados pela quantidade de movimento em cada *codec*. Nesse desse gráfico é possível observar que, para todos os *codecs* os intervalos correspondentes às quantidades de movimento MD e MM estão sobrepostos, o que impossibilita qualquer conclusão. Observando os intervalos correspondentes à quantidade de movimento MR, é possível ver que o HEVC MR obteve um resultado melhor (com uma média mais alta) e que os

intervalos do H.264 MR e MPEG-4 MR estão sobrepostos. Assim, faz-se necessária a execução de testes estatísticos para resultados mais precisos.

Os resultados dos testes de normalidade são mostrados na Tabela B.15, a partir dos quais é possível dizer que os dados não seguem uma distribuição normal.

O resultado do teste de igualdade para a perda de pacotes agrupada pela quantidade de movimento e pelos *codecs* se encontra na Tabela C.15. Nessa Tabela, pode-se observar que as quantidades de movimento MM e MD são estatisticamente iguais independente do *codec* escolhido, ou seja, o p-valor das comparações ficou acima do valor do $\alpha = 0,05$. Quando as quantidades de movimento MR são comparadas, nota-se que os p-valores para o MPEG-4 e H.264 também está acima do valor do α , e portanto são estatisticamente iguais. A diferença é apresentada quando o MPEG-4 e o H.264 são comparados ao HEVC. Ambos os p-valores estão abaixo do α e com a ajuda do gráfico da Figura 5.8, chega-se a conclusão que o HEVC apresenta melhores resultados para vídeos com mais movimento.

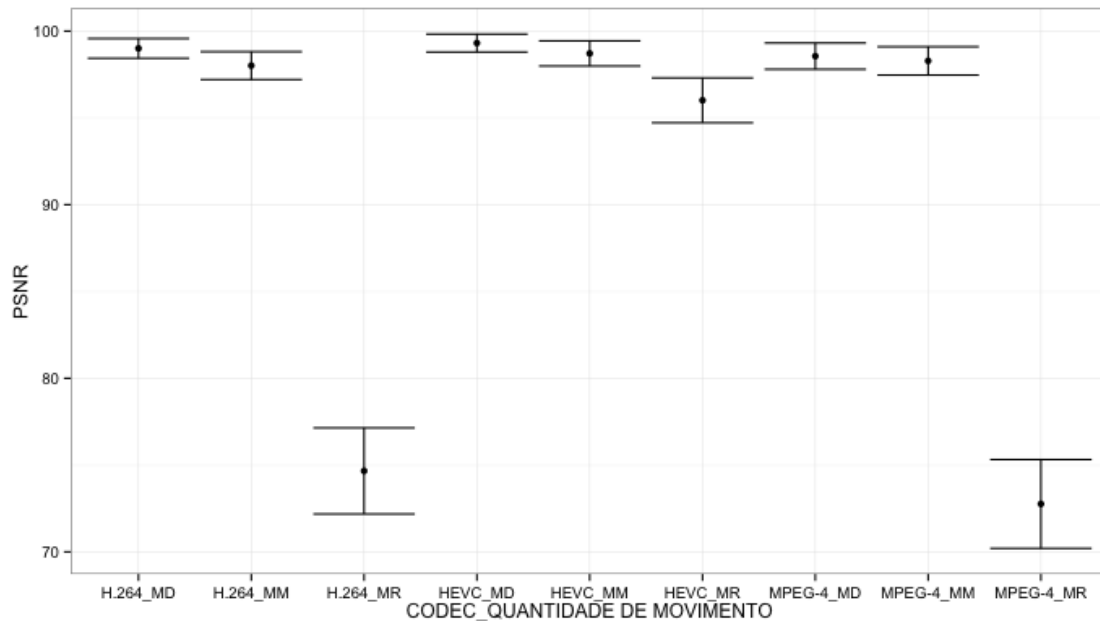
Comparando este resultado com o obtido na subseção 5.2.3, em que o HEVC apresenta melhores resultados na comparação geral de *codecs*, percebe-se que estes melhores resultados do HEVC se dão justamente pela sua capacidade de lidar melhor com vídeos com mais movimento.

5.2.9 *Codecs* x Quantidade de Movimento x PSNR

Para que possamos aprofundar ainda mais os resultados obtidos, faz-se necessária a avaliação do PSNR agrupado pela quantidade de movimento em cada *codec*. Assim, nessa seção estes resultados serão discutidos.

No gráfico da Figura 5.9, são mostrados os resultados obtidos da perda de pacote agrupados pela quantidade de movimento em cada *codec*. Nesse desse gráfico é possível observar que, para todos os *codecs* os intervalos correspondentes às quantidades de movimento MD e MM estão sobrepostos, o que impossibilita qualquer conclusão. Observando os intervalos correspondentes à quantidade de movimento MR, é possível ver que o HEVC MR obteve um resultado melhor (com uma média mais alta) e que os intervalos do H.264 MR e MPEG-4 MR estão sobrepostos. Assim, faz-se necessária a execução de testes estatísticos para resultados mais precisos.

Figura 5.9: Gráfico de intervalos de confiança *Codecs* x Quantidade de Movimento x PSNR.



Os resultados dos testes de normalidade são mostrados na Tabela B.16, a partir dos quais é possível dizer que os dados não seguem uma distribuição normal.

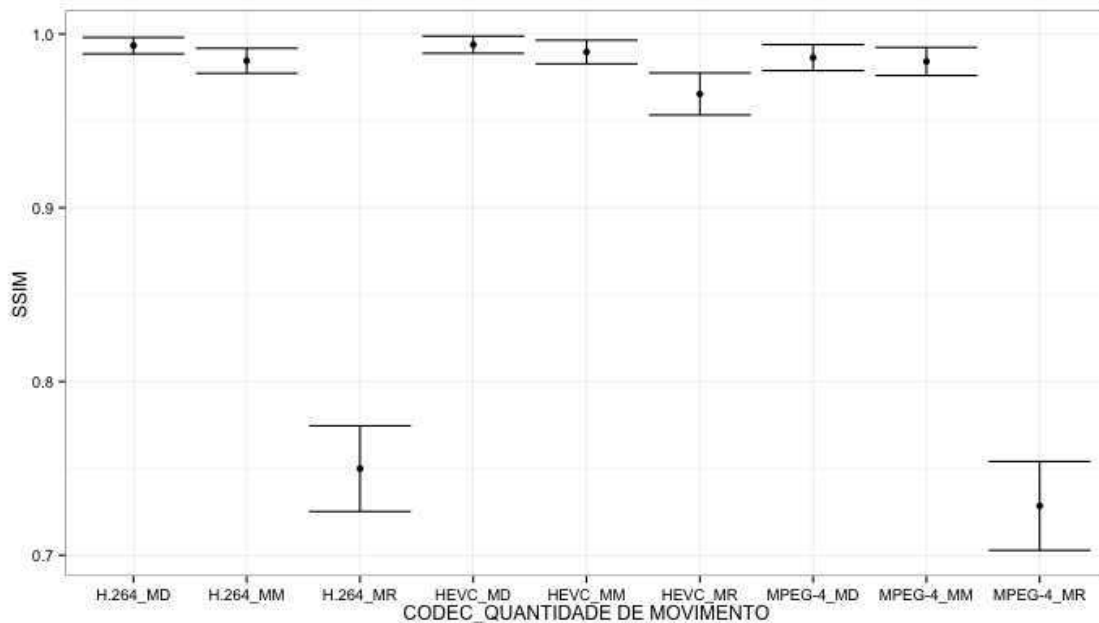
O resultado do teste de igualdade para a perda de pacotes agrupada pela quantidade de movimento e pelos *codecs* se encontra na Tabela C.16. Nessa Tabela, pode-se observar que as quantidades de movimento MM e MD são estatisticamente iguais independente do *codec* escolhido, ou seja, o p-valor das comparações ficou acima do valor do $\alpha = 0,05$. Quando as quantidades de movimento MR são comparadas, nota-se que os p-valores para o MPEG-4 e H.264 também está acima do valor do α , e portanto são estatisticamente iguais. A diferença é apresentada quando o MPEG-4 e o H.264 são comparados ao HEVC. Ambos os p-valores estão abaixo do α e com a ajuda do gráfico da Figura 5.9, chega-se a conclusão que o HEVC apresenta melhores resultados para vídeos com mais movimento.

Comparando este resultado com o obtido na subseção 5.2.4, em que o HEVC apresenta melhores resultados na comparação geral de *codecs*, percebe-se que estes melhores resultados do HEVC se dão justamente pela sua capacidade de lidar melhor com vídeos com mais movimento.

5.2.10 Codecs x Quantidade de Movimento x SSIM

Para que possamos aprofundar ainda mais os resultados obtidos, faz-se necessária a avaliação do SSIM agrupado pela quantidade de movimento em cada *codec*. Assim, nessa seção estes resultados serão discutidos.

Figura 5.10: Gráfico de intervalos de confiança *Codecs* x Quantidade de Movimento x SSIM.



No gráfico da Figura 5.10, são mostrados os resultados obtidos da perda de pacote agrupados pela quantidade de movimento em cada *codec*. Nesse desse gráfico é possível observar que, para todos os *codecs* os intervalos correspondentes às quantidades de movimento MD e MM estão sobrepostos, o que impossibilita qualquer conclusão. Observando os intervalos correspondentes à quantidade de movimento MR, é possível ver que o HEVC MR obteve um resultado melhor (com uma média mais alta) e que os intervalos do H.264 MR e MPEG-4 MR estão sobrepostos. Assim, faz-se necessária a execução de testes estatísticos para resultados mais precisos.

Os resultados dos testes de normalidade são mostrados na Tabela B.17, a partir dos quais é possível dizer que os dados não seguem uma distribuição normal.

O resultado do teste de igualdade para a perda de pacotes agrupada pela quantidade de movimento e pelos *codecs* se encontra na Tabela C.14. Nessa Tabela, pode-se observar que as quantidades de movimento MM e MD são estatisticamente iguais in-

dependente do *codec* escolhido, ou seja, o p-valor das comparações ficou acima do valor do $\alpha = 0,05$. Quando as quantidades de movimento MR são comparadas, nota-se que os p-valores para o MPEG-4 e H.264 também está acima do valor do α , e portanto são estatisticamente iguais. A diferença é apresentada quando o MPEG-4 e o H.264 são comparados ao HEVC. Ambos os p-valores estão abaixo do α e com a ajuda do gráfico da Figura 5.10, chega-se a conclusão que o HEVC apresenta melhores resultados para vídeos com mais movimento.

Comparando este resultado com o obtido na subseção 5.2.5, em que o HEVC apresenta melhores resultados na comparação geral de *codecs*, percebe-se que estes melhores resultados do HEVC se dão justamente pela sua capacidade de lidar melhor com vídeos com mais movimento.

5.3 Potência do Sinal

5.3.1 Potência do sinal x Atraso

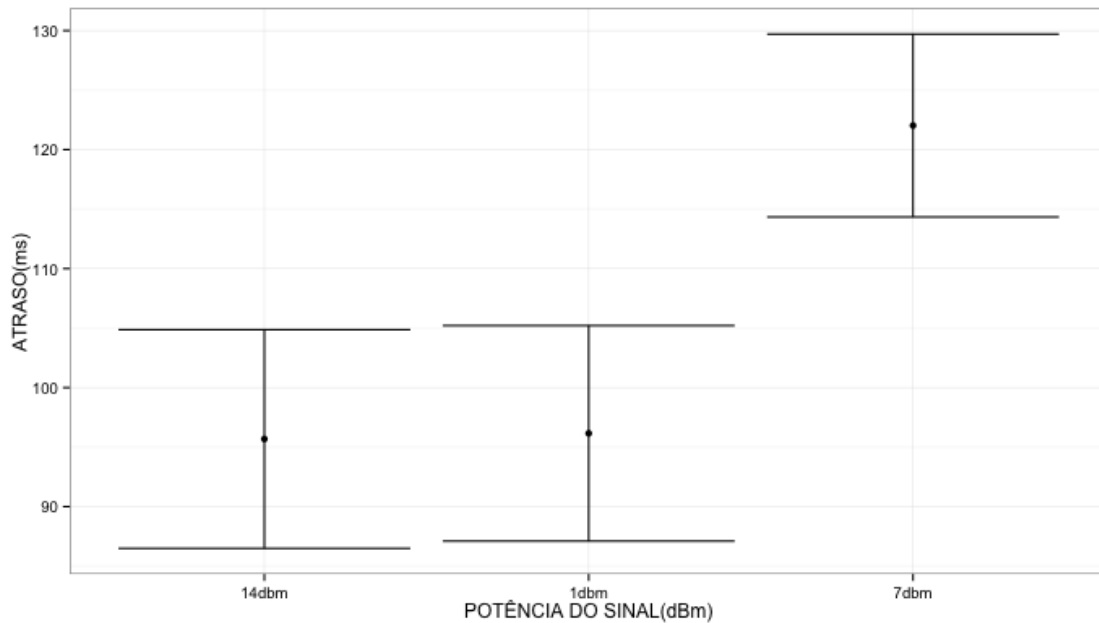
Analisar os dados de atraso relativos à potência do sinal se torna útil para verificar se o sinal influenciou, da forma esperada, o resultado obtido, ou seja, se à medida que a potência é aumentada, o atraso diminui. Com este objetivo, foram gerados os intervalos de confiança do atraso agrupados pela potência de sinal utilizada.

No gráfico da Figura 5.11, podemos observar que o intervalo da potência 14dBm engloba o intervalo da potência 1dBm deixando com indícios de que as duas potências apresentaram o mesmo atraso. Quando comparados ao intervalo da potência 7dBm, observamos que esta apresentou um atraso ligeiramente maior. Para confirmar as observações obtidas, foram realizados mais testes.

Os resultados dos testes de normalidade são mostrados na Tabela B.2, a partir dos quais é possível dizer que os dados não seguem uma distribuição normal.

Na Tabela C.2, é mostrado o resultado da execução dos testes no R. É possível observar que não foi obtido o comportamento esperado, ou seja, à medida que aumentamos a potência o atraso não diminuiu em todos os casos. O atraso obtido foi menor em 14dBm, que foi considerado igual ao atraso em 1dBm, já que o p-valor encontrado foi de 0,57 que é maior que o $\alpha = 0,05$, fazendo com que a Hipótese 10 seja rejeitada

Figura 5.11: Gráfico de intervalos de confiança Atraso x Potência do sinal.



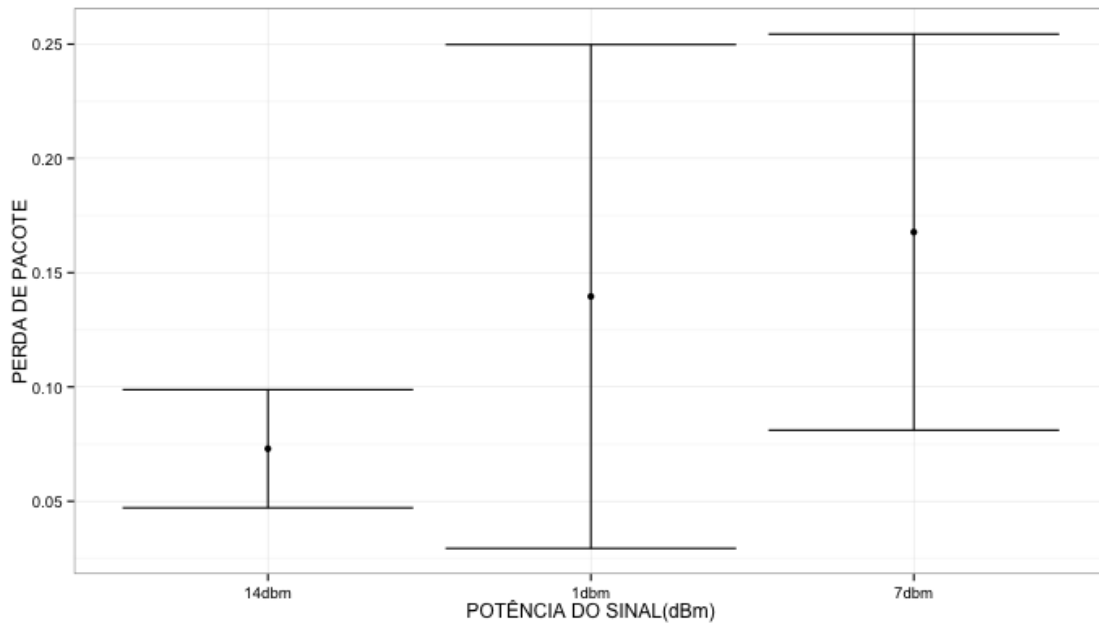
para que a 9 seja aceita. Quando é comparada com a potência de 7dBm, vemos que o p-valor é menor que o $\alpha = 0,05$ e, com isto, a Hipótese 10 é aceita, ou seja, as amostras das potências 14dBm e 1dBm possuem médias diferentes quando comparadas à potência 7dBm. Com isto, tem-se a constatação de que a potência 7dBm apresentou um atraso maior. Isso pode ser justificado, utilizando o fato de termos escolhido um canal no qual já existiam redes no mesmo ambiente, ou seja, durante as transmissões pode ter ocorrido outras transmissões que acabaram impactando no atraso apresentado pelas potências escolhidas. Este fato é aceitável dado que isto ocorre com frequência no tipo de ambiente estudado.

5.3.2 Potência do sinal x Perda de Pacotes

Analisar os dados de perda de pacote relativos à potência do sinal se torna útil para verificar se o sinal influenciou, da forma esperada, o resultado obtido, ou seja, se à medida que aumentamos a potência, a perda de pacote diminui. Com este objetivo, foram gerados os intervalos de confiança do atraso agrupados pela potência de sinal utilizada.

No gráfico da Figura 5.12, é possível ver que o intervalo de 1dBm engloba o intervalo de 7dBm, com o de 14 dBm dando a impressão de ser menor e, assim, confirmando o

Figura 5.12: Gráfico de intervalos de confiança Perda de Pacote x Potência do sinal.



que já era esperado. Entretanto, não podemos chegar a conclusões com base apenas intervalos visto que os pontos centrais de cada intervalo estão dentro do intervalo das outras potências. É necessário realizar outros testes para uma análise com grau de certeza maior.

Os resultados dos testes de normalidade são mostrados na Tabela B.5, a partir dos quais é possível dizer que os dados não seguem uma distribuição normal.

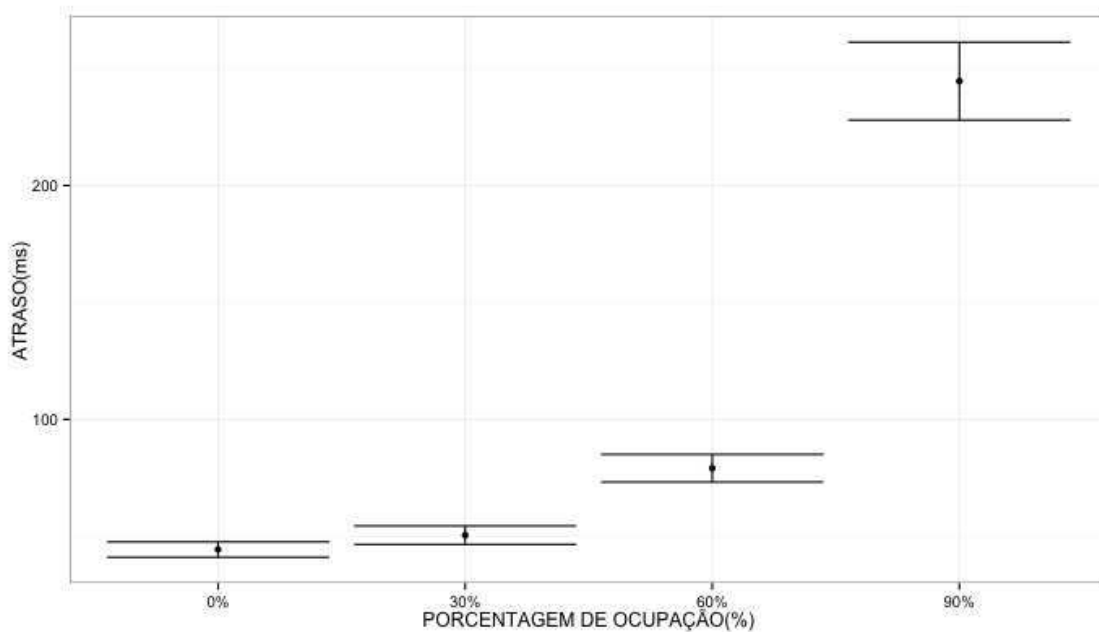
Na Tabela C.5, são mostrados os resultados dos testes realizados no R. Podemos notar que com um p-valor de 0,91 ($> \alpha = 0,05$) a potência 1dBm é considerada igual, estatisticamente, à potência 14dBm, com p-valor 0,42 ($> \alpha = 0,05$) a potência 7dBm é considerada igual, estatisticamente, à potência 14dBm e com p-valor 0,42 ($> \alpha = 0,05$) a potência 7dBm é considerada igual à potência 1dBm. Assim, não se confirma o que era esperado, ou seja, a potência do sinal não influenciou na perda de pacotes testes realizados. Isto se justifica com o fato de que a potência do sinal não estava suficientemente baixa para que houvesse perda de pacote significativamente maior na potência 1dBm do que nas demais potências.

5.4 Porcentagem de Ocupação

5.4.1 Porcentagem de Ocupação x Atraso

Analisar os dados de atraso relativos à porcentagem de ocupação é importante para verificar se a porcentagem influenciou, da forma que pensa-se, o resultado obtido, ou seja, se à medida que a ocupação do canal de transmissão é aumentada, o atraso aumenta. Para verificar esta informação, foram gerados os intervalos de confiança do atraso agrupados pelas porcentagens de ocupação.

Figura 5.13: Gráfico de intervalos de confiança Atraso x Porcentagem de Ocupação.



No gráfico da Figura 5.13, é possível observar que quando aumentamos a porcentagem de ocupação do canal, o atraso aumenta, ou seja, nossa suposição inicial se confirma. A única dúvida que podemos ter é quando comparamos as porcentagens de 0% e 30%, já que existe sobreposição dos intervalos. Para confirmar as observações feitas através dos intervalos de confiança e sanar a dúvida sobre as porcentagens de 0% e 30% faremos mais testes.

Os resultados dos testes de normalidade são mostrados na Tabela B.3, a partir dos quais é possível dizer que os dados não seguem uma distribuição normal.

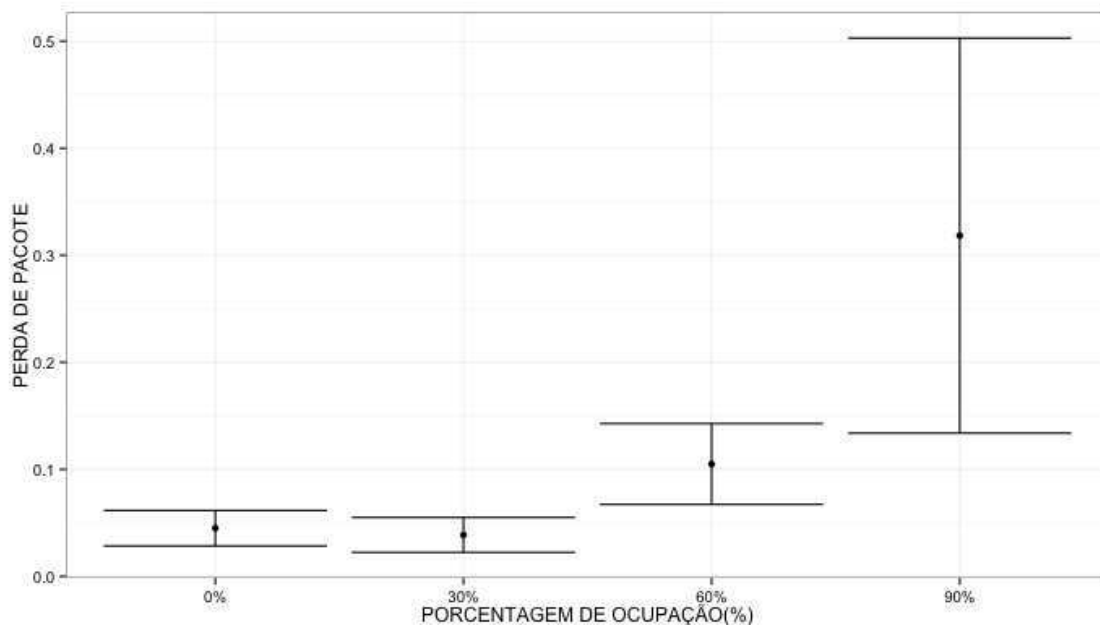
Na Tabela C.3, é mostrado o resultado da execução dos testes no R. Observamos que no caso da porcentagem de ocupação do canal de transmissão, acontece exatamente

o que era esperado, ou seja, à medida que aumentamos a ocupação do canal, o atraso aumenta. Isto é verificado pelos valores sempre abaixo do $\alpha = 0,05$, fazendo com que seja rejeitada a Hipótese 9 e seja aceita a Hipótese 10, ou seja, as porcentagens obtiveram médias diferentes. Unindo esta constatação com a análise dos intervalos de confiança do gráfico da Figura 5.13, chegamos à conclusão que esperávamos.

5.4.2 Porcentagem de Ocupação x Perda de Pacote

Analisar os dados de perda de pacote relativos à porcentagem de ocupação é importante para verificar se a porcentagem influenciou, da forma que pensamos, o resultado obtido. Ou seja, se à medida que aumentamos a ocupação do canal de transmissão, a perda de pacote aumenta. Para verificar esta informação, foram gerados os intervalos de confiança do atraso agrupados pelas porcentagens de ocupação.

Figura 5.14: Gráfico de intervalos de confiança Perda de Pacote x Porcentagem de Ocupação.



No gráfico da Figura 5.14, podemos observar que o intervalo caso base (0%), engloba os pontos centrais do intervalo de 30%, assim não podemos decidir qual dos dois possui a melhor perda de pacotes. Com relação ao caso de 90%, a parte inferior do seu intervalo ainda se encontra entrelaçada com o intervalo de 60% fazendo com que não possamos

tirar conclusões sobre este caso. Assim, precisamos de mais testes para concluir sobre a perda de pacote e a porcentagem de ocupação

Os resultados dos testes de normalidade são mostrados na Tabela B.6, a partir dos quais é possível dizer que os dados não seguem uma distribuição normal.

Na Tabela C.6, é mostrado o resultado da execução dos testes no R. Para aceitar a Hipótese 10 (para que as porcentagens tenham perdas diferentes entre si), é preciso ter um p-valor menor que $\alpha = 0,05$. Assim, pela tabela em questão, as únicas comparações que produzem resultados diferentes são: 90% e 0%, 90% e 30%, 60% e 30%. Precisamos realizar mais um teste para verificar qual porcentagem de ocupação obteve mais e meperda testes.

Realizamos mais um teste utilizando as Hipóteses 11 e 12. A partir deste teste, foram gerados os resultados que podemos observar na Tabela D.2, ou seja, com p-valores inferiores ao $\alpha = 0,05$, rejeitamos a Hipótese nula 11 e aceitamos a alternativa 12, com 90% obtendo mais perda que as demais porcentagens de ocupação, como era esperado. Com relação à comparação entre 60% e 30%, vemos que o p-valor é novamente abaixo do $\alpha = 0,05$, rejeitando a hipótese nula e aceitando a alternativa, ou seja, 60% obteve uma perda maior que 30%.

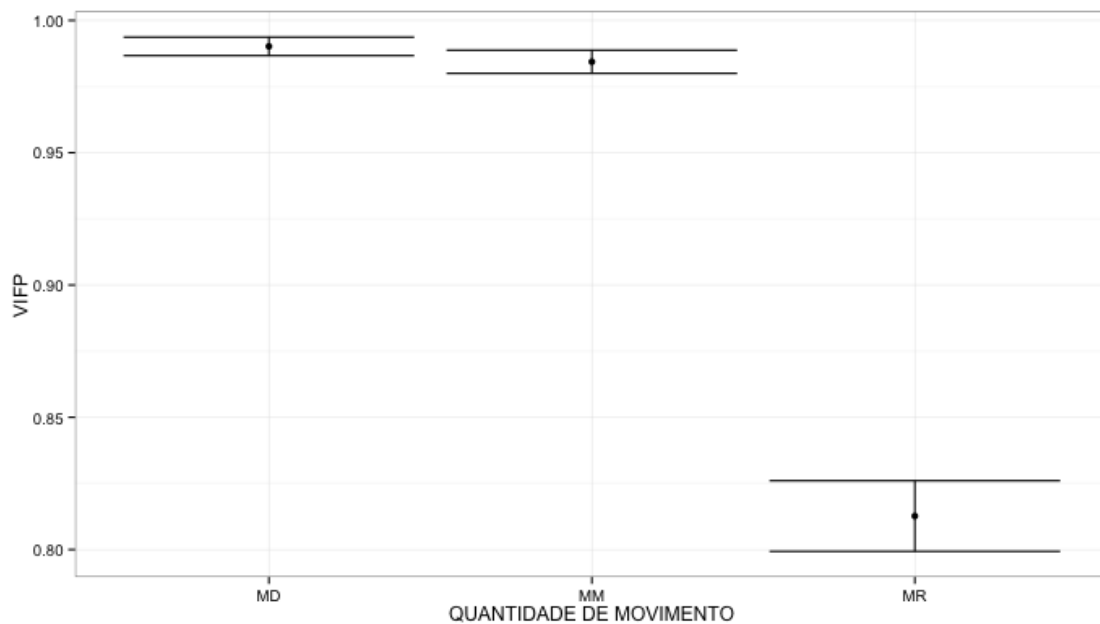
5.5 Quantidade de Movimento

5.5.1 Quantidade de Movimento x VIFP

Para que possamos atingir nossos objetivos devemos, primeiramente, comparar a variável dependente VIFP agrupada pelas categorias de movimento que utilizamos em nossos testes, e assim verificar a influência da quantidade de movimento na qualidade do vídeo recebido de acordo com a variável VIFP. Com este objetivo, foram gerados os intervalos de confiança.

No gráfico da Figura 5.15, é possível notar que quantidade de movimento MR está com o seu intervalo bem abaixo no gráfico, o que diz que ela é a quantidade de movimento que mais foi afetada pela transmissão, perdendo mais qualidade de imagem que as outras. Nada se pode afirmar sobre a relação entre MD e MM, já que os pontos centrais dos intervalos estão englobados pelos dois intervalos. Assim, faz-se necessária

Figura 5.15: Gráfico de intervalos de confiança VIFP x Quantidade de Movimento.



a execução de testes para verificação da influência da quantidade de movimento na qualidade do vídeo recebido.

Os resultados dos testes de normalidade são mostrados na Tabela B.7, a partir dos quais é possível dizer que os dados não seguem uma distribuição normal.

Na Tabela C.7, é mostrado o resultado da execução dos testes no R. É possível observar que a quantidade MR possui p-valor 2^{-16} quando comparado ao MM e 2^{-16} quando comparado ao MD, os dois valores estão bem abaixo do valor $\alpha = 0,05$ e assim devemos rejeitar a Hipótese nula 9 e aceitar a alternativa 10, ou seja, os vídeos recebidos da categoria MR possuem um VIFP menor ao serem transmitidos (de acordo com o intervalo de confiança). Quanto à comparação entre MM e MD, o valor de 0,016 faz com que seja rejeitada a nula e seja aceita a alternativa, ou seja, os vídeos recebidos não possuem a mesma média, estatisticamente.

Para verificar qual dos conteúdos (MD ou MM), apresentou o melhor desempenho, precisamos realizar mais um teste. Este será baseado nas seguintes Hipóteses:

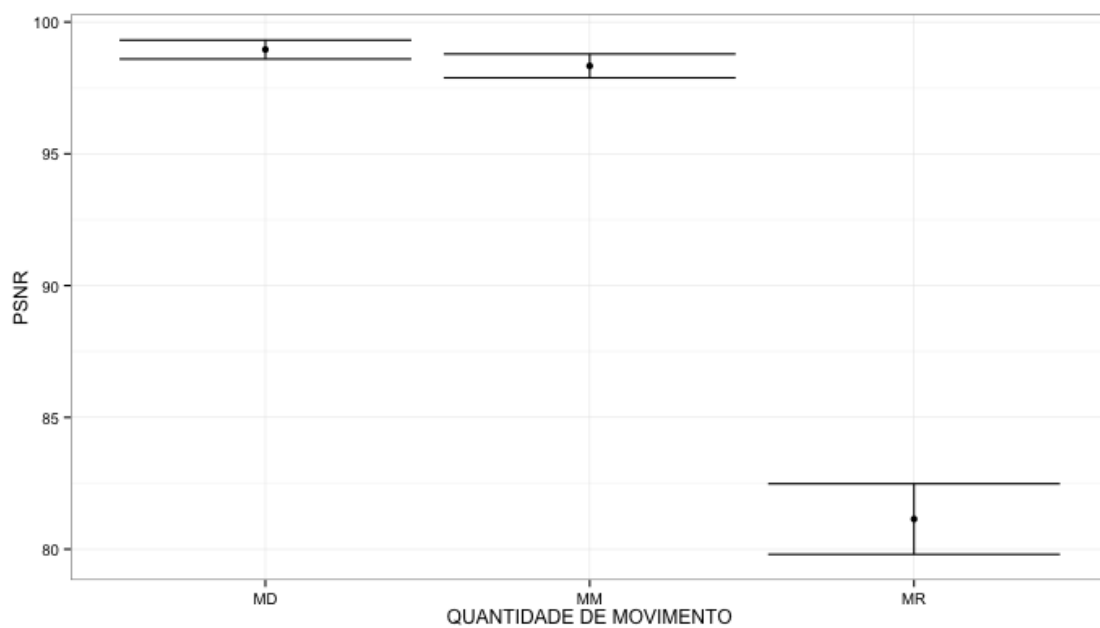
Na Tabela D.3, são mostrados os resultados de comparação do VIFP para as quantidades de movimento MD e MM. Nesta, podemos ver que o p-valor encontrado foi de 0,016 que é menor que o $\alpha = 0,05$, o que faz rejeitar a hipótese nula e aceitar a alternativa, ou seja, os vídeos com quantidade de movimento MD, obtiveram uma menor

perda de qualidade do que os vídeos com quantidade de movimento MM. Este resultado era esperado, visto que os vídeos de movimento MD necessitam de uma menor *bitrate* para que sejam exibidos na qualidade máxima.

5.5.2 Quantidade de Movimento x PSNR

Para que possamos atingir nossos objetivos devemos, agora, comparar a variável dependente PSNR agrupada pelas categorias de movimento que utilizamos em nossos testes, e assim verificar a influência da quantidade de movimento na qualidade do vídeo recebido de acordo com a variável PSNR. Assim, foram gerados os intervalos de confiança.

Figura 5.16: Gráfico de intervalos de confiança PSNR x Quantidade de Movimento.



No gráfico da Figura 5.16, é possível notar que quantidade de movimento MR está com o seu intervalo bem abaixo no gráfico, o que diz que ela é a quantidade de movimento que mais foi afetada pela transmissão, perdendo mais qualidade de imagem que as outras. Nada se pode afirmar sobre a relação entre MD e MM, já que os pontos centrais dos intervalos estão englobados pelos dois intervalos. Assim, faz-se necessária a execução de testes para verificação da influência da quantidade de movimento na qualidade do vídeo recebido.

Os resultados dos testes de normalidade são mostrados na Tabela B.9, a partir dos quais é possível dizer que os dados não seguem uma distribuição normal.

Na Tabela C.9, é mostrado o resultado da execução dos testes no R. É possível observar que a quantidade MR possui p-valor 2^{-16} quando comparado ao MM e 2^{-16} quando comparado ao MD, os dois valores estão bem abaixo do valor $\alpha = 0,05$ e assim devemos rejeitar a Hipótese nula 9 e aceitar a alternativa 10, ou seja, os vídeos recebidos da categoria MR possuem um PSNR menor ao serem transmitidos (de acordo com o intervalo de confiança). Quanto à comparação entre MM e MD, o valor de 0,016 faz com que seja rejeitada a nula e seja aceita a alternativa, ou seja, os vídeos recebidos também não possuem a mesma média, estatisticamente.

Para verificar qual conteúdo obteve os melhores resultados na métrica PSNR, precisamos realizar mais um teste que considera as seguintes Hipóteses:

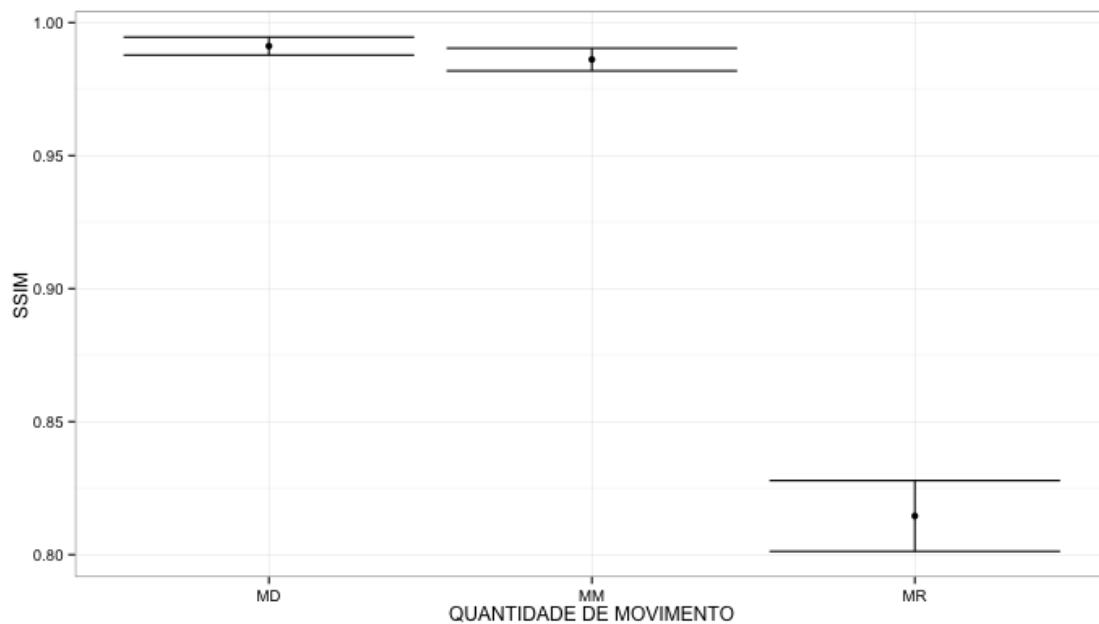
Na Tabela D.5, é mostrado o resultado da execução dos testes no R. Observamos que com um p-valor maior que $\alpha = 0,05$ quando comparada a quantidade de movimento MM, o MR obteve maior perda de qualidade e quando comparado a quantidade de movimento MD, obteve um p-valor de 2^{-16} que é bem inferior ao $\alpha = 0,05$, ou seja a quantidade de movimento MR, obteve maior perda na qualidade do vídeo recebido. Quando comparamos a quantidade MD com a quantidade MM, tem-se um p-valor de 0,016, ou seja, a quantidade MD obteve uma maior qualidade de vídeo recebido.

5.5.3 Quantidade de Movimento x SSIM

Para que possamos atingir nossos objetivos devemos, também, comparar a variável dependente SSIM agrupada pelas categorias de movimento que utilizamos em nossos testes, e assim verificar a influência da quantidade de movimento na qualidade do vídeo recebido de acordo com a variável SSIM. Assim, foram gerados os intervalos de confiança.

No gráfico da Figura 5.17, é possível notar que quantidade de movimento MR está com o seu intervalo bem abaixo no gráfico, o que diz que ela é a quantidade de movimento que mais foi afetada pela transmissão, perdendo mais qualidade de imagem que as outras. Nada se pode afirmar sobre a relação entre MD e MM, já que os pontos centrais dos intervalos estão englobados pelos dois intervalos. Assim, faz-se necessária

Figura 5.17: Gráfico de intervalos de confiança SSIM x Quantidade de Movimento.



a execução de testes para verificação da influência da quantidade de movimento na qualidade do vídeo recebido.

Os resultados dos testes de normalidade são mostrados na Tabela B.11, a partir dos quais é possível dizer que os dados não seguem uma distribuição normal.

Na Tabela C.11, é mostrado o resultado da execução dos testes no R. É possível observar que a quantidade MR possui p-valor 2^{-16} quando comparado ao MM e 2^{-16} quando comparado ao MD, os dois valores estão bem abaixo do valor $\alpha = 0,05$ e assim devemos rejeitar a Hipótese nula 9 e aceitar a alternativa 10, ou seja, os vídeos recebidos da categoria MR possuem um SSIM menor ao serem transmitidos (de acordo com o intervalo de confiança). Quanto à comparação entre MM e MD, o valor de 0,021 faz com que seja rejeitada a nula e seja aceita a alternativa, ou seja, os vídeos recebidos também não possuem a mesma média, estatisticamente.

Para verificar qual a quantidade de movimento que obteve os melhores resultados, realizamos mais um teste que considera as seguintes hipóteses:

Na Tabela D.7, é mostrado o resultado da execução dos testes no R. É possível observar que na comparação entre MM e MD, o valor de 0,021 faz com que seja rejeitada a nula e seja aceita a alternativa, ou seja, os vídeos da quantidade de movimento MD possuem uma média maior que os da quantidade de movimento MM.

5.6 Compressão dos Vídeos e Quantidade de Pacotes

Observar o tamanho do arquivo de cada vídeo dá uma noção da quantidade de compressão envolvida em cada formato (*codec*). A compressão é uma forma de diminuir o espaço ocupado por um arquivo, tentando ao máximo não prejudicar sua qualidade de exibição. No caso de arquivos de texto, não existe perda na qualidade (e consequentemente na informação) do arquivo comprimido, mas no caso de vídeos podemos ter perda de qualidade de imagem com o objetivo de obter um arquivo de tamanho menor.

Tabela 5.1: Tamanho dos arquivos por *codec* utilizado.

Vídeo	Categoria	YUV	H.264	MPEG-4	HEVC
<i>Akiyo</i>	MD	45,6 MB	151 KB	487 KB	64 KB
<i>Silent</i>	MD	45,6 MB	391 KB	495 KB	147 KB
<i>Mother-daughter</i>	MD	4,6 MB	256 KB	466 KB	95 KB
<i>Hall</i>	MM	45,6 MB	437 KB	479 KB	83 KB
<i>News</i>	MM	45,6 MB	312 KB	472 KB	135 KB
<i>Tempete</i>	MM	39,5 MB	1,1 MB	427 KB	305 KB
<i>Bus</i>	MR	22,8 MB	569 KB	328 KB	242 KB
<i>Football</i>	MR	39,5 MB	1,3 MB	461 KB	559 KB
<i>Stefan</i>	MR	45,6 MB	349 KB	229 KB	576 KB

A Tabela 5.1, mostra o tamanho de cada arquivo de cada vídeo formatos encontrados para testes (YUV) e transcodificados para cada formato de compressão que utilizamos testes. É possível ver que todos os formatos de compressão, comprimem significativamente em relação ao formato não comprimido (YUV). Isto diminui a quantidade de pacotes necessária para transmissão do vídeo e a probabilidade de um pacote ser descartado ou perdido durante a transmissão, até mesmo a probabilidade de que seu atraso seja suficiente para que ele seja descartado por não precisar ser exibido mais.

Pode-se verificar na Tabela 5.1 que, apesar do HEVC conseguir uma melhor compressão arquivos com movimento (diminuindo consideravelmente seu tamanho), ele necessita de um tamanho de arquivo maior para os vídeos que possuem mais movi-

mento.

Com o tamanho dos arquivos variando, varia também a quantidade de pacotes necessária para transmissão do mesmo. A Tabela 5.2, mostra a quantidade de pacotes necessária para enviar um vídeo por completo formatos de compressão escolhidos.

Tabela 5.2: Quantidade de pacotes por vídeo e codec utilizado.

Vídeo	Categoria	H.264	MPEG-4	HEVC
<i>Akiyo</i>	MD	108	342	53
<i>Silent</i>	MD	275	352	107
<i>Mother-daughter</i>	MD	182	330	74
<i>Hall</i>	MM	305	339	66
<i>News</i>	MM	220	346	100
<i>Tempete</i>	MM	734	300	217
<i>Bus</i>	MR	396	231	170
<i>Football</i>	MR	890	324	391
<i>Stefan</i>	MR	242	161	404

Capítulo 6

Considerações Finais e Sugestões para Pesquisas Futuras

6.1 Considerações Finais

Com base nas hipóteses definidas na seção 4.2 e nos resultados expostos nos capítulos anteriores, é possível chegar às seguintes considerações:

- Sobre a avaliação de perda de qualidade: os vídeos tem sua qualidade de imagem degradada durante uma transmissão por uma rede sem fio;
- Sobre a avaliação do conteúdo: o conteúdo teve bastante influência na qualidade do vídeo recebido. Ficou constatado também que, quanto mais movimento um vídeo apresentar, maior a possibilidade de ter sua qualidade de imagem degradada;
- Sobre a avaliação dos *codecs*: o HEVC apresentou resultados superiores (para a base de vídeos escolhida) tanto ao H.264 quando ao MPEG-4 em todas as métricas relacionadas à qualidade de vídeo analisadas. Na comparação do H.264 com o MPEG-4, vimos que não existe diferença entre os dois formatos (para a base de vídeos escolhida) com relação à métrica PSNR, mas que o H.264 é superior nas outras duas métricas escolhidas;

6.2 Sugestões para Pesquisas Futuras

A pesquisa obteve as conclusões, principalmente, relacionadas ao formato de compressão HEVC, que se mostrou o melhor formato para transmissões em redes sem fio (o que obteve menos perda de qualidade da imagem do vídeo recebido). Apesar de um desempenho bom comparado aos outros dois formatos, ainda temos margem para melhorar visto que todos os *codecs* apresentaram perda de qualidade devido ao fato de serem transmitidos via rede sem fio.

Para avaliar uma possível melhoria no desempenho ao tentar fornecer uma garantia melhor de entrega dos dados, pode-se realizar um estudo, em que a rede utilize protocolos como *Resource Reservation Protocol* (RSVP) ou mesmo a arquitetura de Serviços Diferenciados, que são propostos para melhorar a qualidade de serviços na Internet e avaliar a qualidade dos vídeos recebidos dentre os formatos de vídeos mais usados atualmente.

Pode-se pensar em um monitoramento da rede enquanto o vídeo é transmitido para diminuir ou aumentar a qualidade do vídeo enviado, fazendo com que o serviço não seja interrompido devido à qualidade da rede.

Bibliografia

- [1] Andrew S. Tanenbaum and David J. Wetherall. *Computer Networks*. Pearson, 2010.
- [2] Sanjit K. Mitra. *Digital Signal Processing*. McGraw Hill, 2010.
- [3] Yao Wang, Jorn Ostermann, and Ya-Quin Zhang. *Video Processing and Communications*. 2001.
- [4] Jack Chan, Kai Zeng, Prasant Mohapatra, Sung-Ju Lee, and Sujata Banerjee. Metrics for evaluating video streaming quality in lossy iee 802.11 wireless networks. *Proceedings IEEE INFOCOM*, pages 1–9, 2010.
- [5] Weber contrast. Disponível em: [http://en.wikipedia.org/wiki/Contrast_\(vision\)#Weber_contrast](http://en.wikipedia.org/wiki/Contrast_(vision)#Weber_contrast). Acesso em 10/01/2015.
- [6] Canny edge. Disponível em: http://en.wikipedia.org/wiki/Canny_edge_detector. Acesso em 10/01/2015.
- [7] Transformada de hough. Disponível em: http://en.wikipedia.org/wiki/Hough_transform. Acesso em 10/01/2015.
- [8] Mpeg-4. Disponível em: . Acesso em 10/10/2015.
- [9] H.264. Disponível em: https://en.wikipedia.org/wiki/H.264/MPEG-4_AVC. Acesso em 10/10/2015.
- [10] Hevc. Disponível em: <http://x265.org/hevc-h265/>. Acesso em 10/10/2015.
- [11] Niveditta Thakur and Swapna Devi. A new method for color image quality assessment. *International Journal of Computer Applications*, 15:10–17, 2011.

-
- [12] Dante Coaquira Begazo, Demóstenes Zegarra Rodriguez, and Miguel Arjona Ramírez. Avaliação de qualidade de vídeo sobre uma rede ip usando métricas objetivas. *Revista Iberoamericana de Sistemas, Cibernética e Informática*, 8:25–29, 2011.
- [13] A. Srivastava, A. B. Lee, E. P. Simoncelli, and S.-C. Zhu. On advances in statistical modeling of natural images. *Journal of Mathematical Imaging and Vision*, 18:17 – 33, 2003.
- [14] Hamid R. Sheikh and Alan C. Bovik. Image information and visual quality. *Image Processing, IEEE Transactions on*, 15, 2006.
- [15] Ogê Marques Filho and Hugo Vieira Neto. *Processamento Digital de Imagens*. Brasport, 1999.
- [16] Stuart J. Russell and Peter Norvig. *Artificial Intelligence A Modern Approach*. Prentice Hall, 2009.
- [17] Weka machine learning software in java. Disponível em: <http://www.cs.waikato.ac.nz/ml/weka/>. Acesso em 10/01/2015.
- [18] Asiya Khan, Lingfen Sun, and Emmanuel Ifeachor. Impact of video content on video quality for video over wireless networks. *Fifth International Conference on Autonomic and Autonomous Systems*, pages 277 – 282, 2009.
- [19] Deyun Gao, Jianfei Cai, and Zhihai He Paul Bao. Mpeg-4 video streaming quality evaluation in ieee 802.11e wlans. *IEEE International Conference on Image Processing*, 1:197–200, 2005.
- [20] Kwangjin Choi, Jun kyun Choi, Jae Hwan Hong, Gyeong Ju Min, and Jongkuk Lee. Comparison of video streaming quality measurement methodologies. *10th International Conference on Advanced Communication Technology*, 2:993–996, 2008.
- [21] Soekris net5501 single board computer. Disponível em: <http://soekris.com/products/net5501.html>. Acesso em 10/10/2015.

-
- [22] Ubiquiti extreme range 2 radio transmissor. Disponível em: http://dl.ubnt.com/xr2_datasheet.pdf. Acesso em 10/10/2015.
- [23] Voyage linux. Disponível em: <http://linux.voyage.hk/>. Acesso em 10/10/2015.
- [24] Ffmpeg. Disponível em: <http://www.ffmpeg.org/>. Acesso em 10/10/2015.
- [25] Iperf. Disponível em: <https://iperf.fr/>. Acesso em 10/10/2015.
- [26] Iperf 3. Disponível em: <http://software.es.net/iperf/>. Acesso em 10/10/2015.
- [27] Vqmt: Video quality measurement tool. Disponível em: <http://mmspg.epfl.ch/vqmt>. Acesso em 10/10/2015.
- [28] Opencv. Disponível em: <http://opencv.org/>. Acesso em 10/10/2015.
- [29] Tcpdump. Disponível em: <http://www.tcpdump.org/>. Acesso em 10/10/2015.
- [30] Claes Wohlin, Claes Wohlin Per Runeson, Martin Host, Magnus C. Ohlsson, Bjorn Regnell, and Anders Wesslén. *Experimentation in Software Engineering*. Springer, 2012.
- [31] Ntp. Disponível em: <http://www.ntp.org/>. Acesso em 10/10/2015.
- [32] Douglas Montgomery and George Runger. *Applied Statistics and Probability for Engineers*. John Wiley & Sons, 2011.
- [33] Video coders - sequências de vídeos yuv. Disponível em: <http://videocoders.com/yuv.html>. Acesso em 10/01/2015.
- [34] Arizona state university - sequências de vídeos yuv. Disponível em: <http://trace.eas.asu.edu/yuv/>. Acesso em 10/01/2015.

Apêndice A

Códigos para Extração de Características dos Vídeos

A.1 Cálculo do Contraste de Weber

Código Fonte A.1: Código C++ para Cálculo do Contraste de Weber

```
1 double weberContrast(IplImage* input) {
2     double contrast = 0;
3
4     cvCvtColor(input, input, CV_BGR2Lab);
5
6     CvMat* background = cvCreateMat(input->height, input->width, CV_8UC1)
7     ;
8     cvSplit(input, background, NULL, NULL, NULL);
9
10    cvSmooth(background, background, CV_BLUR, 5);
11
12    for (int i = 0; i < input->height; i++) {
13        for (int j = 0; j < input->width; j++) {
14            if (static_cast<int>(Mat(background).at<uchar>(i, j)) > 0) {
15                contrast += (static_cast<double>(Mat(input).at<uchar>(i, j
16                ,0)) - static_cast<double>(Mat(background).at<uchar>(i
17                ,j)))
18                / static_cast<double>(Mat(background).at<uchar>(i, j));
19            }
```

```
17     }
18 }
19     contrast /= (input->width * input->height);
20
21     return contrast;
22 }
```

A.2 Cálculo da Soma de Diferenças Absolutas

Código Fonte A.2: Código C++ para Cálculo da Soma de Diferenças Absolutas

```
1 double sumOfAbsoluteDifferences(Video video) {
2     FILE *fin;
3     YUV_Capture *cap = new YUV_Capture();
4     enum YUV_ReturnValue ret;
5
6     IplImage *prevImage = NULL;
7     IplImage *image = NULL;
8     IplImage *diff = NULL;
9     fin = fopen(video.get_dir().c_str(), "rb");
10    if (!fin)
11    {
12        fprintf(stderr, "error: unable to open file: %s\n", video.
13            get_name().c_str());
14        return 1;
15    }
16    ret = YUV_init(fin, video.get_width(), video.get_height(), cap);
17    assert(ret == YUV_OK);
18
19    double frames = 0;
20    double sum = 0;
21    while ((ret = YUV_read(cap)) != YUV_EOF && ret != YUV_IO_ERROR)
22    {
23        if (image == NULL) {
24            image = cvCreateImage(cvSize(video.get_width(), video.
25                get_height()), IPL_DEPTH_8U, 3);
26            assert(image);
```

```
25         cvCvtColor(cap->yrcrb, image, CV_YCrCb2BGR);
26     } else {
27         prevImage = cvCreateImage(cvSize(video.get_width(), video.
28             get_height()), IPL_DEPTH_8U, 3);
29         cvCopy(image, prevImage);
30         image = cvCreateImage(cvSize(video.get_width(), video.
31             get_height()), IPL_DEPTH_8U, 3);
32         assert(image);
33         cvCvtColor(cap->yrcrb, image, CV_YCrCb2BGR);
34     }
35
36     if (diff == NULL) {
37         diff = cvCreateImage(cvSize(video.get_width(), video.
38             get_height()), IPL_DEPTH_8U, 3);
39     }
40
41     if (prevImage != NULL) {
42         cvAbsDiff(image, prevImage, diff);
43         for (int i = 0; i < diff->height; i++) {
44             for (int j = 0; j < diff->width; j++) {
45                 sum += static_cast<double>(Mat(diff).at<uchar>(i, j));
46             }
47         }
48     }
49     frames++;
50 }
```

A.3 Cálculo do Brilho

Código Fonte A.3: Código C++ para Cálculo do Brilho

```
1 double brilho(Video video) {
2     FILE *fin;
3     YUV_Capture *cap = new YUV_Capture();
```

```
4     enum YUV_ReturnValue ret;
5
6     IplImage *frame = NULL;
7     IplImage *prevImage = NULL;
8     IplImage *image = NULL;
9     fin = fopen(video.get_dir().c_str(), "rb");
10    if (!fin)
11    {
12        fprintf(stderr, "error: unable to open file: %s\n", video.
13            get_name().c_str());
14        return 1;
15    }
16    ret = YUV_init(fin, video.get_width(), video.get_height(), cap);
17    assert(ret == YUV_OK);
18
19    double frames = 0;
20    double sum = 0;
21    while ((ret = YUV_read(cap)) != YUV_EOF && ret != YUV_IO_ERROR)
22    {
23        frame = cvCreateImage(cvSize(video.get_width(), video.get_height()
24            ), IPL_DEPTH_8U, 3);
25        cvCvtColor(cap->yrcb, frame, CV_YCrCb2BGR);
26        if (image == NULL) {
27            image = cvCreateImage(cvSize(video.get_width(), video.
28                get_height()), IPL_DEPTH_8U, 1);
29            assert(image);
30            cvCvtColor(frame, image, CV_BGR2GRAY);
31        } else {
32            prevImage = cvCreateImage(cvSize(video.get_width(), video.
33                get_height()), IPL_DEPTH_8U, 1);
34            cvCopy(image, prevImage);
35            image = cvCreateImage(cvSize(video.get_width(), video.
36                get_height()), IPL_DEPTH_8U, 1);
37            assert(image);
38            cvCvtColor(frame, image, CV_BGR2GRAY);
39        }
40    }
```

```
36     if (prevImage != NULL) {
37         double prevMatBri = cvMean(prevImage);
38         double imgMatBri = cvMean(image);
39         sum += abs(imgMatBri - prevMatBri);
40     }
41     frames++;
42 }
43
44 return sum/frames;
45 }
```

A.4 Cálculo do Embaçamento

Código Fonte A.4: Código C++ para Cálculo do Embaçamento

```
1 double blurCannyHough(Video video) {
2     FILE *fin;
3
4     YUV_Capture *cap = new YUV_Capture();
5     enum YUV_ReturnValue ret;
6     LineUtils lu = LineUtils();
7
8     IplImage *frame = NULL;
9     IplImage *src = NULL;
10
11     vector<float> gradients;
12
13     fin = fopen(video.get_dir().c_str(), "rb");
14     if (!fin)
15     {
16         fprintf(stderr, "error: unable to open file: %s\n", video.
17             get_name().c_str());
18         return 1;
19     }
20     ret = YUV_init(fin, video.get_width(), video.get_height(), cap);
21     assert(ret == YUV_OK);
```



```

22     double frames = 0;
23     double sum = 0;
24     while ((ret = YUV_read(cap)) != YUV_EOF && ret != YUV_IO_ERROR)
25     {
26
27         frame = cvCreateImage(cvSize(video.get_width(), video.get_height()
28             ), IPL_DEPTH_8U, 3);
29         cvCvtColor(cap->yrcb, frame, CV_YCrCb2BGR);
30         src = cvCreateImage(cvSize(video.get_width(), video.get_height()),
31             IPL_DEPTH_8U, 1);
32         cvCvtColor(frame, src, CV_BGR2GRAY);
33
34         int widthRoiSize = video.get_width() / 9;
35         int heighthRoiSize = video.get_height() / 9;
36         for(int x = 0; x < src->width / widthRoiSize; ++x) {
37             for(int y = 0; y < src->height / heighthRoiSize; ++y) {
38                 cvSetImageROI(src, cvRect(x*widthRoiSize, y*heighthRoiSize
39                     , widthRoiSize, heighthRoiSize));
40
41                 IplImage *cropSource = cvCreateImage(cvGetSize(src), src
42                     ->depth, src->nChannels);
43
44                 cvCopy(src, cropSource, NULL);
45
46                 Mat dst, cdst;
47                 Canny(Mat(src), dst, 50, 200, 3);
48                 cvtColor(dst, cdst, CV_GRAY2BGR);
49
50                 vector<Vec4i> lines;
51                 HoughLinesP(dst, lines, 1, 0.01, 5);
52                 for( size_t i = 0; i < lines.size(); i++)
53                 {
54                     Vec4i l = lines[i];
55                     line( cdst, Point(l[0], l[1]), Point(l[2], l[3]),
56                         Scalar(0,0,255));
57                 }
58             }
59         }

```

```
54         int pixelASize = dst.rows*dst.cols;
55         vector<Point> points;
56         vector<Point> pointsR;
57         for (size_t i = 0; i < lines.size(); i++) {
58             Vec4i l = lines[i];
59             points.push_back(Point(l[0], l[1]));
60             points.push_back(Point(l[2], l[3]));
61
62         }
63
64         Point *pointsA = &points[0];
65         float pixels[pixelASize];
66         for (int j = 0; j < pixelASize; j++) {
67             pixels[j] = static_cast<double>(Mat(dst).at<uchar>(j
68                 ));
69
70         }
71
72         int maxIndex;
73         int minIndex;
74         lu.GetMinMaxIndicies(pixels, pixelASize, minIndex,
75             maxIndex);
76
77         if (pointsA != NULL)
78             gradients.push_back(lu.CalculateGradient(pointsA,
79                 pixels, pixelASize, minIndex, maxIndex));
80         else
81             gradients.push_back(0);
82
83         cvResetImageROI(src);
84     }
85     sum += lu.GetAverage(&gradients[0], gradients.size());
86     frames++;
87 }
88 return sum/frames;
89 }
```

Apêndice B

Análise de Normalidade

Tabela B.1: Resultados dos testes de normalidade Atraso x *Codec*.

<i>Codec</i>	p-Valor
H.264	$2, 2^{-16}$
MPEG-4	$2, 2^{-16}$
HEVC	$2, 2^{-16}$

Tabela B.2: Resultados dos testes de normalidade Atraso x Potência.

Potência	p-Valor
1dBm	$2, 2^{-16}$
7dBm	$2, 2^{-16}$
14dBm	$2, 2^{-16}$

Tabela B.3: Resultados dos testes de normalidade Atraso x Porcentagem de Ocupação.

Porcentagem de Ocupação	p-Valor
0%	$2, 2^{-16}$
30%	$2, 2^{-16}$
60%	$2, 2^{-16}$
90%	$2, 2^{-16}$

Tabela B.4: Resultados dos testes de normalidade Atraso x *Codec*.

<i>Codec</i>	p-Valor
H.264	$2, 2^{-16}$
MPEG-4	$2, 2^{-16}$
HEVC	$2, 2^{-16}$

Tabela B.5: Resultados dos testes de normalidade Atraso x Potência.

Potência	p-Valor
1dBm	$2, 2^{-16}$
7dBm	$2, 2^{-16}$
14dBm	$2, 2^{-16}$

Tabela B.6: Resultados dos testes de normalidade Perda de pacote x Porcentagem de Ocupação.

Porcentagem de Ocupação	p-Valor
0%	$2, 2^{-16}$
30%	$2, 2^{-16}$
60%	$2, 2^{-16}$
90%	$2, 2^{-16}$

Tabela B.7: Resultados dos testes de normalidade VIFP x Quantidade de Movimento.

Quantidade de Movimento	p-Valor
MD	$2, 2^{-16}$
MM	$2, 2^{-16}$
MR	$2, 2^{-16}$

Tabela B.8: Resultados dos testes de normalidade VIFP x *Codec*.

<i>Codec</i>	p-Valor
H.264	$2, 2^{-16}$
HEVC	$2, 2^{-16}$
MPEG-4	$2, 2^{-16}$

Tabela B.9: Resultados dos testes de normalidade PSNR x Quantidade de Movimento.

Quantidade de Movimento	p-Valor
MD	$2, 2^{-16}$
MM	$2, 2^{-16}$
MR	$2, 2^{-16}$

Tabela B.10: Resultados dos testes de normalidade PSNR x *Codec*.

<i>Codec</i>	p-Valor
H.264	$2, 2^{-16}$
HEVC	$2, 2^{-16}$
MPEG-4	$2, 2^{-16}$

Tabela B.11: Resultados dos testes de normalidade SSIM x Quantidade de Movimento.

Quantidade de Movimento	p-Valor
MD	$2, 2^{-16}$
MM	$2, 2^{-16}$
MR	$2, 2^{-16}$

Tabela B.12: Resultados dos testes de normalidade SSIM x *Codec*.

<i>Codec</i>	p-Valor
H.264	$2, 2^{-16}$
HEVC	$2, 2^{-16}$
MPEG-4	$2, 2^{-16}$

Tabela B.13: Resultados dos testes de normalidade Atraso x *Codec* x Quantidade de Movimento.

<i>Codec</i>	Quantidade Movimento	p-Valor
H.264	MD	$2, 2^{-16}$
H.264	MM	$2, 2^{-16}$
H.264	MR	$2, 2^{-16}$
HEVC	MD	$2, 2^{-16}$
HEVC	MM	$2, 2^{-16}$
HEVC	MR	$2, 2^{-16}$
MPEG-4	MD	$2, 2^{-16}$
MPEG-4	MM	$2, 2^{-16}$
MPEG-4	MR	$2, 2^{-16}$

Tabela B.14: Resultados dos testes de normalidade Perda de Pacote x *Codec* x Quantidade de Movimento.

<i>Codec</i>	Quantidade Movimento	p-Valor
H.264	MD	$2, 2^{-16}$
H.264	MM	$2, 2^{-16}$
H.264	MR	$2, 2^{-16}$
HEVC	MD	$2, 2^{-16}$
HEVC	MM	$2, 2^{-16}$
HEVC	MR	$2, 2^{-16}$
MPEG-4	MD	$2, 2^{-16}$
MPEG-4	MM	$2, 2^{-16}$
MPEG-4	MR	$2, 2^{-16}$

Tabela B.15: Resultados dos testes de normalidade VIFP x *Codec* x Quantidade de Movimento.

<i>Codec</i>	Quantidade Movimento	p-Valor
H.264	MD	$2, 2^{-16}$
H.264	MM	$2, 2^{-16}$
H.264	MR	$2, 2^{-16}$
HEVC	MD	$2, 2^{-16}$
HEVC	MM	$2, 2^{-16}$
HEVC	MR	$2, 2^{-16}$
MPEG-4	MD	$2, 2^{-16}$
MPEG-4	MM	$2, 2^{-16}$
MPEG-4	MR	$2, 2^{-16}$

Tabela B.16: Resultados dos testes de normalidade PSNR x *Codec* x Quantidade de Movimento.

<i>Codec</i>	Quantidade Movimento	p-Valor
H.264	MD	$2, 2^{-16}$
H.264	MM	$2, 2^{-16}$
H.264	MR	$2, 2^{-16}$
HEVC	MD	$2, 2^{-16}$
HEVC	MM	$2, 2^{-16}$
HEVC	MR	$2, 2^{-16}$
MPEG-4	MD	$2, 2^{-16}$
MPEG-4	MM	$2, 2^{-16}$
MPEG-4	MR	$2, 2^{-16}$

Tabela B.17: Resultados dos testes de normalidade SSIM x *Codec* x Quantidade de Movimento.

<i>Codec</i>	Quantidade Movimento	p-Valor
H.264	MD	$2, 2^{-16}$
H.264	MM	$2, 2^{-16}$
H.264	MR	$2, 2^{-16}$
HEVC	MD	$2, 2^{-16}$
HEVC	MM	$2, 2^{-16}$
HEVC	MR	$2, 2^{-16}$
MPEG-4	MD	$2, 2^{-16}$
MPEG-4	MM	$2, 2^{-16}$
MPEG-4	MR	$2, 2^{-16}$

Apêndice C

Testes Não-Paramétricos de Igualdade

Tabela C.1: Resultados dos testes de comparação de igualdade SSIM x *Codec*.

x / y	H.264	HEVC
HEVC	2^{-16}	-
MPEG-4	0,016	2^{-16}

Tabela C.2: Resultados dos testes de comparação Atraso x Potência.

x / y	14dBm	1dBm
1dBm	0,57	-
7dBm	$2, 2^{-16}$	$2, 2^{-16}$

Tabela C.3: Resultados dos testes de comparação Atraso x Porcentagem de Ocupação.

x / y	0%	30%	60%
30%	$< 2^{-16}$	-	-
60%	$< 2^{-16}$	$< 2^{-16}$	-
90%	$< 2^{-16}$	$< 2^{-16}$	$< 2^{-16}$

Tabela C.4: Resultados dos testes de comparação Perda de Pacotes x *Codec*.

x / y	H.264	HEVC
HEVC	0,74	-
MPEG-4	0,27	0,51

Tabela C.5: Resultados dos testes de comparação Perda de Pacotes x Potência.

x / y	14dBm	1dBm
1dBm	0,91	-
7dBm	0,42	0,42

Tabela C.6: Resultados dos testes de comparação de igualdade Perda de Pacote x Porcentagem de Ocupação.

x / y	0%	30%	60%
30%	0,42	-	-
60%	0,25	0,02	-
90%	0,01	0,00	0,42

Tabela C.7: Resultados dos testes de comparação de igualdade VIFP x Quantidade de Movimento.

x / y	MM	MR
MR	2^{-16}	-
MD	0,02	2^{-16}

Tabela C.8: Resultados dos testes de comparação de igualdade VIFP x *Codec*.

x / y	H.264	HEVC
HEVC	2^{-16}	-
MPEG-4	0,03	2^{-16}

Tabela C.9: Resultados dos testes de comparação de igualdade PSNR x Quantidade de Movimento.

x / y	MM	MR
MR	2^{-16}	-
MD	0,016	2^{-16}

Tabela C.10: Resultados dos testes de comparação de igualdade PSNR x *Codec*.

x / y	H.264	HEVC
HEVC	2^{-16}	-
MPEG-4	0,061	2^{-16}

Tabela C.11: Resultados dos testes de comparação de igualdade SSIM x Quantidade de Movimento.

x / y	MM	MR
MR	2^{-16}	-
MD	0,021	2^{-16}

Tabela C.12: Resultados dos testes de comparação de igualdade Atraso x Quantidade de Movimento x *Codec*.

x / y	H.264	H.264	H.264	HEVC	HEVC	HEVC	MPEG-4	MPEG-4
	MD	MM	MR	MD	MM	MR	4 MD	4 MM
H.264_MM	$3,1^{-16}$	-	-	-	-	-	-	-
H.264_MR	$2,1^{-19}$	1	-	-	-	-	-	-
HEVC_MD	$4,2^{-21}$	0,00054	0,0028	-	-	-	-	-
HEVC_MM	$1,2^{-30}$	$1,8^{-16}$	$1,8^{-12}$	$1,5^{-10}$	-	-	-	-
HEVC_MR	$9,5^{-79}$	$2,3^{-66}$	$1,1^{-69}$	$3,6^{74}$	$1,7^{-34}$	-	-	-
MPEG-4_MD	0,00033	0,0038	0,0038	$3,6^{-6}$	$8,5^{-14}$	$6,3^{-61}$	-	-
MPEG-4_MM	$1,7^{-12}$	1	1	0,00055	$4,3^{-13}$	$3,1^{-56}$	0,044	-
MPEG-4_MR	$8,3^{-51}$	$1,2^{-24}$	$3,6^{-24}$	$4,3^{-13}$	0,0038	0,032	$1,7^{-38}$	$2,3^{-30}$

Tabela C.13: Resultados dos testes de comparação de igualdade Perda de Pacote x Quantidade de Movimento x *Codec*.

x / y	H.264 MD	H.264 MM	H.264 MR	HEVC MD	HEVC MM	HEVC MR	MPEG-4 MD	MPEG-4 MM
H.264_MM	1	-	-	-	-	-	-	-
H.264_MR	1	1	-	-	-	-	-	-
HEVC_MD	1	1	1	-	-	-	-	-
HEVC_MM	1	1	1	1	-	-	-	-
HEVC_MR	0,82	1	1	0,089	1	-	-	-
MPEG-4_MD	1	1	1	1	1	1	-	-
MPEG-4_MM	1	1	1	1	1	1	1	-
MPEG-4_MR	1	1	1	0,26	1	1	1	1

Tabela C.14: Resultados dos testes de comparação de igualdade SSIM x Quantidade de Movimento x *Codec*.

x / y	H.264 MD	H.264 MM	H.264 MR	HEVC MD	HEVC MM	HEVC MR	MPEG-4 MD	MPEG-4 MM
H.264_MM	0,39	-	-	-	-	-	-	-
H.264_MR	$3,5^{-50}$	$4,4^{-46}$	-	-	-	-	-	-
HEVC_MD	1	0.1	$4,7^{-49}$	-	-	-	-	-
HEVC_MM	1	1	1^{-48}	1	-	-	-	-
HEVC_MR	0,00041	0.1	$7,2^{-36}$	0,0001	0,0037	-	-	-
MPEG-4_MD	1	1	$2,5^{-45}$	0,51	1	0,022	-	-
MPEG-4_MM	0,94	1	$6,2^{-43}$	0,1	1	0,08	1	-
MPEG-4_MR	$2,7^{-53}$	$2,4^{-49}$	0,08	$3,1^{-52}$	1^{-51}	$4,4^{-40}$	$1,2^{-47}$	$3,1^{-48}$

Tabela C.15: Resultados dos testes de comparação de igualdade VIFP x Quantidade de Movimento x *Codec*.

x / y	H.264	H.264	H.264	HEVC	HEVC	HEVC	MPEG	MPEG
	MD	MM	MR	MD	MM	MR	4 MD	4 MM
H.264_MM	0,37	-	-	-	-	-	-	-
H.264_MR	$6,2^{-49}$	$6,3^{-46}$	-	-	-	-	-	-
HEVC_MD	1	0,12	$5,1^{-49}$	-	-	-	-	-
HEVC_MM	1	1	$1,4^{-48}$	1	-	-	-	-
HEVC_MR	0,00054	0,14	$1,2^{-35}$	0,00011	0,01	-	-	-
MPEG-4_MD	1	1	$2,5^{-45}$	0,6	1	0,016	-	-
MPEG-4_MM	1	1	$5,7^{-43}$	0,14	1	0,092	1	-
MPEG-4_MR	4^{-52}	$6,1^{-49}$	0,096	$3,3^{-49}$	$2,1^{-51}$	$8,4^{-40}$	$1,2^{-47}$	$3,4^{-48}$

Tabela C.16: Resultados dos testes de comparação de igualdade PSNR x Quantidade de Movimento x *Codec*.

x / y	H.264	H.264	H.264	HEVC	HEVC	HEVC	MPEG	MPEG
	MD	MM	MR	MD	MM	MR	4 MD	4 MM
H.264_MM	0,58	-	-	-	-	-	-	-
H.264_MR	$8,7^{-49}$	1^{-45}	-	-	-	-	-	-
HEVC_MD	1	0,08	$5,1^{-49}$	-	-	-	-	-
HEVC_MM	1	1	$1,4^{-48}$	1	-	-	-	-
HEVC_MR	0,00061	0,17	$1,6^{-35}$	$6,9^{-5}$	0,0064	-	-	-
MPEG-4_MD	1	1	$2,6^{-45}$	0,68	1	0,015	-	-
MPEG-4_MM	1	1	$5,7^{-43}$	0,15	1	0,087	1	-
MPEG-4_MR	$5,9^{-52}$	$1,4^{-48}$	0,15	$3,6^{-52}$	$2,8^{-51}$	$1,5^{-39}$	$1,6^{-47}$	$4,4^{-48}$

Apêndice D

Testes Não-Paramétricos de Alternativa Maior

Tabela D.1: Resultados dos testes de comparação Atraso x *Codec*.

x / y	H.264	HEVC
HEVC	$< 2^{-16}$	-
MPEG-4	$< 2^{-16}$	1,000

Tabela D.2: Resultados dos testes de comparação Perda de Pacote x Porcentagem de Ocupação.

x / y	0%	30%
60%	-	0,01270
90%	0,00521	0,00017

Tabela D.3: Resultados dos testes de comparação VIFP x Quantidade de Movimento.

x / y	MM
MD	0,016

Tabela D.4: Resultados dos testes de comparação VIFP x *Codec*.

x / y	H.264	HEVC
HEVC	2^{-16}	-
MPEG-4	1,000	1,000

Tabela D.5: Resultados dos testes de comparação PSNR x Quantidade de Movimento.

x / y	MM	MR
MR	1,000	-
MD	0,016	2^{-16}

Tabela D.6: Resultados dos testes de comparação PSNR x *Codec*.

x / y	H.264	HEVC
HEVC	2^{-16}	-
MPEG-4	-	1,000

Tabela D.7: Resultados dos testes de comparação de igualdade SSIM x Quantidade de Movimento.

x / y	MM
MD	0,021

Tabela D.8: Resultados dos testes de comparação SSIM x *Codec*.

x / y	H.264	HEVC
HEVC	2^{-16}	-
MPEG-4	1,000	1,000